SISTEM INFORMASI PENENTUAN RUTE PATROLI DENGAN METODE ALGORITMA GENETIKA

Nama : Aditya Yusuf

NIM : 00.41010.0180

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

SEKOLAH TINGGI

MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER

SURABAYA

2007

UNIVERSITAS Dinamika

SISTEM INFORMASI PENENTUAN RUTE PATROLI

DENGAN METODE ALGORITMA GENETIKA

SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer

Oleh :

Nama : Aditya Yusuf

NIM : 00.41010.0180

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

UNIVERSITAS Dinamika

SEKOLAH TINGGI

MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER

SURABAYA

2007

Berusahalah dan berdoa semampumu

tetap serahkan hasilnya pada ALLAH

UNIVERSITAS Dinamika

SISTEM INFORMASI PENENTUAN RUTE PATROLI

DENGAN METODE ALGORITMA GENETIKA

Di susun oleh :

Nama : Aditya Yusuf

NIM : 00410100180

Surabaya, Juni 2007

Telah diperiksa, diuji dan disetujui

Disetujui :

Pembimbing I	Pembimbing II
Tutut Wurijanto, M.Kom	Soetam Rizky W, S.Kom
NIDN 0703056702	NIDN 0711097601

Mengetahui :

Wakil Ketua Bidang Akademik

Drs. Antok Supriyanto, M.MT
NIDN. 0726106201

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER SURABAYA

UNIVERSITAS Dinamika

ABSTRAKSI

Dalam menentukan rute patroli sering menjadi kendala pada Satuan SAMAPTA Polwiltabes Surabaya. Faktor banyaknya kejadian di jalan sering menjadi kendala dalam menetukan rute patroli yang optimal. Untuk membantu menentukan rute patroli perlu adanya sistem optimasi rute patroli yang mengatasi permaslahan diatas.

Dalam tugas Akhir ini akaan merancang sistem penentuan rute patroli dengan menggunakan metode algoritma genetika pada SAMAPTA Polwiltabes Surabaya. Algoritma genetika merupakan suatu metode yang menggunakan seleksi alam yang merupakan bagian utama dari prinsip evolusi sebagai dasar pemikiran untuk menyelesaikan suatu permasalahan. Prinsip ini dikemukakan oleh Charles Darwin, dimana tanpa menghiraukan prinsip dasar penurunan sifat, Darwin mengemukakan penggabungan kualitas induk pada generasi berikutnya, disamping itu bahwa individu yang mampu beradaptasi dengan lingkungannya akan mempunyai kesempatan hidup yang lebih besar.

Berdasarkan hasil simulasi yang telah dilakukan pada Tugas Akhir ini dengan menggunakan data detail kejadian pada tahun 2006 dengan menggunakan *one-point crosssover*, mutasi, generasi sebesar 30 dan populasi sebesar 30 diperoleh solusi terbaik dari kasus optimalisasi rute patroli. Hasil nilai fitness yang diperoleh sebesar 6,025 dengan probabilitas mutasi 10% dan probabilitas *crossover* 45% untuk jumlah generasi 30 dan individu 30 pada 1 populasi.

UNIVERSITAS Dinamika

KATA PENGANTAR

Dengan rasa syukur ke hadirat Allah SAW, berkat rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir ini. Penyusunan Tugas Akhir ini adalah syarat dalam menyelesaikan Program Studi Strata Satu di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya. Terima kasih kepada Ayah dan Bunda yang selalu mendorong penulis menyelesaikan Tugas Akhir ini, tak lupa ucapan terima kasih juga penulis tujukan untuk Bapak Tutut Wurijanto, M.Kom selaku dosen Pembimbing I yang telah banyak membimbing dan membantu dalam menyelesaikan Tugas Akhir ini, Bapak Soetam Rizky W, S.Kom, selaku dosen Pembimbing II yang telah membimbing penulis dalam menyelesaikan Tugas Akhir.

Banyak pihak yang telah memberikan bantuan dalam menyusun Tugas Akhir ini, untuk itu penulis juga menyampaikan rasa terima kasih kepada :

1. Kepada Ayah, Bunda dan saudara-saudaraku mbak Nita, mas Awal, Ryan dan Rifqy yang telah memberikan do'a, dukungan dan perhatian kepada penulis.
2. Semua teman-teman yang mendukung penulias menyelesaikan Tugas Akhir ini antara lain Catur, Jiwo, Candra, Dafiq, Sarkini, dan tak lupa Vinda.
3. Khususnya mas Awal dan Catur yang telah membantu penulis menyelesaikan tugas akhir ini.
4. Tak lupa untuk semua orang yang telah membantu penulis yang tidak bisa disebutkan namanya, penulis ucapkan terima kasih.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, namun penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi semua

UNIVERSITAS Dinamika

pihak dan dapat ikut menunjang perkembangan teknologi dan ilmu pengetahuan, khususnya ilmu komputer.

Surabaya, Juni 2007

Penulis

UNIVERSITAS Dinamika

DAFTAR ISI

Halaman

ABSTRAKSI v

KATA PENGANTAR vi

DAFTAR ISI viii

DAFTAR TABEL xi

DAFTAR GAMBAR xii

BAB I PENDAHULUAN 1

1.1 Latar Belakang Masalah 1

1.2 Perumusan Masalah 2

1.3 Pembatasan Masalah 3

1.4 Tujuan 3

1.5 Sistematika Penulisan 4

BAB II LANDASAN TEORI 5

2.1 Algoritma Genetika 5

2.1.1 Menentukan Representasi Kromosom 7

2.1.2 Inisialisasi Populasi 10

2.1.3 Menentukan Rumusan Fungsi Fitness 10

2.1.4 Parameter Genetika 11

2.1.5 Proses Seleksi 13

2.1.6 Operasi Genetik 13

2.1.7 Kriteria Berhenti 19

2.1.8 Breder Genetic Algorithn 19

2.2 Rute Kendaraan 20

UNIVERSITAS Dinamika

2.2.1 Travelling Salesman Problem 21
2.2.2 Multiple travelling Salesman Problem..... 22
2.2.3 Vehicel Routing Problem..... 23
2.2.4 Chinese Postman Problem..... 24
2.3 Konsep Rute Patroli 27
2.4 Geographic Information System (GIS) 27
2.4.1 Proses GIS 30
BAB III PERANCANGAN SISTEM 31
3.1 Identifikasi Masalah..... 32
3.2 Perancangan Algoritma..... 34
3.2.1 Pengkodean (Representasi Kromosom) 34
3.2.2 Inisialisasi Populasi..... 35
3.2.3 Fungsi Fitness..... 35
3.2.4 Seleksi 37
3.2.5 Operator Genetika 40
3.2.6 Kriteria Berhenti..... 44
3.2.7 Breder GA (Pelestarian Kromosom)..... 44
3.3 Geographis Information System (GIS) 44
3.4 Desain Sistem..... 46
3.4.1 Context Diagram 47
3.4.2 Data Flow Diagram (DFD) 48
3.4.3 Entity Relationship Diagram (ERD)..... 50
3.4.4 Struktur Database..... 51
BAB IV IMPLEMENTASI DAN EVALUASI..... 55

UNIVERSITAS Dinamika

4.1 Kebutuhan Sistem 55
4.1.1 Perangkat Keras (Hardware) 55
4.1.2 Perangkat Lunak (Sofware)..... 55
4.2 Implementasi Sistem 56
4.3 Evaluasi Sistem 65
4.3.1 Uji Coba I Terhadap Generasi..... 66
4.3.2 Uji Coba II Terhadap Populasi..... 67
4.3.3 Uji Coba III Terhadap Probabilitas Crossover..... 68
4.3.4 Uji Coba IV Terhadap Probabilitas Mutasi..... 69
BAB V PENUTUP..... 72
5.1 Kesimpulan 72
5.2 Saran..... 72
DAFTAR PUSTAKA 74
LAMPIRAN 75

UNIVERSITAS Dinamika

DAFTAR TABEL

Halaman

Tabel 2.1 Tipe – tipe Permasalahan Pembentukan Rute 20

Tabel 3.1 Tabel Data Jalan 52

Tabel 3.2 Tabel Data Jarak Protokol 52

Tabel 3.3 Tabel Data Petugas 52

Tabel 3.4 Tabel Hasil Proses 52

Tabel 3.5 Tabel Mobil Patroli 53

Tabel 3.6 Tabel Jadwal Patroli 53

Tabel 3.7 Tabel Detail Kejadian 54

Tabel 3.8 Tabel Jenis-Jenis Kejadian 54

Tabel 3.9 Tabel Hasil Akhir Proses GA 54

Tabel 4.1 Uji Coba I Generasi 66

Tabel 4.2 Uji Coba II Populasi 67

Tabel 4.3 Uji Coba III Probabilitas Crossover 68

Tabel 4.4 Uji Coba IV Probabilitas Mutasi 69

UNIVERSITAS Dinamika

DAFTAR GAMBAR

Halaman

Gambar 2.1 Gambaran Umum Algoritma Genetika 6

Gambar 2.2 Kromosom Dalam Algoritma Genetika 7

Gambar 2.3 Permutation Encoding 8

Gambar 2.4 Binary Encoding 9

Gambar 2.5 Proses Crossover dengan Satu Crossover Point 14

Gambar 2.6 Ilustrasi Dari Operator Mutasi Inversi 17

Gambar 2.7 Ilustrasi Dari Operator Mutasi Insersi 18

Gambar 2.8 Ilustrasi Dari Operator Mutasi Pemindahan 18

Gambar 2.9 Travelling Salesman Problem 21

Gambar 2.10 Multiple Travelling Salesman Problem 23

Gambar 2.11 Vehicle Routing Problem 24

Gambar 2.12 Chinese Postman Problem 24

Gambar 2.13 Alur Proses GIS 30

Gambar 3.1 Diagram Alir Penyelesaian Proses 33

Gambar 3.2 Kromosom Rute Patroli 34

Gambar 3.3 Nilai Fitness Kromosom Rute Patroli 36

Gambar 3.4 Proses Crossover 40

Gambar 3.5 Kromosom Hasil Crossover 42

Gambar 3.6 Proses Mutasi 43

Gambar 3.7 Contoh Hasil Rute 45

Gambar 3.8 Proses Optimasi Rute Patroli 46

Gambar 3.9 Context Diagram Optimasi Rute Patroli 47

UNIVERSITAS Dinamika

Gambar 3.10 DFD level 0 48
Gambar 3.11 DFD level 1 Proses 1 49
Gambar 3.12 Gambar ERD-CDM (*Conceptual Data model*) 50
Gambar 3.13 Gambar ERD-PDM (*Conceptual Data model*) 51
Gambar 4.1 Form Login 56
Gambar 4.2 Form Menu Utama 57
Gambar 4.3 Form Master Mobil 58
Gambar 4.4 Master Petugas Piket 59
Gambar 4.5 Master Jenis Kejadian 60
Gambar 4.6 Jadwal petugas Piket 61
Gambar 4.7 Pengaduan Kejadian / Perkara 62
Gambar 4.8 Proses GA 63
Gambar 4.9 Data Jalan Dan Jumlah Kejadian 63
Gambar 4.10 Form Proses Optimasi Rute Patroli 64
Gambar 4.11 Form Hasil Akhir 65
Gambar 4.12 Grafik Uji Coba I 67
Gambar 4.13 Grafik Uji Coba II 68
Gambar 4.14 Grafik Uji Coba III 69
Gambar 4.15 Grafik Uji Coba IV 70
Gambar 4.16 Laporan Rute 71

UNIVERSITAS Dinamika

DAFTAR LAMPIRAN

Halaman

Lampiran 1 Data-data kejadian di Polwiltabes Surabaya 79

UNIVERSITAS Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Surabaya merupakan salah satu kota terbesar ke dua di Indonesia, yang juga merupakan kota metropolitan. Penduduk Surabaya yang terdiri dari beragam suku dan ras, dimungkinkan banyak timbul terjadinya kejahatan yang dapat meresahkan masyarakat. Meningkatnya kasus kejahatan di Surabaya yang antara lain penganiayaan berat (anirat), pencurian dengan kekerasan (curas), pencurian dengan pemberatan (curat), pencurian kendaraan bermotor (curanmor), kebakaran, penyakit masyarakat (contohnya: minuman keras, narkoba), unjuk rasa, maupun gangguan-gangguan lainnya yang dapat menimbulkan rasa tidak aman bagi masyarakat, hal ini membuat jajaran SAMAPTA Polwiltabes Surabaya dituntut lebih meningkatkan tugas dalam pengamanan kota. Salah satu usaha meningkatkan keamanan kota yaitu dengan menempatkan pos-pos anggota SAMAPTA di daerah-daerah yang diprediksi banyak terjadi tindak kejahatan, atau dengan lebih meningkatkan rute patroli di daerah-daerah yang diprediksi banyak terjadi tindak kejahatan, pendekatan ini juga dilakukan agar jajaran kepolisian khusunya anggota dapat SAMAPTA lebih mendekatkan diri pada masyarakat, sehingga masyarakat dapat lebih mudah melaporkan suatu kejadian dan lebih cepat ditangani oleh anggota SAMAPTA tanpa harus ke kantor polisi.

Salah satu usaha dalam dalam mengurangi suatu tindak kejahatan yaitu dengan mengoptimalkan rute patroli. Masih terdapat hambatan dalam pemilihan rute patroli, dimana area yang dijangkau sangat luas, sedangkan jumlah unit mobil

patroli SAMAPTA yang sedang bertugas di lapangan terbatas, padahal tidak semua jalan dapat dilalui oleh unit patroli. Sehingga dari hambatan tersebut masih ditemukan laporan-laporan kejadian dari masyarakat, laporan-laporan dari korban tindak kejahatan, maupun temuan kejadian-kejadian oleh anggota SAMAPTA untuk yang terlambat untuk ditangani. Hal ini tentunya dapat mengurangi kredibilitas jajaran kepolisian.

Berdasarkan pada hambatan di atas, maka diperlukan suatu sistem yang dapat membantu mengoptimalkan rute patroli SAMAPTA, sehingga dapat mengatasi permasalahn pihak kepolisian dalam mengurangi tindak kejahatan. Dimana untuk menghasilkan solusi optimal yang cukup memuaskan (*satisfactory solution*) dari permasalahan di atas, akan digunakan model *heuristic search*, yang telah banyak digunakan dan terbukti cukup tepat untuk permasalahan optimasi (Kusumadewi,2003:279) dan salah satu perkembangan dari model tersebut adalah Algoritma Genetika. Lebih daripada itu, akan digunakan *Geographic Information System* (GIS), sehingga pemilihan rute patroli dapat tergambarkan dengan jelas. Penggunaan metode ini diharapkan dapat membantu jajaran kepolisian khusunya Satuan SAMAPTA Polwiltabes Surabaya dalam memilih rute patroli anggota untuk mengamankan kota lebih optimal, berdasarkan data detail kejadian yang pernah terjadi dari setiap daerah.

1.2 Perumusan Masalah

Berdasarkan pada latar belakang masalah di atas, maka permasalahan pada tugas akhir ini, adalah bagaimana mengoptimalkan rute patroli anggota SAMAPTA Polwiltabes Surabaya di wilayah Surabaya dengan menggunakan metode Algoritma Genetika.

UNIVERSITAS Dinamika

1.3 Pembatasan Masalah

Adapun batasan-batasan masalah yang digunakan yaitu :

1. Data detail kejadian kejahatan yang terdapat Polwiltabes Surabaya, yang telah terjadi di tiap-tiap ruas jalan protokol di Surabaya.
2. Yang termasuk jalur patroli adalah jalan protokol, karena jalur protokol adalah jalur utama kota Surabaya.
3. Hanya mengambil jalur protokol sebagai contoh, karena jalan protokol merupakan akses vital bagi arus transportasi di dalam kota, maupun luar kota.
4. Tidak menghitung biaya operasional yang dikeluarkan oleh kepolisian, dan jumlah anggota pada pos patroli.
5. Unit patroli maupun anggota SAMAPTA diasumsikan telah siap bertugas.
6. Inputan tujuan dilakukan secara manual, yaitu berdasarkan data kejadian tiap jalan protokol pada tiap bulan, dan tidak melakukan klik atau inputan pada peta.
7. Jalan protokol diasumsikan terhubung semua, dan panjang jalan diukur secara manual.
8. Menggunakan database Microsoft Office Acess

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah optimalisasi rute patroli anggota SAMAPTA berdasarkan detail kejadian yang telah dilaporkan oleh anggota SAMAPTA Polwiltabes Surabaya dengan menggunakan metode Algoritma Genetika di wilayah Surabaya.

UNIVERSITAS Dinamika

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Dalam bab ini diuraikan mengenai latar belakang permasalahan, rumusan permasalahan, pembatasan masalah, tujuan, dan sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini berisi uraian mengenai teori-teori yang digunakan dalam penyelesaian permasalahan.

BAB III : METODE PENELITIAN/PERANCANGAN SISTEM

Bab ini menjelaskan metode penelitian yang digunakan dalam merancang dan membangun sistem menggunakan metode Genetika Algoritma untuk mengoptimalisasikan rute patroli.

BAB IV : IMPLEMENTASI DAN EVALUASI

Bab ini memberikan gambaran berupa implementasi sistem, *input* dan *output* serta uji coba yang akan digunakan pada optimalisasi dengan menggunakan metode Genetika Algoritma.

BAB V : PENUTUP

Berisi kesimpulan pembahasan permasalahan yang telah dilakukan dan saran bagi pengembangan sistem yang dibuat.

UNIVERSITAS Dinamika

BAB II

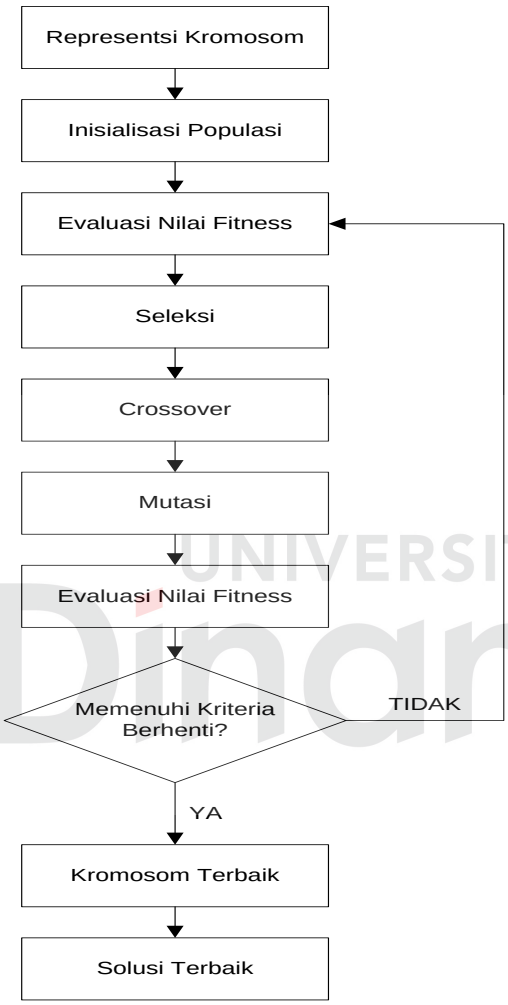
LANDASAN TEORI

2.1 Algoritma Genetika

Algoritma genetika merupakan suatu metode yang menggunakan seleksi alam yang merupakan bagian utama dari prinsip evolusi sebagai dasar pemikiran untuk menyelesaikan suatu permasalahan. Prinsip ini dikemukakan oleh Charles Darwin, dimana tanpa menghiraukan prinsip dasar penurunan sifat, Darwin mengemukakan penggabungan kualitas induk pada generasi berikutnya, disamping itu bahwa individu yang mampu beradaptasi dengan lingkunganya akan mempunyai kesempatan hidup yang lebih besar.

Penggunaan prinsip genetika pada komputer dimulai pada tahun 1950 ketika beberapa ahli Biologi menggunakan komputer untuk simulasi sistem biologi. Akhir tahun 1975 John Holland dari Universitas Michigan melalui paper yang berjudul "*Adaption in Natural and Artificial System*" mengunakan konsep dasar algoritma genetika. Algoritma genetika bekerja dengan suatu populasi *string* dan melakukan proses pencarian nilai optimal secara paralel, dengan mengunakan operator genetika. Algoritma genetika akan melakukan rekombinasi antar individu. Algoritma genetika memiliki elemen dasar berupa *string* yang tersusun dari rangkaian *substring* (gen), yang masing-masing merupakan kode dari parameter dalam ruang solusi dimana suatu *string* (kromosom) menyatakan kandidat solusi. Kumpulan *string* dalam populasi berkembang dari generasi ke generasi melalui operator genetika. Pada setiap iterasi, individu-individu (Kromosom) dalam populasi itu akan dievolusi dan diseleksi untuk menentukan

populasi pada generasi berikutnya. Populasi ini akan terus berulang sampai menemukan suatu parameter dengan nilai yang paling optimal sesuai dengan yang diinginkan (Kusumadewi,2003:232). Adapun struktur umum algoritma genetika dapat diilustrasikan pada Gambar 2.1.



Gambar 2.1 Gambaran Umum Algoritma Genetika

Istilah yang digunakan dalam algoritma genetika ini hampir sama dengan istilah yang dipakai dalam bidang biologi genetika, antara lain Gen, Kromosom, Populasi, Fungsi *Fitness*, dan operator genetika yang meliputi mutasi dan *crossover*.

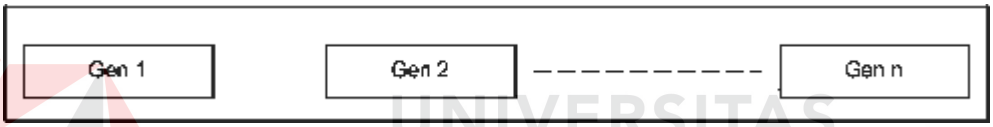
UNIVERSITAS Dinamika

1. **Gen**

   Gen adalah suatu sel dari suatu kromosom atau nilai yang terdapat dalam algoritma genetika ini dapat dibentuk oleh sebuah *byte* bahkan tidak menutup kemungkinan suatu *string*. Gen ini mewakili sebagian kecil dari solusi permasalahan, dari kumpulan gen ini diharapkan terdapat terdapat salah satu gen yang unggul atau yang dapat menjadi sosuli paling optimal.

2. **Kromosom**

   Individu dalam populasi disebut *string*, *genotype* atau kromosom terdiri dari unit yang dinamakan Gen, Karakter, *Decoder*. Kromosom ini dapat mewakili suatu solusi, dimana dapat diilustrasikan dalam Gambar 2.2



Gambar 2.2 Kromosom dalam Algoritma Genetika

3. **Populasi.**

   Populasi adalah sekumpulan kromosom yang dapat berisi *string*, *byte*, atau bisa disebut gen, yang dapat merepresentasikan suatu solusi.

4. **Fungsi Fitness (Fungsi Evaluasi)**

   Fungsi evaluasi adalah suatu formula yang digunakan untuk menilai suatu kromosom dan digunakan pada setiap generasi.

2.1.1 Menentukan Representasi Kromosom

Untuk mempresentasikan kromosom dilakukan dengan proses *encoding* atau teknik penyandian yang meliputi gen dan kromosom dimana gen merupakan

UNIVERSITAS Dinamika

bagian dari kromosom (Kusumadewi,2003:233), dibawah ini akan dijelaskan beberapa proses *encoding* yang biasa digunakan dalam beberapa kasus tertentu.

A. *Permutation Encoding*

Untuk jenis *Permutation Encoding* ini digunakan untuk permasalahan proses pengurutan, misalnya terdapat kasus optimasi jadwal atau pada kasus *traveling salesman*. Pada *Permutation Encoding*, setiap Gen pada kromosom berupa angka dimana dapat ditampilkan seperti Gambar 2.3

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

Gambar 2.3 *Permutation Encoding*

Permutation Encoding hanya berlaku untuk permasalahan pengurutan. Dalam kasus-kasus yang ada pada *Permutation Encoding* terdapat beberapa jenis *crossover* dan mutasi yang harus dibuat untuk mempertahankan kromosom agar tetap konsisten. Contoh penggunaan *Permutation Encoding* ini ada pada kasus *travelling sales problem*, dimana terdapat beberapa kota dengan jarak masing-masing. Pada kasus *traveling sales problem* ini seorang *salesman* harus mengunjungi semua kota yang ada. Permasalahannya adalah menentukan urutan kota yang akan dikunjungi untuk meminimalisasi jarak yang harus ditempuh.

B. *Binary Encoding*

Binary encoding adalah jenis encoding yang paling sering digunakan karena kasus pertama yang ada pada Algoritma Genetika menggunakan algoritma jenis ini. Setiap kromosom pada *Binary encoding* merupakan bit 0 dan 1 di mana dapat ditampilkan pada Gambar 2.4

UNIVERSITAS Dinamika

Kromosom A	1001101011010010
Kromosom B	1001011001011001

Gambar 2.4 *Binary Encoding*

Binary encoding dapat memberikan banyak kemungkinan pada kromosom meskipun pada jumlah Gen yang sedikit. Di lain pihak jenis *encoding* ini tidak cukup natural untuk beberapa kasus tertentu dan kadang-kadang harus dilakukan koreksi setelah melakukan *crossover* atau mutasi, contoh penggunaan *Binary encoding* adalah pada permasalahan *knapsack* atau pengepakan, di mana ada beberapa barang dengan jumlah dan ukuran masing-masing dan *knapsack* harus memberikan kapasitasnya untuk barang-barang tersebut, permasalahanya adalah bagaimana memilih barang untuk memaksimalkan jumlah barang sehingga dapat ditampung oleh *knapsack* tanpa harus menambah kapasitasnya.

C. *Value Encoding*

Penempatan nilai yang ada secara langsung dapat diterapkan pada *Value Encoding* jenis ini dapat menampung nilai yang rumit dan berbeda seperti tipe real. Penggunaan *Binary encoding* untuk jenis permasalahan ini sangatlah sulit dilakukan, setiap kromosom pada *Value Encoding* adalah suatu nilai tertentu, dimana nilai tersebut dapat berupa bilangan desimal (5,32,3,9 atau lainnya), bilangan real (1.4545,0.3543, 3.6,5.468 atau lainnya), *character* sampai dengan tipe obyek yang rumit

Value Encoding dapat digunakan untuk permasalahan yang spesial, untuk itu penggunaan *Value Encoding* ini sering kali harus menciptakan jenis *crossover* dan mutasi sendiri untuk memecahkan permasalahannya. Contoh penggunaan *Value Encoding* adalah menentukan bobot untuk jaringan neural, dimana jaringan

UNIVERSITAS Dinamika

neural tersebut telah diberikan arsitektur tertentu. Permasalahannya adalah bagaimana mancari bobot untuk masukan neural yang digunakan untuk memproses jaringan neural untuk menghasilkan output yang diharapkan.

2.1.2 Inisialisasi Populasi

Proses inisialisasi adalah pembentukan populasi awal yang diperoleh dengan cara rekombinasi kromosom sebanyak ukuran populasi yang ditentukan. Pada inisialisasi populasi ditentukan nilai awal dari sejumlah untaian kromosom yang juga merupakan kemungkinan solusi permasalahan yang dicari. Populasi awal ini dapat dibangun dengan cara *heuristik* atau acak. Dalam inisialisasi populasi ini tetap memperhatikan batasan-batasan yang akan digunakan dalam suatu permasalahan yang akan diselesaikan.

Adapun dalam inisialisasi populasi menggunakan batasan-batasan sebagai berikut :

1. **Jumlah kejadian / ruas jalan > = 3 tiap bulan**
2. **t = S / v**

Batasan (1) jumlah kejadian yang terjadi pada suatu jalan protokol lebih atau sama dengan tiga untuk dinyatakan sebagai jalan yang banyak terjadi suatu tindak kejahatan (2) t menunjukkan total waktu satu mobil patroli mengunjungi seluruh tujuan, S menunjukkan total jarak antara suatu jalan dengan jalan lain dalam satu rute, sedangkan v adalah kecepatan mobil yang merupakan sebuah (Foster,2004:49).

2.1.3 Menentukan Rumusan Fungsi Fitness

Perhitungan pada permasalahan optimalisasi rute yang timbul dapat di formulasikan sebagai berikut :

UNIVERSITAS Dinamika

$$Eval(K_1) = \sum_{i}^{r} \quad \text{.....} \quad (4)$$

dimana, r = 1,2,3,...,n

i = 1,2,3,...max pop_size

Untuk rumusan fungsi fitness (3) menunjukkan jumlah total waktu tempuh dari suatu mobil patroli. Parameter yang digunakan adalah : r sebagai rute patroli, K_i sebagai kromosom ke – i.

2.1.4 Parmeter Genetika

A. Ukuran Populasi (uk_pop)

Ukuran populasi akan mempengaruhi kinerja dan efektifitas dari algoritma genetika. Jika ukuran populasi kecil maka populasi tidak akan menyediakan cukup materi untuk mencakup ruang permasalahan, sehingga pada umumnya kinerja algoritma genetika menjadi tidak optimal atau buruk dan apabila penggunaan populasi yang besar dapat mencegah terjadinya konvergensi pada wilayah lokal (Kusumadewi,2003:235).

B. Jumlah Generasi

Merupakan jumlah perulangan (*iterasi*) dilakukannya rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan *output* dan lama iterasi (waktu proses algoritma genetika). Jumlah generasi yang besar dapat mengarahkan ke arah solusi yang optimal, namun dalam mendapatkan solusinya akan membutuhkan akan membutuhkan waktu komputasi yang lama. Sedangkan jika jumlah generasinya terlalu sedikit maka solusi yang di dapatkan akan terjebak pada lokal optimal.

UNIVERSITAS Dinamika

C. Probabilitas Crossover (Pc)

Probabilitas *crossover* digunakan untuk mengendalikan frekuensi operator *crossover*. Probabilitas *crossover* menyatakan seberapa banyak *crossover* yang dilakukan sehingga kromosom yang tidak melakukan *crossover* akan menghasilkan suatu individu baru yang sama dengan indukya. Jika melakukan *crossover* akan menghasilkan individu yang memiliki sifat dari kromosom induknya. Jika probabilitas *crossover*nya 100% maka semua kromosom yang terbentuk adalah dari proses *crossover* dan jika probablilitas *crossover*nya 0% maka kromosom yang terbentuk berasal dari kromosom sebelumnya tetapi belum tentu generasi yang baru sama dengan generasi sebelumnya. Pada umumnya sebaiknya angka probabilitas *crossover* adalah tinggi, sekitar 80% sampai 95% (tetapi pada kasus-kasus tertentu angka probabilitas *crossover* adalah 60% akan menghasilkan solusi yang lebih baik). Dengan proses *crossover* diharapkan generasi yang baru akan membawa sifat yang baik dari kromosom induknya sehingga kromosom baru yang terbentuk akan memberikan solusi yang optimal.

Dalam hal ini, pada populasi terdapat Pc * uk_pop struktur (individu) yang melakukan *crossover* (Kusumadewi,2003:350).

D. Probabilitas Mutasi (Pm)

Mutasi digunakan untuk meningkatkan variasi populasi. Probabilitas mutasi ini menentukan berapa banyak dari kromosom melakukan proses mutasi karena frekuensi terjadinya mutasi tersebut menjadi Pm * uk_pop * N, dimana N adalah panjang struktur (gen) dalam suatu individu.. Apabila tidak dilaksanakan proses mutasi maka kromosom yang dihasilkan tepat sama dengan induknya. Probabilitas mutasi 100% berarti semua kromosom akan berubah dan jika

probabilitas 0% berarti tidak ada bagian kromosom yang berubah. Berbeda dengan probabilitas *crossover*, angka probabilitas mutasi sebaiknya sangat rendah. Angka terbaik yang pernah dihasilkan untuk menentukan angka probabilitas mutasi adalah berkisar antara 0,5% sampai 1%. Mutasi dilakukan untuk mencegah algoritma genetika terjebak dalam *local extreme*, tetapi proses ini tidak boleh dilakukan terlalu sering karena algoritma genetika akan barubah menjadi pencarian acak atau *random search* (Kusumadewi,2003:235,246)..

2.1.5 Proses Seleksi

Proses ini digunakan untuk menentukan kromosom-kromosom yang akan bertahan dan masuk pada generasi selanjutnya. Penentuan suatu kromosom akan dimasukkan pada generasi selanjutnya didasarkan pada nilai fitness untuk tiap-tiap kromosom. Semakin baik nilai fitness maka probabilitas kromosom tersebut akan terpilih semakin besar. Metode yang sering digunakan dalam proses seleksi ini adalah roulete wheel.

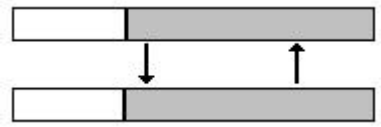
2.1.6 Operasi Genetik

A. *Crossover*

Crossover adalah operator algoritma genetika yang membutuhkan parameter dua kromosom atau lebih. Dua buah kromosom atau lebih tersebut disebut kromosom induk. Operator ini akan menghasilkan dua buah kromosom baru atau lebih. Ada beberapa jenis proses *crossover* yang sering digunakan dalam algoritma genetika antara lain:

A.1 *One-Point Crossover*

Contoh kerja operator ini adalah dengan menentukan *crossover point* (gen tertentu). Kromosom baru pertama berisi gen pertama sampai gen *crossover point* dari kromosom induk pertama ditambah dengan gen dari *crossover point* sampai gen terakhir dari kromosom induk kedua. Kromosom baru kedua berisi gen pertama sampai gen *crossover point* dari induk kedua ditambahkan dengan gen dari *crossover point* sampai gen dari kromosom induk pertama.



Gambar 2.5. Proses *Crossover* dengan Satu *Crossover* Point

Dari ilustrasi di atas maka contoh penerapan metode *One-Point Crossover* adalah sebagai berikut:

Parent 1: 1 2 3 4 5 | 6 7 8 9

Parent 2: 4 5 3 6 8 | 9 7 2 1

Setelah proses *crossover* turunan yang dapat dihasilkan adalah dari kedua *parent* diatas adalah:

Parent 1: 1 2 3 4 5 | 9 7 2 1

Parent 2: 4 5 3 6 8 | 6 7 8 9

A.2 *Two-Point Crossover*

Proses *Two-Point Crossover* hampir sama dengan prosedur *One-Point Crossover*, kecuali pada *Two-Point Crossover* harus dipilih dua *crossover point* dan hanya gen yang ada di antara kedua *crossover point* itu yang akan ditukarkan.

Metode ini dapat menjadi bagian awal dan akhir dari kromosom dan hanya menukar bagian tengahnya saja.

UNIVERSITAS Dinamika

A.3 *N-Point Crossover*

Prosedur *N-Point Crossover* hampir sama baik dengan prosedur *one-point crossover* maupun *two-point crossover*, hanya saja dalam *n-point crossover* ini harus dipilih n *crossover point* dan hanya gen di antara *crossover point* ganjil dan genap yang dapat ditukarkan sedangkan gen diantara genap dan ganjil operator *crossover* tidak berubah. Atau dengan kata lain harus dipilih posisi n dan hanya bit antara ganjil dan genap posisi *crossover* yang akan dihilangkan.

Contoh: P1= 9 7 6 3 2 8

P2= 2 1 9 7 4 5

Jika didapatkan angka random untuk n=3 dan diacak 2, 5 dan 6 sebagai posisi dari gen yang akan di *crossover*, didapatkan kromosom turunan:

T1= 9 ***1*** 6 3 ***4*** ***5***

T2= 2 ***7*** 9 7 ***2*** ***8***

A.4 *Uniform Crossover*

Metode ini telah dipelajari secara detail dan di populerkan oleh *Syswerda* dengan menjadikan *crossover* ini lebih baik dari penyelesaian masalah secara tradisional (Gen, 1997:215). Cara kerja metode ini adalah diacak angka untuk setiap posisi bit 1 sampai dengan L bagi dua kromosom parent.

Contoh: P1= 9 7 6 3 2 8

P2= 2 1 9 7 4 5

Misalnya didapatkan angka random antara 0 sampai dengan 1 = 0.1, 0.6,0.3,0.7,0.4 maka posisi dari gen dimana angka yang telah ditemukan tadi kurang dari 0.5 akan dilakukan proses *crossover* sedang sekarang didapat kromosom turunan:

UNIVERSITAS Dinamika

T1= 9* 1 6 3 4* 5*

T2= 2* 7 9 7 2* 8*

Jadi untuk setiap gen dari parent pertama memiliki probabilitas 0.5 untuk dihilangkan dari gen yang bersangkutan dari parent yang kedua.

B. Operator Mutasi

Mutasi adalah operator yang membutuhkan satu perameter. Kromosom operator ini merupakan nilai suatu gen dari sebuah kromosom sehingga kromosom yang baru ini berbeda dengan kromosom yang lama. Sekumpulan kejadian dengan suatu nilai pelanggaran maksimal dapat dengan mudah dihilangkan selama evaluasi *fitness*, tujuan dari proses mutasi ini, untuk mempertahankan kehilangan *permanent* dari suatu bit atau gen. Seluruh proses mutasi ini menjanjikan keuntungan melalui pengarahan mutasi kemana mutasi ini tersebut sangat dibutuhkan. Operator mutasi digunakan untuk melakukan modifikasi satu atau lebih dari nilai gen dalam individu yang sama. Mutasi memastikan bahwa probabilitas untuk pencarian pada daerah tertentu dalam persoalan tidak akan pernah nol dan mencegah kehilangan total materi genetika setelah pemilihan dan penghapusan. Mutasi ini bukanlah operator genetika yang utama, yang dilakukan secara acak pada gen dengan kemungkinan yang lebih kecil (Kusumadewi,2003:246).

<u>Langkah-langkah algoritma genetika mutasi :</u>

Step 1 = [Tentukan kromosom atau *parent* yang akan dimutasi]
i ← random N-1
Pos ← Pop [i]

Step 2 = [Tentukan poisi *allele* yang akan dimutasi]
K ← random N-1
L ← random N-1

UNIVERSITAS Dinamika

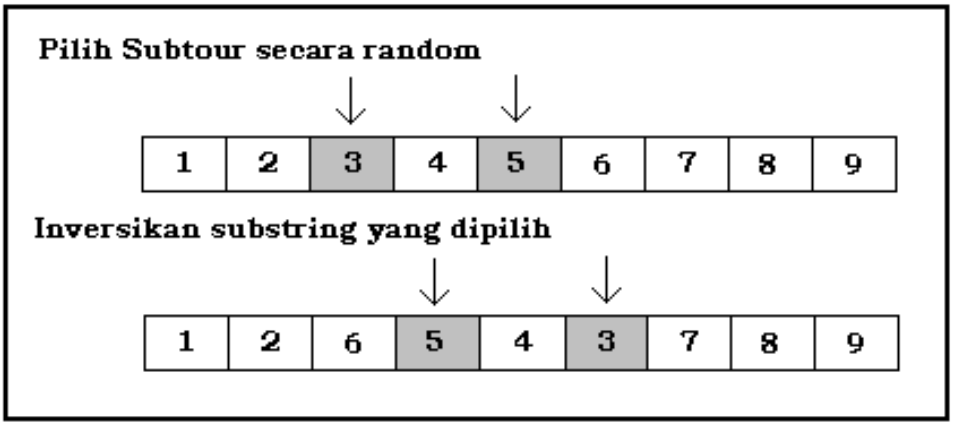
Step 3 = [salin *allele* pada posisi K ke dalam variable temp]
Temp ⟵ Pop [K]
Step 4 = [Lakukan pertukaran *allele*]
Pop [K] ⟵ Pop [L]
Pop [L] ⟵ temp
Step 5 = [Selesai]
Return

N adalah jumlah kromosom yang akan diproses. Pop adalah kromosom yang akan dimutasi. Variabel K dan L digunakan untuk menyimpan posisi dari *allele* yang akan di tukarkan dalam proses mutasi. Variabel temp digunakan untuk menyimpan nilai *allele* yang ditunjuk oleh posisi K.

Terdapat empat operator yang biasa digunakan untuk representasi permutasi, yaitu :

B.1 Mutasi Inversi (*Inversion Mutation*)

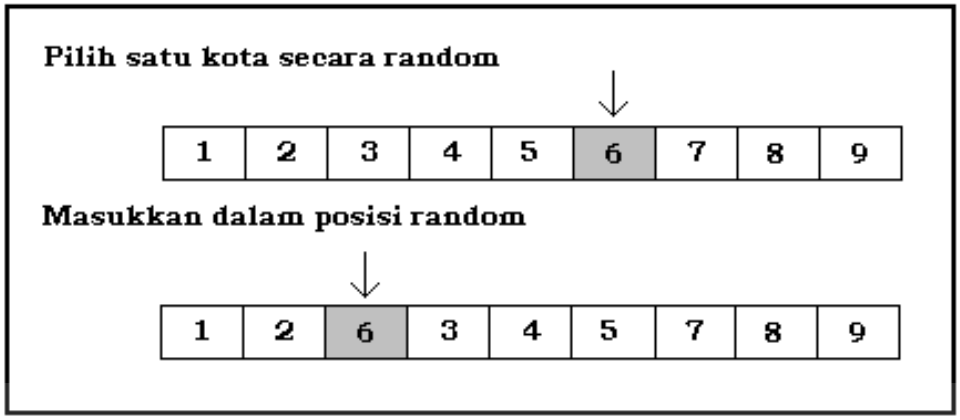
Mutasi Inversi bekerja seperti teknik pengulangan. Mutasi ini bekerja pada satu kromosom dan membalik urutan diantara dua demen yang dipilih secara acak pada kromosom. Mutasi Inversi memilih dua posisi didalam kromosom secara random dan kemudian menginversikan substring antara dua posisi ini.



Gambar 2.6 Ilustrasi dari Operator Mutasi Inversi

B.2 Mutasi Insersi (Insertion Mutation)

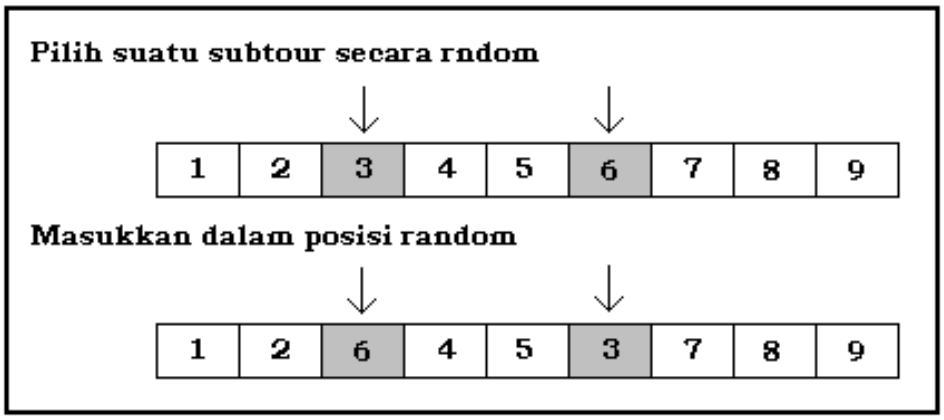
Proses yang dilakukan untuk jenis mutasi insersi ini adalah dengan memilih secara random gen dalam sebuah kromosom dan menyisipkannya pada posisi lain secara random juga sehingga dihasilkan sebuah kromosom baru.



Gambar 2.7 Ilustrasi dari Operator Mutasi Insersi

B.3 Mutasi Pemindahan (*Displacement Mutation*)

Mutasi Pemindahan suatu subtour secara random dan memasukkannya dalam posisi random. Insersi dapat dijelaskan sebagai kasus khusus dalam displacement dimana substring hanya terdiri dari satu kota.



Gambar 2.8 Ilustrasi dari Operator Mutasi Pemindahan

UNIVERSITAS Dinamika

2.1.7 Kriteria Berhenti

Proses *crossover* dan mutasi juga dilakukan perhitungan nilai *fitness* sehingga setiap populasi dinilai dari nilai *fitness*-nya. Hal yang sama dilakukan berulang-ulang sampai mencapai generasi dan hasil terbaik diperoleh.

Setelah melakukan proses algoritma genetika diatas maka proses pencarian solusi akan berhenti jika telah memenuhi salah satu kretiria dibawah ini yaitu :

1. Nilai fitness terbaik telah dicapai. Dimana nilai fitness terbaik, yang terbaik akan dihasilkan secara berurutan dalam N generasi.
2. Nilai fitness terbaik dalam proses generasi memiliki selisih (nilai error) tertentu dengan batas nilai fitness yang telah ditetapkan.
3. Telah mencapai generasi ke – N.

UNIVERSITAS Dinamika

2.1.8 *Breeder* Genetic Algorithm

Seperti yang telah diketahui bahwa algoritma genetika adalah metode seleksi yang dilakukan secara random, sehingga ada kemungkinan bahwa kromosom yang sebenarnya sudah baik tidak bbisa turut serta pada generasi berikutnya karena tidak lolos seleksi. Untuk itu perlu adanya pelestarian kromosom-kromosom terbaik, sehingga kromosom-kromosom yang sudah baik tersebut bisa lolos seleksi. Pada breeder GA ini digunakan parameter r , yang menunjukkan kromosom-kromosom terbaik. Kromosom-kromosom ini akan tetap dipertahankan pada generasi berikutnya dengan cara menggantikan sebnayak r kromosom pada generasi tersebut secara acak.

2.2 Rute Kendaraan

Dasar permasalahan pembentukan rute adalah adanya sekumpulan *node* atau busur yang harus dilayani oleh suatu unit patroli. Tidak ada batasan kapan dan bagaimana urutan pelayanan *entity – entity* yang bersangkutan.

Permasalahannya adalah untuk membentuk suatu rute yang paling optimal dari sekumpulan rute yang memungkinkan untuk masing – masing unit patroli. Terdapat beberapa karakteristik yang dapat digunakan untuk mengidentifikasi tipe – tipe pembentukan rute, antara lain: posisi permintaan, arah perjalanan, jumlah unit patroli. Adapun ketiga tipe permasalahan pembentukan rute tersebut dapat ditunjukkan pada tabel 2.1 berikut ini :

Tabel 2.1 Tipe – tipe Permasalahan Pembentukan Rute

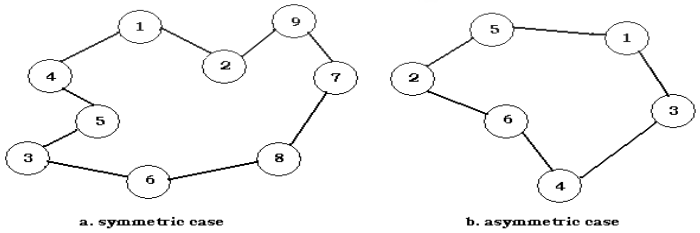
Tipe	**Demand**	**Busur Jarak**	**Jumlah Depot**	**Jumlah Kendaraan**
TSP	Disetiap node	Berarah/tdk	1	=1
MTSP	Disetiap node	Berarah/tdk	1	>1
VRP	Disetiap node & bervariasi besarnya	Berarah/tdk	1	>1
CPP	Sepanjang busur perjalanan	Berarah/tdk	1	≥1

UNIVERSITAS Dinamika

Dari keempat permasalahan rute diatas yang akan digunakan adalah CPP (*Chinese Postman Problem*). Menurut S.Novelianti (Makalah Ilmiah Solusi Traveliing Problem Menggunakan Algoritma Greendy,2006) dari karakteristik tipe permasalahan pembentukan rute tersebut maka dapat dijelaskan sebagai berikut:

2.2.1 *Traveling Salesman Problem* (TSP)

Traveling Salesman Problem berkenaan dengan penentuan biaya perjalanan yang minimal saat melalui setiap node tepat satu kali, dimana jumlah kendaraan hanya satu dengan kapasitas tidak terbatas. Biaya dikatakan *symmetric* jika biaya perjalanan antara dua lokasi tidak tergantung pada arah perjalanan, dan permasalahan tersebut sebagai *symmetric traveling salesman problem*, contoh dapat dilihat pada gambar 2.9.a Sebaliknya jika biaya *asymmetric* maka permasalahannya disebut dengan *asymmetric* traveling salesman problem, contoh dapat dilihat pada gambar 2.9.b



Gambar 2.9 Traveling Salesman Problem

Berikut ini adalah istilah – istilah penting yang perlu digunakan jika terdapat N titik dan m garis :

1. Siklus Hamiltonian adalah suatu siklus yang mampu melewati masing – masing node i € N tepat satu kali. Salah satu contohnya dimisalkan satu network G = (N,A,C) sedangkan dimana N adalah suatu

himpunan node, A adalah himpunan busur, dan C = (cij) adalah matrik biaya, bahwa cij adalah biaya untuk bergerak atau jarak dari node i ke node j. Dalam TSP dibutuhkan siklus Hamiltonian dalam network G pada total biaya minimum.

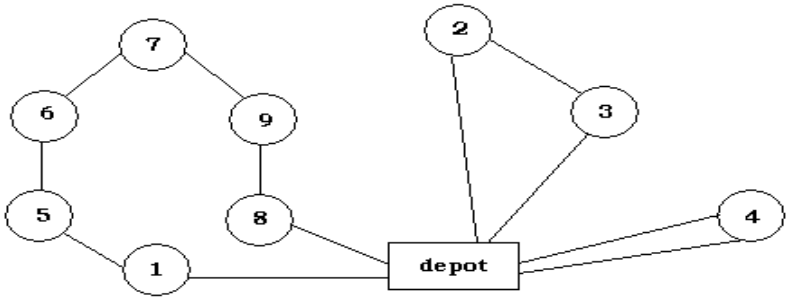
2. *Link* adalah sebuah garis yang menghubungkan dua titik.
3. *Chain* adalah sebuah urutan dari *link* sehingga titik terakhir *link* pertama merupakan permulaan dari *link* kedua dan seterusnya, kecuali untuk *link* pertama dan *link* terakhir; dimana setiap titik dan *link* dalam chain hanya ada satu.
4. Hamiltonian *chain* adalah sebuah rantai yang melalui semua n titik.
5. Hamiltonian *circuit* adalah sebuah rantai yang melalui semua n titik dan titik pertama dan titik terakhir adalah sama.
6. *Tour* adalah sebuah sirkuit hamiltonian
7. *Subtour* adalah sebuah sirkuit hamiltonian yang melalui himpunan bagian dari n titik.
8. *The traveling salesman problem tour* adalah sirkuit Hamiltonian terpendek.
9. *Spanning Tree* adalah kumpulan dari n-1 *link* yang menghubungkan n titik sehingga setiap titik dapat dijangkau dari titik yang lain.
10. *Shortest spanning tree* (SST) adalah *spanning tree* yang terpendek dari *spanning tree* yang ada.

2.2.2 *Multiple Traveling Salesman Problem* (MTSP)

Multiple Traveling Salesman Problem adalah generalisasi dari *traveling salesman problem*, dimana dibutuhkan lebih dari satu salesman atau kendaraan

UNIVERSITAS Dinamika

dalam melakukan perjalanan. M kendaraan berangkat dan kembali ke depot, tidak ada batasan jumlah node sehingga setiap kendaraan boleh melakukan kunjungan kecuali ada batasan jika setiap kendaraan hanya boleh mengujungi satu node. Contoh MTSP dengan 3 kendaraan dapat ditunjukkan pada gambar 2.10

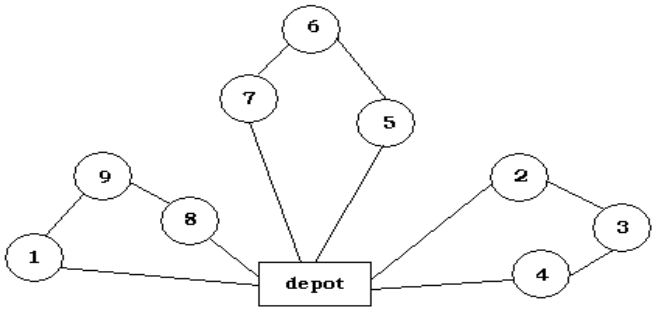


Gambar 2.10 Multiple Traveling Salesman Problem

2.2.3 *Vehicle Routing Problem* (VRP)

Vehicle Routing Problem dapat didefinisikan sebagai permasalahan perancangan rute kendaraan pengiriman yang diketahui kapasitasnya. Adapun tujuan dari mendesain rute kendaraan ini adalah untuk meminimasi beberapa fungsi obyektif seperti meminimasi jumlah armada yang digunakan dan meminimasi total jarak perjalanan yang harus di tempuh.

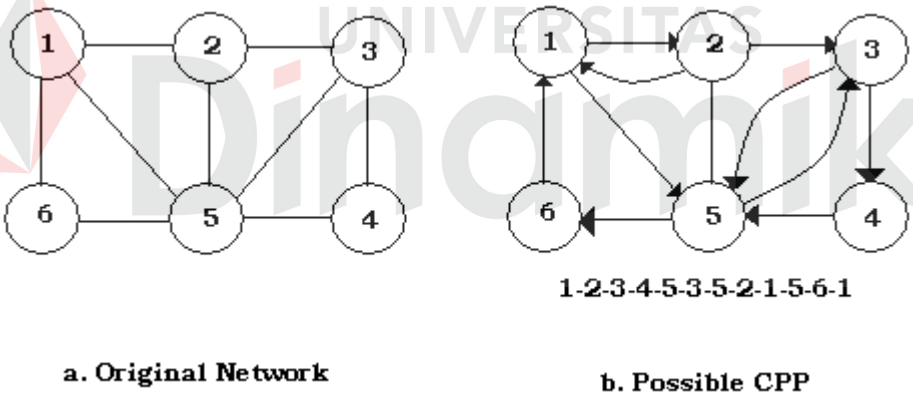
Vehicle Routing Problem dapat didefinisikan sebagai permasalahan perancangan rute kendaraan pengiriman yang diketahui kapasitasnya. Adapun tujuan dari mendesain rute kendaraan ini adalah untuk meminimasi beberapa fungsi objektif seperti meminimasi jumlah armada yang digunakan dan meminimasi total jarak perjalanan yang harus di tempuh. Contoh *Vehicle Routing Problem* berup rute kendaraan yang melayani 9 titik demand dapat dilihat pada Gambar 2.11



Gambar 2.11 Vehicle Routing Problem

2.2.4 *Chinese Postman Problem* (CPP)

Chinese Postman Problem adalah permasalahan merancang rute untuk melalui setiap busur paling sedikit satu kali dengan resource atau unit yang terbatas. Dalam hal ini berarah atau tidaknya busur dapat diselesaikan dengan algoritma *polynomially – bounded summer*.



Gambar 2.12 *Chinese Postman Problem*

Menurut Kwan Mei-Ko (jurnal *Chinese Mathematics*,1960) masalah *Chinese Postman Problem* pertama kali diformulasikan dalam bentuk masalah untuk menentukan sisi terpendek bagi seorang tukang pos untuk melewati semua jalan yang ada dan kembali ke tempat semula, karena untuk mendapatkan jalan yang pendek. Dalam istilah *graf* definisi CPP adalah mencari lintasan pada suatu

graf berbobot yang terhubung yang melewati semua sisi (minimal sekali) dengan jumlah bobot minimum dari suatu simpul kembali ke simpul awal.

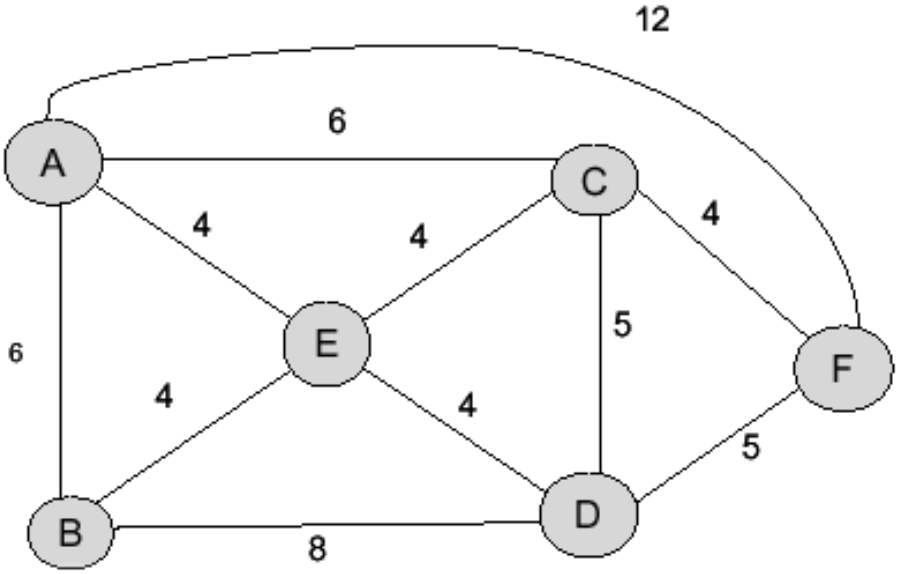
Masalah ini dapat terjadi pada *graf* berarah, tidak berarah, dan *graf* campuran. Solusi untuk *graf* tidak berarah dan berarah dapat diselesaikan secara efisien dalam waktu polinomial, sedangkan solusi untuk *graf* campuran ternyata sulit didapat (termasuk dalam kategori masalah NP-Complete). Dasar untuk menyelesaikan masalah CPP adalah keberadaan sirkuit dan lintasan Euler. Leonhard Euler berhasil menemukan bahwa, sirkuit Euler (lintasan dari satu simpul, melewati semua sisi, kembali ke simpul tersebut dengan melewati suatu titik) ada jika derajat semua *graf* adalah genap. Dan lintasan Euler (lintasan dari suatu simpul, melewati semua sisi) ada jika tepat ada dua buah simpul berderajat ganjil. ***Solusi*** Pada *graf* yang memiliki sirkuit Euler, maka lintasan manapun yang diambil hasilnya akan sama, jumlah bobot minimum adalah jumlah bobot semua sisi yang ada. Untuk *graf* yang tidak memiliki sirkuit Euler, kita perlu melewati satu atau lebih sisi dua kali. Kita tidak bisa sembarangan membuat sebuah sisi baru yang menghubungkan dua sisi yang berderajat ganjil (seperti dalam kehidupan nyata, kita juga tidak bisa sembarangan membuat jalan baru).

Untuk membentuk sirkuit Euler, kita seolah-olah membuat sisi baru dengan menggandakan sisi yang sudah ada pada lintasan yang menghubungkan dua simpul berderajat ganjil. Untuk membentuk sirkuit terpendek, maka lintasan yang digandakan pun yang terpendek. Contoh *graf* berikut tidak memiliki sirkuit Euler, karena ada dua simpul berderajat ganjil (B dan F). Kita perlu mencari jalur terpendek antara B dan F, beberapa jalur yang ada antara lain:

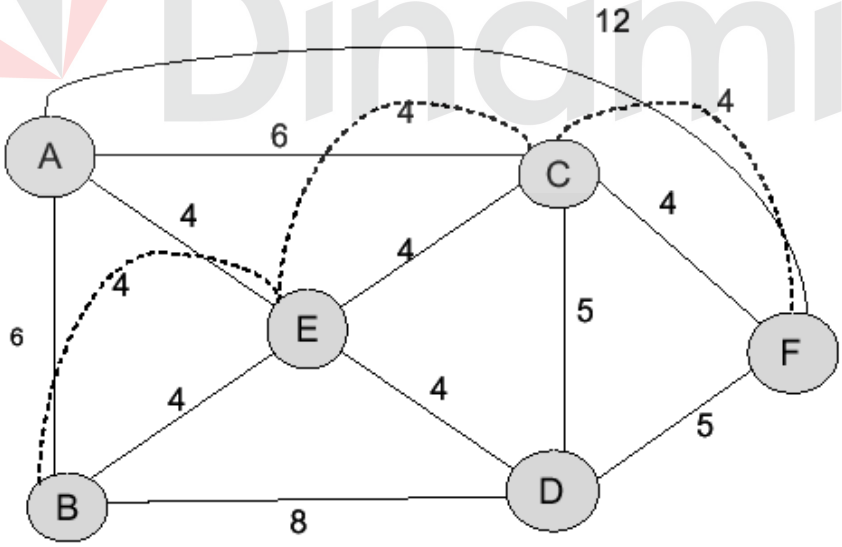
- B & E & D & F = 13
- B & A & F = 18

UNIVERSITAS Dinamika

- B & E & C & F = 12
- B & D & F = 13



Dan seterusnya yang terpendek adalah 12 (B & E & C & F), sehingga kita gandakan sisi lintasan tersebut, hasilnya :



Dengan sembarang simpul, kita pasti dapat memilih lintasan yang akan kembali ke simpul tersebut (membentuk sirkuit). Jumlah bobot minimum adalah jumlah bobot semula (62) ditambah dengan jumlah bobot jalur yang digandakan (12).

2.3 Konsep Rute Patroli

Dalam menentukan suatu rute patroli, Polwiltabes Surabaya menggunakan data detail kejadian kejahatan yang telah terkumpul. Detail kejadian kejahatan per satu bulan kemudian dilaporkan dan di olah oleh KANIT Patroli, yang kemudian oleh KANIT Patroli dimutuskan rute patroli mana yang harus dilewati selama satu bulan kedepan oleh unit patroli.

Dalam menentukan rute patroli, KANIT Patroli berdasarkan pada banyaknya kejadian kejahatan di suatu wilayah, kejadian kejahatan tersebut seperti penganiayaan berat (anirat), pencurian dengan kekerasan (curas), pencurian dengan pemberatan (curat), pencurian kendaraan bermotor (curanmor), kebakaran, penyakit masyarakat (contohnya: minuman keras, narkoba), kecelakaan, unjuk rasa, maupun gangguan-gangguan masyarakat lain yang memerlukan bantuan petugas SAMAPTA. Kejadian tersebut dapat berupa suatu kejadian kejahatan yang merupakan temuan petugas, kejadian kejahatan yang merupakan laporan oleh korban, maupun suatu kejadian kejahatan yang merupakan laporan oleh masyarakat. Detail kejadian tersebut oleh KANIT Patroli digunakan sebagai aucuan dalam menentukan rute berdasarkan prioritas banyaknya kejadian dalam suatu wilayah (Sumber Internal Polwiltabes Surabaya,2006).

2.4 *Geographic Information System* (GIS)

Sistem Informasi Geografis dikembangkan pertama kali oleh Candian. Sistem Informasi Geografis didefinisikan sebagai sistem yang diimplementasikan dengan hardware dan software untuk menyusun, menyimpan, memperbaharui/

merubah data, memanipulasi, mengolah, menganalisa dan mengkombinasikan data biasa dengan data geografi untuk menghasilkan informasi berkualitas yang berbasis geografi.

Dalam Sistem Inormasi Geografis, dunia nyata (*real world*) dijelakan dengan menggunakan peta digital (data geografi), yang menjelaskan lokasi suatu tempat dan data atribut ini berupa sebuah tabel. Kedua tipe data yang berbeda yaitu data geografi dan data atribut yang berbeda disimpan dalam suatu database. Dengan cara tradisional, data geografi ditampilkan dari peta dengan menggunakan simbol, garis dan warna. Banyak peta mempunyai *legend* yang berisi dan menjelaskan elemen-elemen data geometrik, misalnya : garis hitam tebal untuk jalan besar, gari hitam tipis untuk jalan kecil, dan masih banyak lagi yang lainnya. Peta dapat menampilkan dan menyimpan data geografi dengan efektif, tetapi peta juga punya keterbatasan, yaitu: informasi yang disimpan banyak yang ditampilkan didalam peta dan biasanya untuk keperluan tertentu. Perubahan pada peta jarang sekali terjadi karena hampir selalu berbeda diantara kebutuhan pemakai yang banyak. Sedangkan user, dalam melakukan analisanya terhadap informasi geografis yang ada juga membutuhkan perangkat keras komputer dan perangkat lunak pendukung.

Menurut Eddy Prahasta(2001) Sistem Informasi Geografi adalah sebuah rangkaian sistem yang memanfaatkan teknologi digital untuk melakukan analisis spasial, Sistem Informasi Geografi bekerja dengan mneggunakan referensi data dengan bentuk geografi. Bentuk data yang digunakan dalam Sistem Informasi Geografis adalah bentuk data spasial. Data spasial berupa *layer* yang dapat disusun menjadi satu kesatuan.

UNIVERSITAS Dinamika

Data spasial terdiri dari :

1. Peta

Spasial data menyimpan data dalam bentuk gambar secara fisik, sehingga diperlukan proses scanning gambar peta. Tipe yang ada berupa titik (tidak nenpuyai linier), garis (tidak memiliki dimensi), area/*polygons* (bersifat 2 dimensi dibatasi oleh garis yang mengelilingi area terebut).

2. Data Vektor

Posisi dari peta yang terdiri dari X, Y, Z. Format vektor berbentuk garis yang dapat menggambarkan suatu lokasi dan batas wilayah secara tepat.

3. Data Raster

Merupakan gambar peta secara general seperti *pixel* yang disimpan dalam matrix grid.

Sistem Informasi Geografi biasanya digunakan untuk masalah dalam skala daerah yang luas, dan banyak diaplikasikan oleh semua bidang kegiatan usaha dengan permasalahan yang luas.

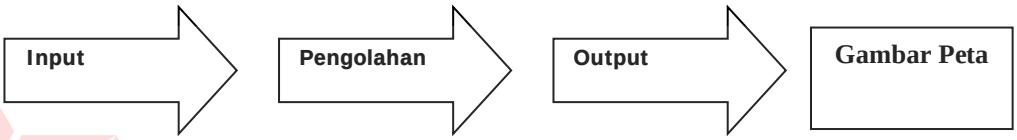
Sedangkan tujuan utama dari GIS adalah sebagai tools atau alat untuk membantu memecahkan persoalan-persoalan kehidupan yang berkaitan dengan lokasi atau ruang. Fungsi dasar peta (GIS) adalah menempatkan sesuatu sesuai keberadaan atau kejadiannya di muka bumi. Beberapa keuntungan lain yang didapat dari GIS antara lain adalah dengan GIS terutama jika menggunakan komputer maka perubahan yang terjadi bisa digambarkan dengan cepat jika dibandingkan dengan cara manual yang harus menggambarkan segala sesuatunya dari awal misalnya menggambar peta desa lagi dan kemudian menambahkan informasi baru pada peta tersebut. Dengan GIS, sejak awal peta desa menjadi

UNIVERSITAS Dinamika

obyek tersendiri yang terpisah dari obyek lainnya misal lokasi satu rumah, di mana bisa dipakai lagi untuk keperluan lain. GIS mempunyai fungsi penyimpanan yang terstruktur sesuai keinginan si pemakai. Sehingga dengan begitu beberapa hal yang tidak perlu (misal penggambaran manual dan pengulangan) menjadi tidak selalu diperlukan, sehingga pekerjaan bisa lebih sederhana dan efektif.

2.4.1 Proses GIS

Sebagai sebuah sistem maka tentu saja alur kerja GIS sangat penting untuk diketahui, secara sederhana prosesnya pada Gambar 2.13:



Gambar 2.13 Alur Proses GIS

Dimana :

Input : proses pemasukan data-data yang diperlukan (peta-peta, data-data lainnya)

Pengolahan : data-data tersebut diproses, diseleksi, “dimanipulasi” sesuai dengan maksud dan kebutuhan, dalam proses ini juga bisa dilakukan analisa informasi.

Output : keluaran hasil proses berupa peta-peta Geografis atau informasi geografis bisa juga ditandai dengan data-data seperti koordinat.

Dalam proses ini dapat diketahui bahwa alur kerja GIS menghasilkan informasi yang berbentuk peta.

UNIVERSITAS Dinamika

BAB III

METODE PENELITIAN

Dalam memecahkan masalah dalam tugas akhir ini ditempuh langkah: indentifikasi masalah, perumusan masalah, batasan masalah, tujuan, studi lapangan dan studi pustaka yang di lanjutkan dengan perancangan model.

Algoritma pencarian dengan menggunakan model reproduksi dinamakan *genetic algorithm* (GA). Proses reproduksi dilakukan dengan jalan merekombinasi 2 (dua) individu menjadi individu lain. Proses GA dalam mencari suatu nilai optimal adalah:

1. Menentukan model representasi genetika dari masalah yang dihadapi yang berupa model dari sistem buatan, representasi individu, pembatasan, fungsi evaluasi.
2. Pembangkitan generasi awal dengan memproses database.
3. Menentukan nilai *fitness* (nilai ketepatan) dari tiap individu berdasarkan struktur gennya. Nilai *fitness* ini dijadikan sebagai ukuran apakah individu tersebut sudah optimal atau belum.
4. Pemilihan individu dengan nilai *fitness* terbaik untuk dijadikan induk dalam menghasilkan individu baru.
5. Proses reproduksi yang terdiri dari c*rossover* dan mutasi proses reproduksi ini dihasilkan individu baru.

Dengan melakukan proses di atas secara berulang-ulang, diharapkan induk yang baik akan menghasilkan suatu generasi dengan individu yang lebih baik.

UNIVERSITAS Dinamika

3.1 Identifikasi Masalah

Di dalam POLWILTABES Surabaya terdapat suatu kesatuan yang mempunya tugas untuk berpatroli yaitu SAMAPTA. SAMAPTA memiliki beberapa unit patroli yang bertugas setiap hari. Unit patroli berupa mobil patroli yang setiap mobilnya berisi 2 atau lebih tergantang kapasitas mobil petugas anggota SAMAPTA. Jalan protokol adalah rute partoli utama yang harus dilakukan, karena jalan protokol adalah akses jantung kota, dimana tidak hanya kendaraan yang dari dalam kota tetapi dari luar kota juga.. Melihat banyaknya kendaraan yang melewati jalan protokol, dan dikhawatirkan banyak terjadi tindak kejahatan maupun kecelakaan, maka setidaknya ada 1 unit patroli yang berpatroli setiap waktu.

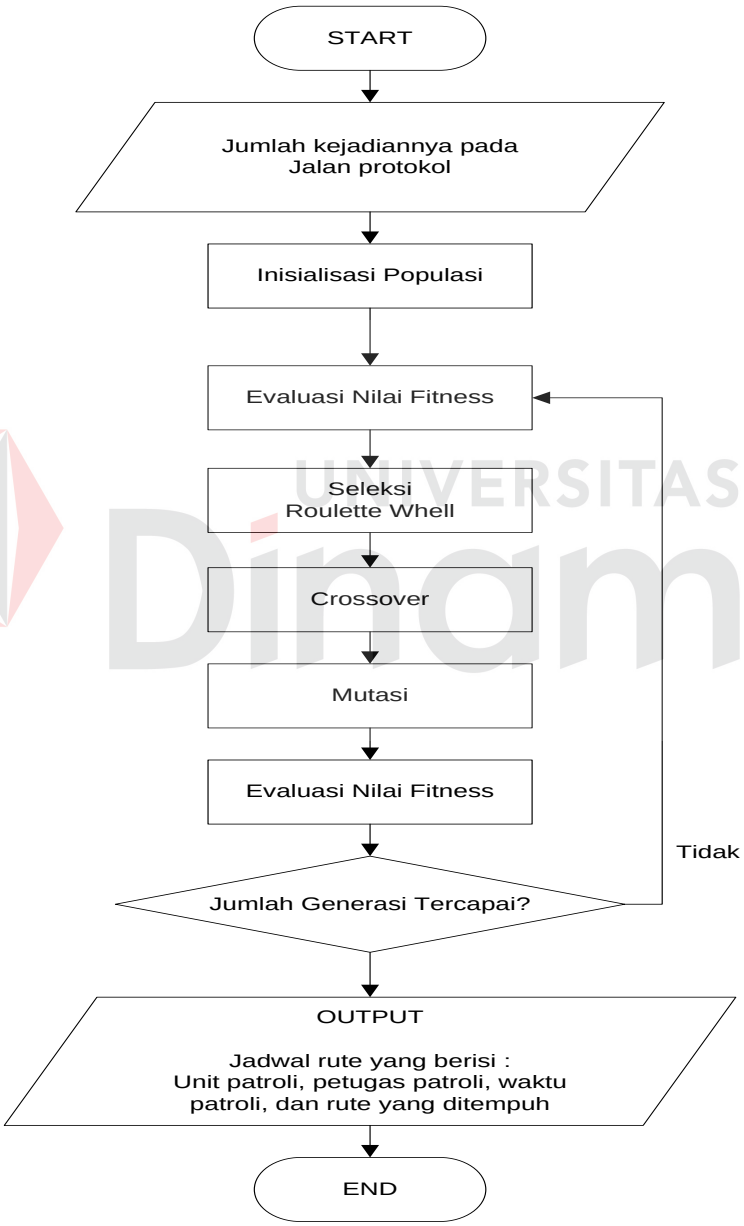
Permasalahan yang timbul adalah dalam pembuatan rute patroli yang efektif berdasarkan banyaknya kejadian dapat dikatakan secara manual, sebab penentuan memang hanya masih mengandalkan data banyaknya kejadian tiap bulannya.

Dari analisis diatas, untuk dapat menyelesaikan permasalahan penentuan rute patroli, akan diterapkan dengan pendekatan metode algoritma genetika dengan mengabaikan sistem sebelumnya (manual) yang digunakan oleh SAMAPTA POLWILTABES SURABAYA.

Data banyaknya kejadian dalam setiap ruas jalan akan dicatat, kemudian diurutkan berdasarkan bulan. Jalan yang diambil adalah jalana protokol, karena jalan protokol dianggap jalan vital, karena jalan protokol berada di tengah kota dan menghubungkan antara ujung kota dengan perbatasan kota. Setiap data yang dicatat akan dihitung banyaknya kejadian pada jalan protokol dimana banyakanya

kejadian minimal tiga (3) baik tindak kejahatan maupun kecelakaan dinyatakan sebagai jalan yang banyak terjadi tindak kejahatan.

Data yang dipakai adalah data banyaknya kejadian yang meliputi jenis kejadian, waktu kejadian, unit patroli yang menangani, petugas yang menangani, jenis laporan, tempat kejadian, identitas pelaku, identitas korban, dan keterangan.



Gambar 3.1. Diagram Alir Penyelesaian Rute Patroli

3.2 Perancangan Algoritma Genetika

Pada tahap desain sistem ini akan dilakukan penerapan proses algoritma genetika dari penentuan rute patroli dengan mengimplementasikan data yang didapat dari masing-masing ruas jalan selama beberapa bulan dalam masing–masing proses yang ada pada algoritma genetika.

3.2.1 Pengkodean (Representasi Kromosom)

Pengkodean adalah proses menterjemahkan suatu obyek permasalahan ke dalam metode algoritma genetika. Dalam permasalahan penentuan rute ini kromosom dipresentasikan dalam bentuk *list*, yang mempresentasikan jarak antar jalan yang akan dikunjungi. Setiap kromosom harus memenuhi batasan–batasan yang sudah dijelaskan diatas. Dengan contoh terdapat 6 tujuan dimana setiap ruas jalan akan diwakili dengan kode jalan, maka dapat dipresentasikan sebagai berikut

Jlp001 = jl. Perak Barat

Jlp002 = jl.Perak Timur

Jlp005 = jl. Veteran

Jlp007 = jl. Tunjungan

Jlp004 = jl. Indrapura

Jlp009 = jl. Baluran

Jlp001	**Jlp002**	**Jlp005**	**Jlp007**	**Jlp004**	**Jlp009**

Gambar 3.2 Kromosom Rute Patroli

UNIVERSITAS Dinamika

3.2.2 Inisialisasi Populasi

Proses inisialisasi adalah pembentukan populasi awal yang diperoleh dengan cara rekombinasi kromosom sebanyak ukuran populasi yang ditentukan. Dalam permasalahan ini proses rekombinasi kromosom

Dimana populasi menyatakan banyaknya kromosom yang terbentuk dalam populasi dalam satu generasi. Apabila ukuran populasinya kecil maka algoritma genetika memiliki kemungkinan yang sedikit untuk melakukan proses kromosom dan hanya sebagian kecil dari ruang pencarian yang dapat digunakan. Sedangkan jika terlalu banyak kromosom maka proses dalam algoritma genetika menjadi lambat. Ukuran populasi yang baik adalah berkisar antara 20 sampai 30, walaupun untuk beberapa kasus tertentu ukuran populasi antara 50 sampai 100 adalah yang lebih baik. Dalam proses ini, akan dilakukan rekombinasi kromosom sampai diperoleh satu populasi. Dalam sistem setiap populasi akan daibatasi sampai dengan 10000 kromosom, mengingat jika banyaknya jalan yang terjadi kejadian akan menimbulkan *overload memory*.

3.2.3 Fungsi Fitness

Setelah mendapatkan satu populasi awal dari algoritma inisialisasi populasi di atas, maka selanjutnya tiap-tiap kromosom akan di evaluasi. Populasi baru dari perhitungan batasan – batasan fungsi *fitness* akan dievalusi yang gunanya adalah menghitung total waktu yang ditempuh oleh setiap rute, dan pendefinisiannya adalah sebagai berikut :

$$Eval(K_i)\sum_{r}^{n} Tot\ r/\ v \quad \text{.....} \quad (4)$$

dimana, $r = 1,2,3,...,n$

n = 1,2,3,...max pop_size

v = kecepatan kendaraan (Km / jam)

Untuk rumusan fungsi *fitness* (4) menunjukkan jumlah total waktu tempuh dari suatu jalan. Parameter yang digunakan adalah : r sebagai jalan, K_i sebagai kromosom ke – *i*, dan v sebagai kecepatan mobil dalam menempuh rute yang dinyatakan dalam bentuk kromosom.

Dari contoh diatas maka dari hasil random sesuai dengan jumlah populasi maka didapat 5 kromosom, dari tiap kromosom tersebut maka dihitung jarak antara kode jalan satu ke kode jalan lain, menurut urutannya kemudian dibagi dengan kecepatan (v) misalkan kecepatan mobil 40 Km/jam .

Jlp001	Jlp005	Jlp004	Jlp009	Jlp007	Jlp002

Jlp001 ke Jlp005 = 22 Km Jlp004 ke Jlp009 = 35 Km
Jlp005 ke Jlp004 = 13 Km Jlp009 ke Jlp007 = 37 Km
Jlp007 ke Jlp002 = 37 Km
(22 Km + 13 Km + 35 Km + 37 Km + 37 Km) / 40 = 3,6 Jam
Eval (K_2) = 3,6 jam

Jlp005	Jlp009	Jlp004	Jlp001	Jlp007	Jlp002

Eval (K_3) = 3,525 jam

Jlp009	Jlp005	Jlp007	Jlp001	Jlp004	Jlp002

Eval (K_4) = 3,1 jam

Jlp002	Jlp007	Jlp004	Jlp009	Jlp001	Jlp005

Eval (K_5) = 3,275 jam

Jlp004	Jlp005	Jlp001	Jlp009	Jlp007	Jlp002

Eval (K_6) = 3,75 jam

Gambar 3.3 Nilai *Fitness* Kromosom Rute

UNIVERSITAS Dinamika

3.2.4 Seleksi

Proses seleksi bertanggung jawab untuk melakukan pemilihan terhadap individu (parent) yang hendak diikutkan dalam proses reproduksi. Pada permasalahan ini akan digunakan metode seleksi *roulette whell*. Dalam penerapan *roulette whell*, terdapat dua bagian user fungsi yaitu fungsi inisialisasi *roulette whell* dan fungsi penentuan *roulette whell*. Penentuan inisialisasi *roulette whell* bertujuan untuk mendapatkan probabilitas seleksi, dan probabilitas ini didapat berdasarkan nilai *fitness* tiap individu.

Adapun langkah-langkah yang diperlukan dalam inisilaisasi *roulette whell* adalah sebagai berikut :

1. Menjumlahkan nilai *fitness* dari semua individu yang terdapat dalam populasi.

$$Total\ Fitness = \sum_{1}^{Pop\ size} Eval\ (Ki)$$

2. Menghitung probabilitas terpilihnya setiap individu (P[i]) dengan membagi nilai *fitness* masing-masing individu dengan total nilai *fitness* populasi.

$$P[i] = Evel(Ki)\ /\ Total\ Fitness$$

3. Menghitung probabilitas kumulatif untuk setiap individu dalam populasi (Q[i]).

$$Q[i] = \sum_{j=i+1}^{i=pop\ size} P[j]$$

Tidak semua individu yang terdapat pada populasi melakukan operasi genetik. Hal itu disebabkan tidak semua individu lulus dalam seleksi. Untuk

masalah seleksi, *Roulette whell* dipergunakan dalam program untuk menentukan individu-individu mana yang akan dipakai dalam operasi genetik. *Roulette whell* dilakukan setelah didapatkan probabilitas seleksi.

Langkah-langkah dari *Roulette whell* adalah :

1. Membangkitkan bilangan acak (R) bertipe real dengan range [0..1] sejumlah individu dalam populasi.
2. Apabila Q[i] £ R, dan Q[i+1]>R maka pilih kromosom ke (i+1) sebagai kandidat induk.

Dari masalah di atas maka dalam langkah inisialisai *roulette whell* didapatkan hasil seperti dibawah ini :

1. Menjumlahkan nilai *fitness*

   $\text{Eval}(K_1) + \text{Eval}(K_2) + \text{Eval}(K_3) + \text{Eval}\ (K_4) + \text{Eval}\ (K_5) =$

   3,6 + 3,524 + 3,1 + 3,275 + 3,75 = 17,25 jam
2. Menghitung probabilitas setiap individu

   P[1] = 3,600 / 17,25 = 0,209

   P[2] = 3,524 / 17,25 = 0,204

   P[3] = 3,100 / 17,25 = 0,180

   P[4] = 3,275 / 17,25 = 0,190

   P[5] = 3,750 / 17,25 = 0,217
3. Menghitung probabilitas kumulatif

   Q[1] = 0,209

   Q[2] = 0,209 + 0,204 = 0,413

   Q[3] = 0,209 + 0,204 + 0,180 = 0,593

   Q[4] = 0,209 + 0,204 + 0,180 + 0,190 = 0,783

UNIVERSITAS Dinamika

Q[5] = 0,209 + 0,204 + 0,179 + 0,189 + 0,217 =1,000

Dan Langkah dari *roulette whell*-nya adalah :

1. Membangkitkan bilangan acak R dengan range [0..1] sebanyak 5 atau sejumlah dengan individu dalam populasi.

   Misalnya didapatkan dari hasil random sebagai berikut :

   R[1] = 0,281 R[3] = 0,473 R[5] = 0,665
   R[2] = 0,882 R[4] = 0,784

2. Apabila Q[i] £ R, dan Q[i+1]>R maka pilih kromosom ke (i+1) sebagai kandidat induk.

R[1]	£	Q[1]
0,281	£	0,209
	Dan	
Q[2]	>	R[1]
0,413	>	0,281

Jadi Q[2] menjadi kromosom baru yang pertama.

R[2]	£	Q[4]
0,882	£	0,783
	Dan	
Q[5]	>	R[2]
1	>	0,882

Jadi Q[5] menjadi kromosom baru yang kedua.

R[3]	£	Q[2]
0,473	£	0,413
	Dan	
Q[3]	>	R[3]
0,593	>	0,473

Jadi Q[3] menjadi kromosom baru yang ketiga.

R[4]	£	Q[4]
0,784	£	0,783
	Dan	
Q[5]	>	R[4]
1,000	>	0,784

Jadi Q[5] menjadi kromosom baru yang keempat.

R[5]	£	Q[3]
0,665	£	0,593
	Dan	

UNIVERSITAS Dinamika

Q[4]	>	R[5]
0,783	>	0,665

Jadi Q[4] menjadi kromosom baru yang kelima.

Kromosom yang telah mengalami seleksi :

K[1] =	**K[2] =**	**Jlp005**	**Jlp009**	**Jlp004**	**Jlp001**	**Jlp007**	**Jlp002**
K[2] =	**K[5] =**	**Jlp004**	**Jlp005**	**Jlp001**	**Jlp009**	**Jlp007**	**Jlp002**
K[3] =	**K[3] =**	**Jlp009**	**Jlp005**	**Jlp007**	**Jlp001**	**Jlp004**	**Jlp002**
K[4] =	**K[5] =**	**Jlp004**	**Jlp005**	**Jlp001**	**Jlp009**	**Jlp007**	**Jlp002**
K[5] =	**K[4] =**	**Jlp002**	**Jlp007**	**Jlp004**	**Jlp009**	**Jlp001**	**Jlp005**

3.2.5. Operator Genetika

Setelah dilakukannya seleksi maka dari parent – parent yang terpilih akan dilakukan proses kerja yang ada pada algoritma genetika, yaitu diperlukan operator–operator genetika. Pada permasalahan optimasi rute patroli ini memerlukan dua jenis operator genetika yaitu *crossover* dan mutasi.

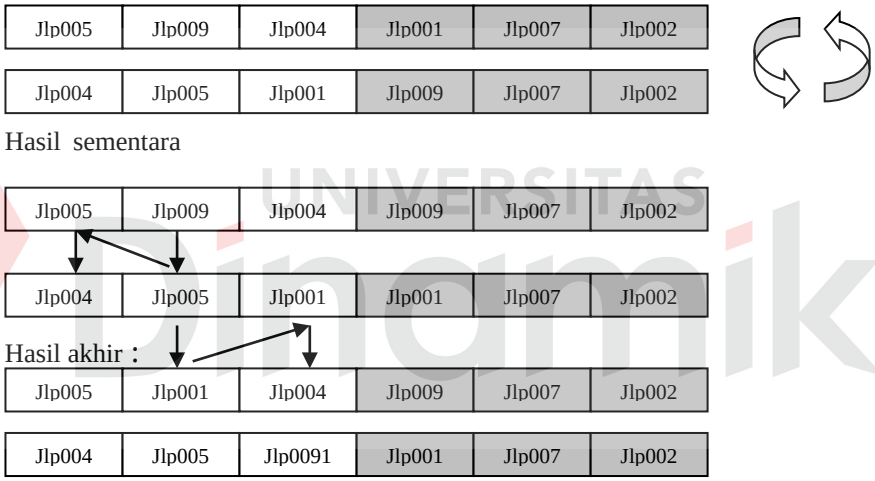
A. Crossover

Pada permasalahan optimasi rute patroli akan menggunakan jenis *one point crossover* sebab menggunakan permutation encoding yang dalam pembentukan kromosom dilakukan dengan mengambil kode jalan yang sesuai dengan permasalahan nama jalan yang banyak terjadi kejadian. One point *crossover* dilakukan dengan cara memilih secara acak posisi dalam kromosom induk (parent) pertama kemudian saling menukar gen dengan induk kedua dengan posisi yang sama, sebab dalam pembentukan kromosom terbentuk dari permutation encoding dimana permutation encoding digunakan untuk permasalahan proses pengurutan. Dan jumlah kromosom yang mengalami *crossover* dipengaruhi oleh parameter *crossover* (probabilitas *crossover*) yang telah ditentukan yaitu dengan cara menghitung Pc * uk_pop.

UNIVERSITAS Dinamika

Untuk mengetahui kromosom-kromosom mana yang akan di *crossover* kita terlebih dahulu membangkitkan bilangan random antara [0....1], sejumlah pop size. Kemudian kita bandingkan antara bilangan random dengan pc, jika nilainya lebih kecil dari pc maka kromosom tersebut terkena *crossover*.

Sedangkan untuk posisi pemotongan diinputkan probabilitas *crossover* (Pc) adalah 50%(0,5). Maka jumlah posisi pemotongan adalah 0,5*5 = 2,75 (3) gen. Dalam permasalahan ini dipakai one cut *crossover*, dan setelah didapat kromosom induk (parent) berdasarkan nilai random serta posisi pemotongan didapat secara acak pula maka proses *crossover*nya adalah sebagai berikut :



Gambar 3.4 Prosess *Crossover*

Setelah kedua individu selesai di*crossover*, dimungkinkan terjadi kekacauan dalam sebuah kromosom, yaitu terjadi perulangan dari jalan yang dilewati. Karena itu setelah terjadi *crossover*, maka dilakukan perbaikan pada masing-masing kromosom. Setiap tujuan yang kembar pada kromosom 1 (dari bagian yang tidak di tukar) dibandingkan dengan tujuan pada posisi yang sama pada kromosom 2, bila tujuan pada kromosom 2 ini juga kembar, maka kedua tujuan akan ditukar, bila tidak maka dicari tujuan pada kromosom 2 yang

posisinya sama dengan posisi tujuan pada kromosom 2 tadi, tetapi pada kromosom 1. Berikut adalah lagkah dari proses crosover :

Step 1 = [buat random sebanyak jumlah kromosom]
Step 2 = [tentukan kromosom yang akan di crossover]
Step 3 = [tentukan panjang kromosom yang akan di crossover]
Step 4 = [silangkan panjang dari kromosom parent dengan kromosom child]
Step 5 = [cek isi kromosom parent apakah ada yang sama]
Step 6 = [cek isi child apakah ada yang sama]
Step 7 = [jika ada maka silangkan isi dari kromosom parent dan kromosom child]

Berikut adalah kromosom-kromosom setelah di *crossover* :

K[2]' =	Jlp005	Jlp001	Jlp004	Jlp009	Jlp007	Jlp002
K[5]' =	Jlp004	Jlp005	Jlp009	Jlp001	Jlp007	Jlp002
K[3] =	Jlp009	Jlp005	Jlp007	Jlp001	Jlp004	Jlp002
K[5] =	Jlp004	Jlp005	Jlp001	Jlp009	Jlp007	Jlp002
K[4] =	Jlp002	Jlp007	Jlp004	Jlp009	Jlp001	Jlp005

Gambar 3.5 Kromosom Hasil *Crossover*

B. Mutasi

Dalam permasalahan ini digunakan mutasi inversi, sebab menggunakan permutation encoding dimana mutasi yang sesuai dengan optimasi. Proses mutasi dilakukan dengan cara mengganti satu sub kromosom (gen) yang dipilih secara acak dengan suatu nilai baru. Untuk memilih posisi gen yang mengalami mutasi prosesnya adalah kita hitung dengan Pm * uk_pop * N. Dalam masalah optimasi misalnya ditentukan probabilitas mutasinya (Pm) adalah 5%(0,05) , maka didapat bahwa jumlah gen yang akan dimutasi adalah 0,05*6*5 = 1,5 (dibulatkan menjadi 2) gen. Posisi gen yang mengalami mutasi ditentukan secara acak.dengan cara menentukan bilangan acaka antara 1 sampai panjang semua gen L -1 [1.....30]. Misal bilangan acak yang terbangkit adalah 2, 5 dan 10,dan proses mutasi pada optimasi rute ini adalah sebagai berikut:

UNIVERSITAS Dinamika

K[2]"=	Jlp005	**Jlp007**	Jlp004	Jlp009	**Jlp001**	Jlp002
K[5] =	Jlp004	Jlp005	Jlp009	**Jlp007**	Jlp001	Jlp002
K[3] =	Jlp009	Jlp005	Jlp007	Jlp001	Jlp004	Jlp002
K[5] =	Jlp004	Jlp005	Jlp001	Jlp009	Jlp007	Jlp002
K[4] =	Jlp002	Jlp007	Jlp004	Jlp009	Jlp001	Jlp005

Gambar 3.6 Proses Mutasi

Setelah proses mutasi selesai, maka dalam permasalahan optimasilasi rute ini telah dilakukan satu iterasi dalam algoritma genetika atau disebut dengan satu generasi.

Setelah menghasilkan kromosom akhir setlah mutasi maka dilakukan pengecekan nilai *fitness* untuk nebdapatkan kromosom yang paling baik sebagai solusi dari rute patroli

K[2]"=	Jlp005	**Jlp007**	Jlp004	Jlp009	**Jlp001**	Jlp002
K[5] =	Jlp004	Jlp005	Jlp009	**Jlp007**	Jlp001	Jlp002
K[3] =	Jlp009	Jlp005	Jlp007	Jlp001	Jlp004	Jlp002
K[5] =	Jlp004	Jlp005	Jlp001	Jlp009	Jlp007	Jlp002
K[4] =	Jlp002	Jlp007	Jlp004	Jlp009	Jlp001	Jlp005

K[2]" = 2,275 jam

K[5] = 2,.975 jam

K[3] = 3,1 jam

K[5] = 3,75 jam

K[4] = 3,275 jam

Karena dalam masalah rute diharapkan dapat memberikan solusi yang paling optimal yaitu dengan menempuh waktu yang paling sedikit maka yang memenuhi syarat adalah kromosom **K[2]"** karena memiliki waktu yang paling sedikit yaitu **2,275 jam.**

K[2]"=	Jlp005	**Jlp007**	Jlp004	Jlp009	**Jlp001**	Jlp002

3.2.6 Kriteria Berhenti

Dalam sistem ini kriteria berhenti menggunakan banyaknya jumlah generasi yang diinputkan, setelah melakukan perancangan sistem maka hasil rancangan tersebut akan diuji untuk mengetahui apakah rancangan yang telah dibuat sudah memenuhi kebutuhan sistem berdasarkan pada inputan yang menunjukkan proses berhenti. Pada optimalisasi rute patroli ini, proses akan berhenti setelah mencapai generasi ke – N (sesuai dengan inputan pada generasi).

3.2.7 Breeder GA (Pelestarian Kromosom)

Pada proses optimasilasi rute ini kromosom-kromosom terbaik yang dihasilkan akan dipertahankan, sehingga dapat lolos pada proses seleksi dan dapat digunakan dalam generasi berikutnya.

3.3 GIS (Goegrafis Information System)

Setelah mendapatkan hasil dari kromosom terbaik maka kromosom tersebut akan di tampilkan ke dalam sebuah peta.

Contohnya dari hasil proses GA maka didapatkan kromosom yang terbaik adalah :

K[2]"=	Jlp005	**Jlp007**	Jlp004	Jlp009	**Jlp001**	Jlp002

Dimana kode jalan mewakili nama jalan yaitu:

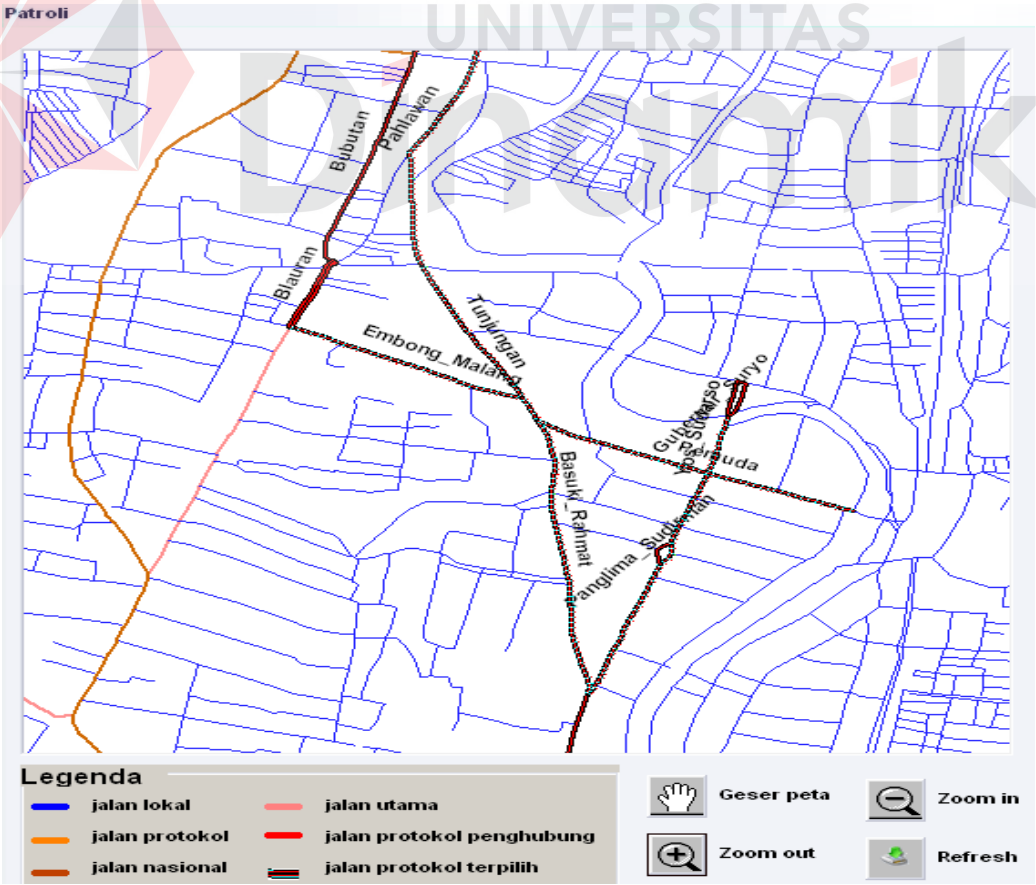
Jlp005 = Jl.Veteran
Jlp007 = Jl.Tunjungan
Jlp004 = Jl.Indrapura
Jlp009 = Jl.Blauran
Jlp001 = Jl.Perak Barat
Jlp002 = Jl.Perak Timur

UNIVERSITAS Dinamika

Kemudian dari kode jalan tersebut Form MapViewForm akan mengambil database idjalan kedalam database map info.

Perintah untuk menampilkan nama jalan protokol berdasarkan idJalan yang merupakan kromosom trbaik dari proses seleksi, crossover, dan mutasi. Kemudian data dari formgrafik akan di tampilkan ke dalam MapViewForm adalah sebagai berikut:

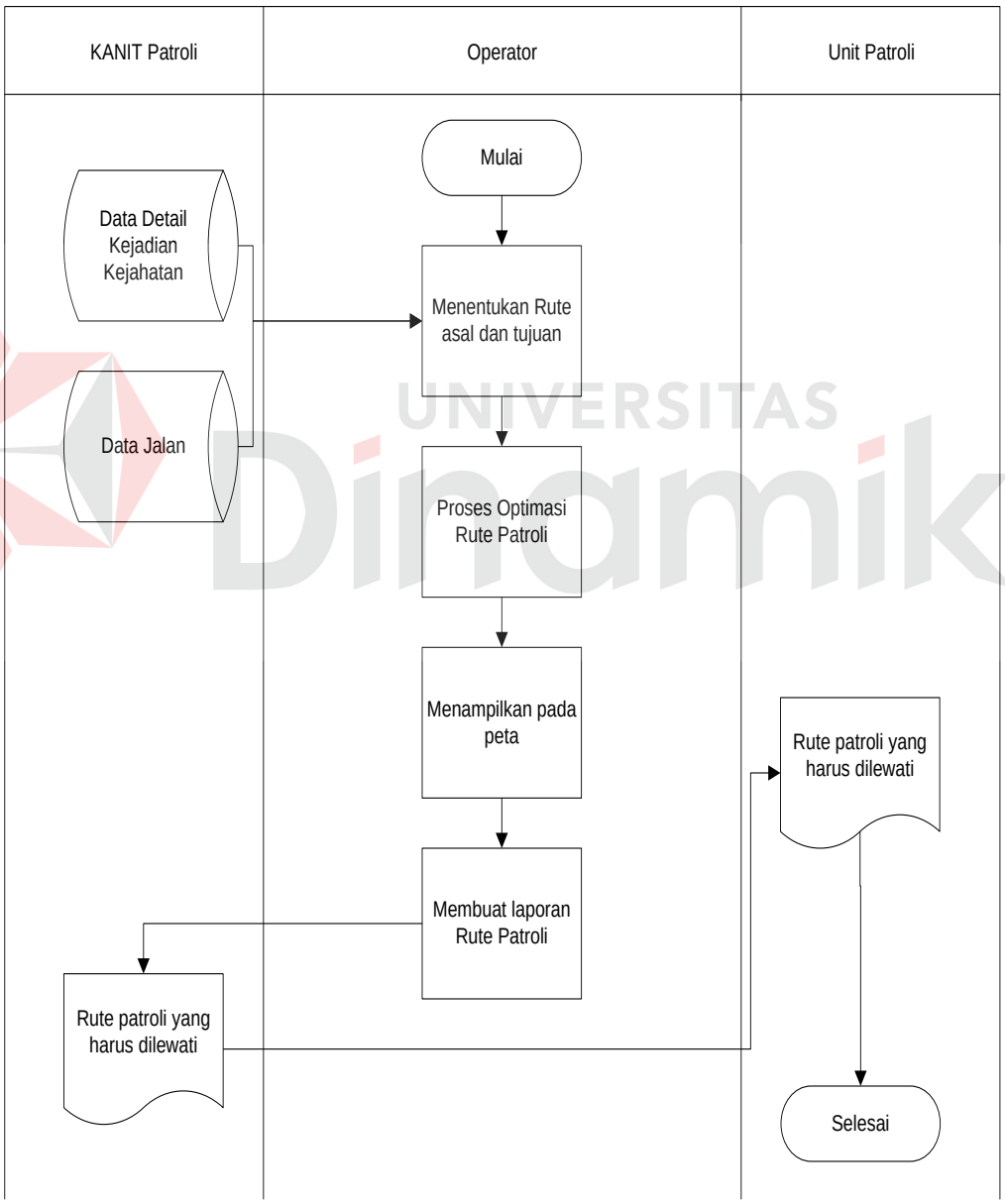
```
FormGrafik
id_jalan = Right(kode_jl, phasil)
query = "Select * from jl_protokol where " + id_jalan
Mapviewform.mapinfo.do query
Mapviewform.mapinfo.eval ("jl_protokol.id")

MapViewForm
mapinfo.do "select * from jl_protokol order by nama_jalan into TEMP_WORK"
nRows=Val(mapinfo.eval("tableinfo(TEMP_WORK,"+Str$(TAB_INFO_NROWS)
+ ")"))
   mapinfo.do "close table temp_work"
   mapinfo.do "close table selection"
```



Gambar 3.7 Contoh Hasil Rute

Pada Gambar 3.7 Contoh Hasil Rute, garis yang berkelip adalah Jalan protokol yang terpilih untuk dilewati, sedangkan garis merah adalah jalan protokol penghubung, meskipun tidak terpilih tetapi tetap di tampilkan karena untuk menjadikan rute tersebut terhubung memerlukan jalan lain yang memungkinkan untuk dilewati.

3.4 Desain Sistem

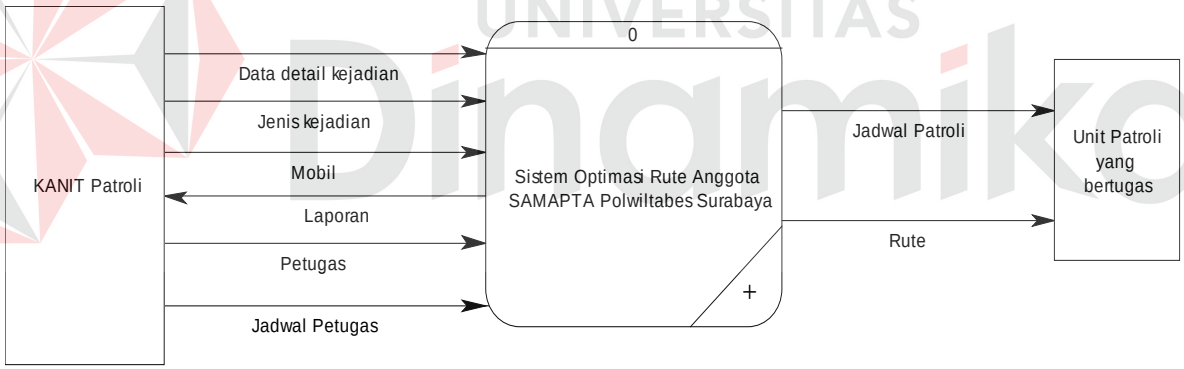


Gambar 3.8 Proses Optimasi Rute Patroli

Berdasarkan diagram alur diatas, proses algoritma genetika dimulai dengan mengambil data kejadian pada tiap ruas jalan, selanjutnya dilakukan proses optimasi menggonakan Algoritma Genetika, hasil dari optimasi akan ditampilkan ke dalam bentuk peta seperti pada gambar 3.7 Output dari proses ini akan menghasilkan suatu jadwal rute patroli yang optimal. Dimana hasil rute patroli akan dibuat laporan untuk diberikan kepada unit yang bertugas.

3.4.1 Context Diagram

Context diagram menggambarkan sistem pertama kali secara garis besar dari semua hubungan antara sistem pengendalian bahan baku ini dengan lingkungan sekitarnya. Pembentukan Context diagram dapat dilihat pada pada Gambar 3.9.

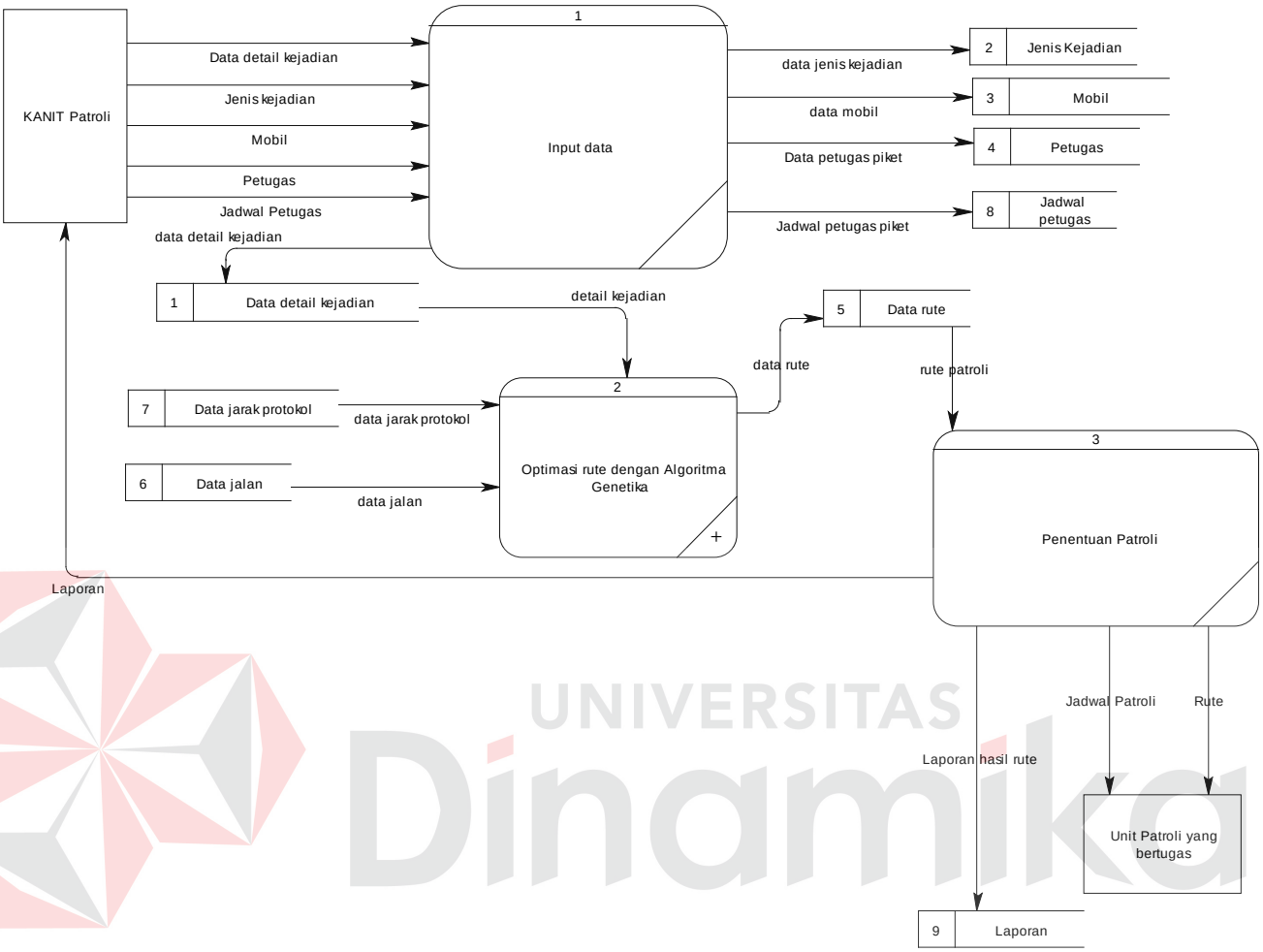


Gambar 3.9 Context Diagram Optimasi Rute Patroli

3.4.2 Data Flow Diagram (DFD)

Representasi dari sebuah sistem secara grafis yang digambarkan dengan sejumlah simbol tertentu untuk menunjukkan perpindahan data dalam proses suatu sistem.

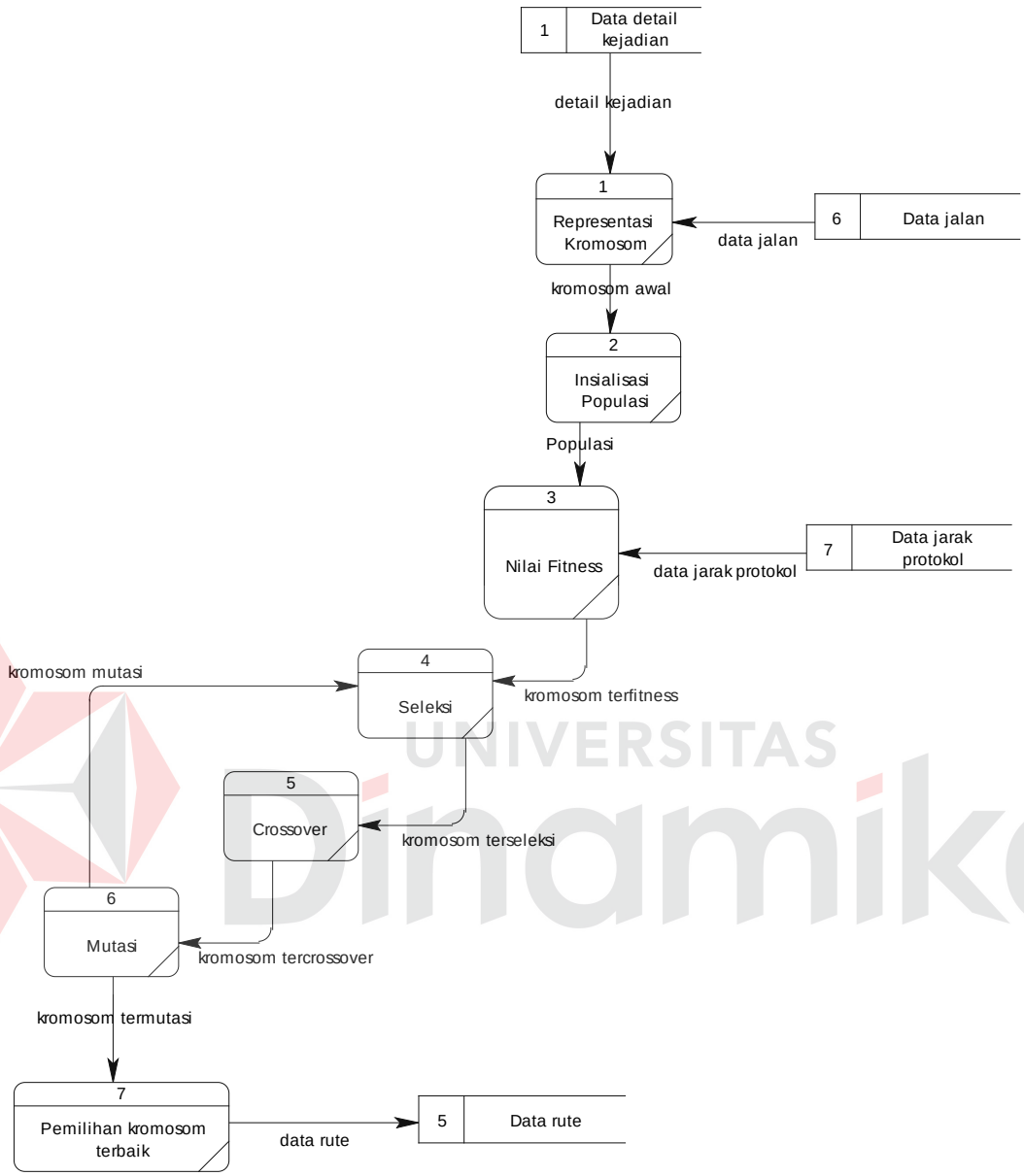
A. DFD Level 0



Gambar 3.10 DFD Level 0

Sedangkan DFD pada level dibawahnya dari masing-masing proses di DFD level 1 penentuan rute patroli dengan menggunakan Algoritma Genetika dapat dilihat pada gambar 3.10.

B. DFD Level 1 Proses 1

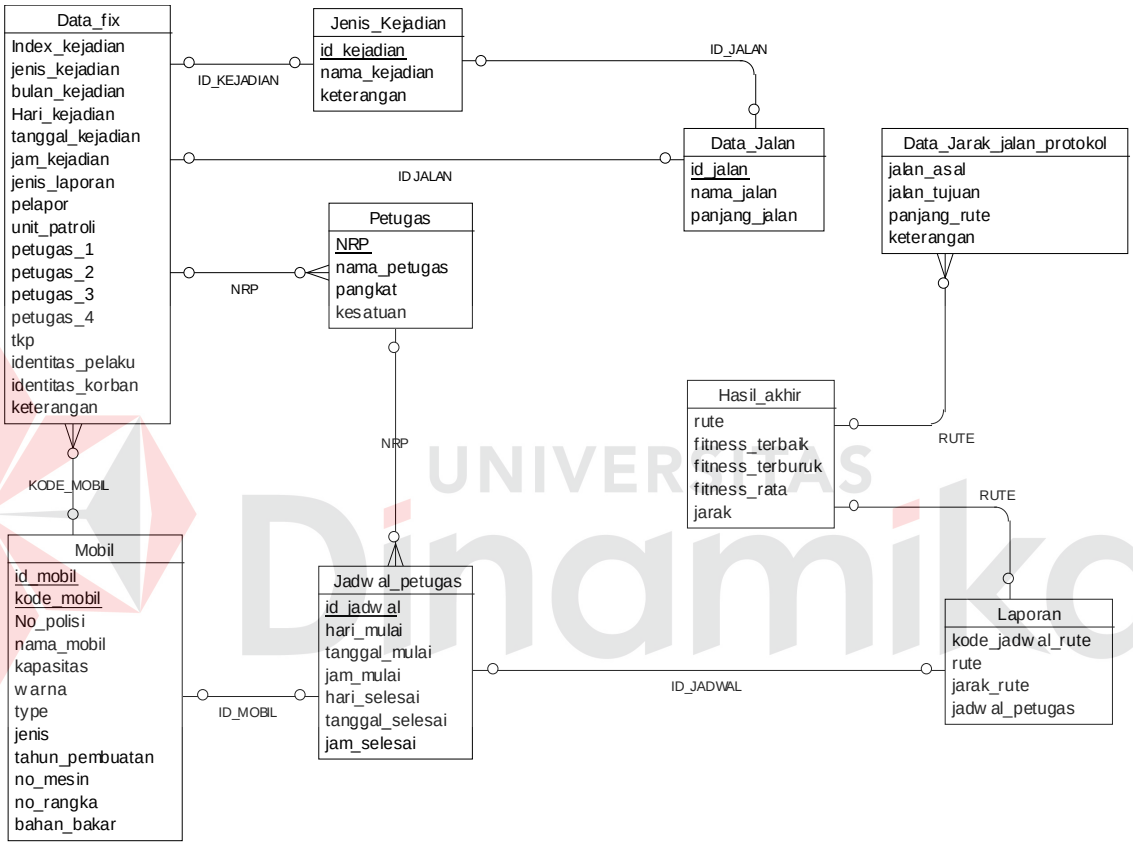


Gambar 3.11 DFD Level 1 Proses 1

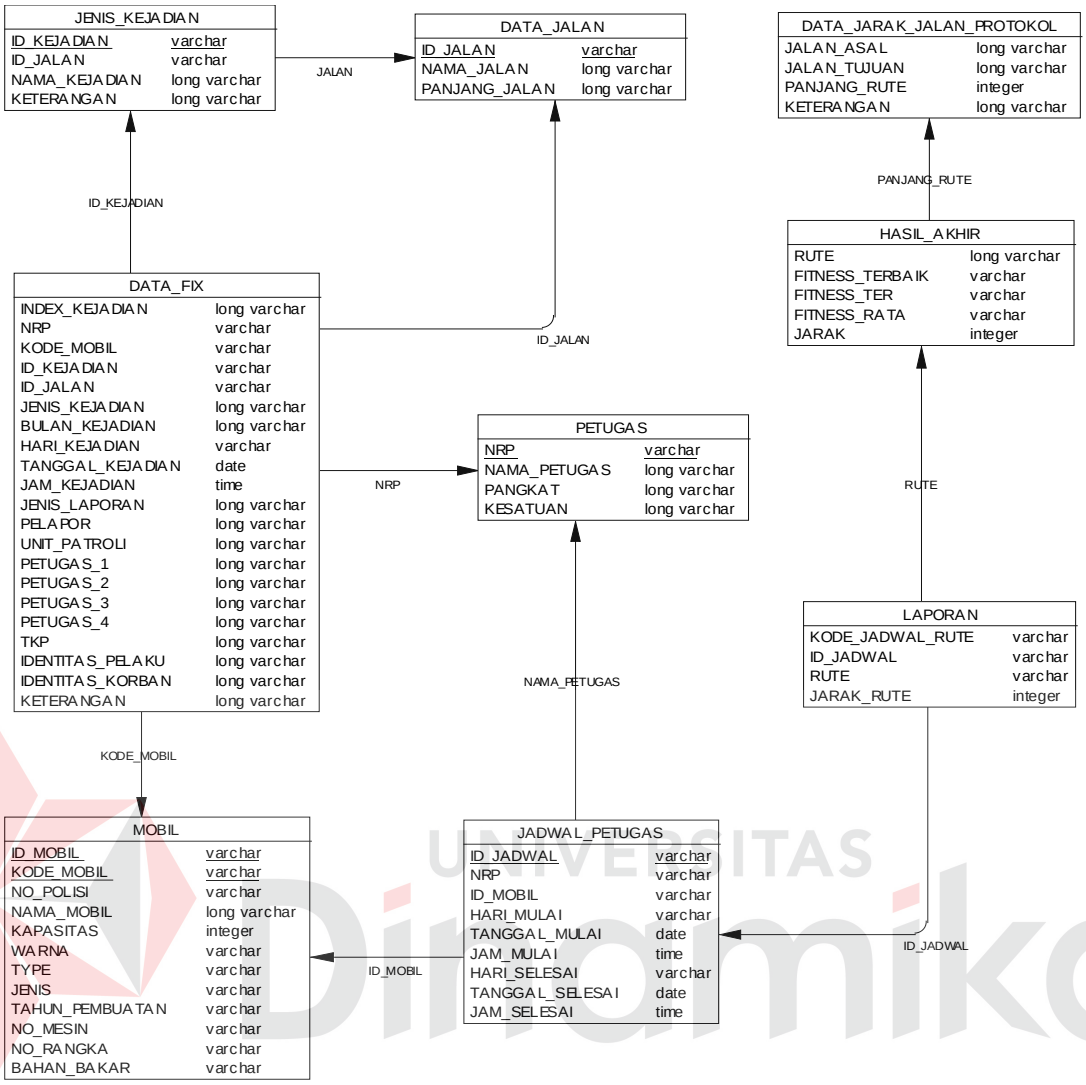
UNIVERSITAS Dinamika

3.4.3 *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) digunakan untuk menggambarkan hubungan data yang digunakan dalam sistem. ERD terdiri dari *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM).



Gambar 3.12 Gambar ERD-CDM (*Conceptual Data model*)



Gambar 3.13 Gambar ERD-PDM (*Physical Data model*)

3.4.4 Struktur Database

Adapun struktur database yang digunakan adalah dengan menggunakan sistem database Microsoft® Access®, dimana dalam struktur database ini berisis tabel-tabel database yang akan diguanakana dalama aplikasi, struktur database dalam aplikasi ini adalah:

1. Tabel Data Jalan

Fungsi : Menyimpan data jalan.

Tabel 3.1 Tabel Data Jalan

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	ID Jalan	Pk	Varchar	50	ID jalan
2	Nama Jalan		Varchar	255	Nama Jalan
3	Panjang Jalan		Single	-	Panjang Jalan

2. Tabel Data Jarak Protokol

Fungsi : Menyimpan data jarak antara jalan asal dan jalan tujuan.

Tabel 3.2 Tabel Data Jarak Protokol

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	ID Jalan Asal	PK	Varchar	50	ID Jalan Asal
2	ID Jalan Tujuan	PK	Varchar	50	ID Jalan Tujuan
3	Panjang Rute		Integer	-	Jarak antar jalan
4	Keterangan		Varchar	255	ID jalan yang dilewati

3. Tabel Data Petugas

Fungsi : Menyimpan Data Petugas Piket.

Tabel 3.3 Tabel Data Petugas Piket

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	NRP	Pk	Varchar	50	NIP Petugas Patroli
2	Nama		Varchar	255	Nama Petugas Patroli
3	Pangkat		Varchar	255	Pangkat Petugas Patroli
4	Kesatuan		Varchar	50	Kesatuan Petugas Ptroli

4. Tabel Data Hasil Proses

Fungsi : Menyimpan Data Hasil proses GA.

Tabel 3.4 Tabel Proses GA

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	Rute		Byte	-	Rute
2	*Fitness*_Best		Text	-	*Fitness* terbaik
3	*Fitness*_Mean		Text	-	Rata-rata *fitness*
4	*Fitness*_Min		Text	-	*Fitness* terburuk
5	Panjang jalan		Text	-	Panjang rute jalan

UNIVERSITAS Dinamika

5. Tabel Data Mobil Patroli

Fungsi : Menyimpan Data Mobil Patroli

Tabel 3.5 Tabel Data Mobil

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	ID Mobil	Pk	Varchar	50	ID Mobil
2	No.Polisi		Varchar	15	Nomor Polisi
3	Kode Mobil	Pk	Varchar	5	Kode setiap mobil patroli
4	Unit Patroli		Varchar	50	Unit Patroli
5	Nama Mobil		Varchar	50	Nama Mobil Patroli
6	Kapasita		Integer	-	Jumlah anggota dalam satu mobil
7	Warna		Varchar	50	Warna mobil
8	Merk/Type		Varchar	50	Type mobil
9	Jenis		Varchar	50	Jenis mobil
10	Tahun Pembuatan		Date/Time	-	Tahun pembuatan mobil
11	No Mesin		Varchar	50	No mesin mobil
12	No Rangka		Varchar	50	No rangka mobil
13	Bahan Bakar		Varchar	50	Bahan bakar mobil

6. Tabel Data Jadwal Patroli

Fungsi : Menyimpan Data Jadwal Patroli

Tabel 3.6 Tabel Jadwal Patroli

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	ID Jadwal	PK	Varchar	50	ID Jadwal
2	NRP	FK	Varchar	50	NRP Petugas
3	Hari Mulai		Varchar	50	Hari Mulai
4	Tanggal Mulai		DateTime	-	Tanggal Mulai patroli
5	Jam Mulai		DateTime	-	Jam mulai petroli
6	Hari Selesai		Varchar	-	Hari Selesai
7	Tanggal Selesai		DateTime	-	Tanggal selesai patroli
8	Jam Selesai		DateTime	-	Jam selesai patroli

UNIVERSITAS Dinamika

7. Tabel Detail Kejadian

Fungsi : Menyimpan Data Detail Kejadian

Tabel 3.7 Tabel Data Detail Kejadian

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	Index	PK	Integer	-	Index
2	ID Kejadian	FK	Varchar	50	ID Kejadian
3	Jenis Kejadian		Varchar	50	Jenis Kejadian
4	Tanggal/bulan/tahun		DateTime	-	Tahun
6	Hari		Varchar	50	Hari
7	Jam		DateTime	-	Jam
8	Kode Mobil	FK	Varchar	50	Kode Mobil
9	Unit Patroli		Varchar	50	Unit Patroli
10	Petugas 1		Varchar	255	Petugas 1
11	Petugas 2		Varchar	255	Petugas 2
12	Petugas 3		Varchar	255	Petugas 3
13	Petugas 4		Varchar	255	Petugas 4
14	NRP	FK	Varchar	50	NRP
15	Jenis Laporan		Varchar	200	Jenis Laporan
16	ID Jalan	FK	Varchar	50	ID Jalan
17	TKP		Varchar	255	TKP
18	Identitas Korban		Varchar	-	Identitas Korban
19	Identitas Pelaku		Varchar	-	Identitas Pelaku
20	Keterangan		Varchar	-	Keterangan

8. Tabel Data Jenis-Jenis Kejadian

Fungsi : Menyimpan Jenis-Jenis Kejadian

Tabel 3.8 Tabel Detail Jenis Kejadian

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	ID Kejadian	Pk	Varchar	50	ID Kejadian
2	Nama Kejadian		Varchar	255	Nama Kejadian
3	Keterangan		Varchar	255	Keterangan

9. Tabel Laporan Proses GA

Fungsi : Menyimpan Laporan Proses GA

Tabel 3.9 Tabel Proses GA

No	Nama Field	Key	Tipe Data	Lebar Field	Keterangan
1	Kode jadwal rute	PK	Varchar	5	Kode jadwal rute patroli
2	ID Jadwal	FK	Varchar	5	Id jadwal petugas
3	Rute		Varchar	300	ID Rute
4	Pajang Rute		Integer	50	Panjang rute

UNIVERSITAS Dinamika

BAB IV

EVALUASI DAN IMPLEMENTASI

4.1 Kebutuhan Sistem

Untuk dapat menjalankan aplikasi optimalisasi rute patroli, maka diperlukan spesifikasi sistem yang tepat agar dapat menunjang berjalannya aplikasi secara optimal.

Kebutuhan sistem ini terdiri dari kebutuhan terhadap perangkat keras (*hardware*) dan perangkat lunak (*software*) minimal yang harus dipenuhi sehingga aplikasi ini dapat berjalan dengan baik.

4.1.1 Perangkat Keras (Hardware)

Perangkat keras komputer adalah komponen-komponen fisik peralatan yang membentuk suatu sistem komputer, serta peralatan-peralatan lain yang mendukung komputer dalam menjalankan tugasnya. Adapun perangkat keras yang diperlukan dalam aplikasi ini adalah:

1. AMD Sempron(tm) 2500 + 1,76 Ghz.
2. Memori DDR sebesar 512MB.
3. Kapasitas harddisk sebesar 80 GB.
4. Mouse, Keyboard, dan Printer

4.1.2 Perangkat Lunak (Software)

Perangkat lunak yang diperlukan adalah program komputer yang diperlukan untuk mengoperasikan fungsi dari perangkat keras. Adapun perangkat lunak yang diperlukan dalam perancangan dan pembuatan aplikasi ini adalah:

1. Tersedia perangkat lunak Microsoft Access xp.
2. Membuat koneksi ODBC.
3. Tersedia perangkat lunak Microsoft Visual Basic 6.0.
4. Tersedia perangkat lunak Map Info 7.5
5. Menjalankan setup program.
6. Menjalankan program.

4.2 Implementasi Sistem

Perangkat lunak ini mempunyai empat menu utama yang terletak di bagian kiri dari tampilan aplikasi.

Gambar 4.1 Form Login

Form Login adalah menu utama untuk masuk dalam aplikasi. Form Login ini juga bertujuan untuk mencatat pelapor dalam setiap data kejadian. Dalm form Loin terdapat dua tombol yaitu Login dan Cancel. Tombol Lgin berguna untuk masuk ke dalam aplikasi, sedangkan Cancel bertujuan untuk membatalkan masuk dalam aplikasi.

Gambr 4.2 Menu Utama

Pada tampilan utama aplikasi terdapat Sub Menu Maintenance, Sub Menu Transaksi, Sub Menu Proses dan Sub Menu Keluar. Di dalam Sub Menu Maintenance terdapat menu-menu untuk memasukkan data-data yang ebrhubungan dengan Sub Menu Transaksi antara lain menu Mobil, menu Jenis Kejadian dan menu Petugas. Di dalam Sub Menu Transaksi yang berhubungan dengan proses pembentukan rute menu-menu tersebut yaitu menu Jenis Perkara, Jadwal Petugas. Di dalam Sub Menu Proses terdapat menu Proses Algoritma Genetika yang merupakan menu untuk menghasilkan rute patroli. Dan yang etrakhir adalah Sub Menu Keluar yang berfungsi untuk keluar dari aplikasi.

UNIVERSITAS Dinamika

Gambar 4.3 Master Mobil

Pada gambar 4.3 menunjukkan form Master Mobil yang berfungsi untuk memasukkan data mobil patroli. Inputan pada form Master Mobil berisi ID Mobil, No Polisi, Kode Mobil, Nama Mobil, dan Kapasita Personil, Warna, Merk/type, Jenis Kendaraan, Tahun Pembuatan, No Mesin, No Rangka, dan Bahan Bakar. Pada form Master Mobil terdapat tombol ADD NEW untuk memasukkan data baru, SAVE untuk menyimpan data, EDIT untuk merubah data, CANCEL untuk membersihkan form, dan CLOSE untuk keluar dari menu.

UNIVERSITAS Dinamika

Gambar 4.4 Master Petugas Piket

Pada gambar 4.4 menunjukkan form Master Petugas Piket yang berfungsi untuk memasukkan data petugas yang akan melakukan patroli, from ini akan berhubungan dengan Form Jadwal Petugas. Inputan pada form Master petugas berisi NRP atau Nomor Induk Petugas, Nama, Pangkat dan Kesatuan. Pada form Master Petugas terdapat tombol proses ADD NEW untuk memasukkan data baru, SAVE untuk menyimpan data, EDIT untuk merubah data, CANCEL untuk membersihkan form , dan CLOSE untuk keluar dari menu dan menuju menu utama.

UNIVERSITAS Dinamika

Gambar 4.5 Master Jenis Kejadian

Pada gambar 4.5 menunjukkan form Master Jenis Kejadian yang berfungsi untuk memasukkan data jenis-jenis kejadian yang terjadi contohnya jenis kejadian nama kejadian curat dengan keterangan pencurian dengan pemberatan. Inputan pada form Master jenis kejadian berisi Kode Kejadian, Nama Kejadian, dan Keterangan. Pada form Master Jenis Kejadian terdapat tombol proses ADD NEW untuk memasukkan data baru, SAVE untuk menyimpan data, EDIT untuk merubah data, CANCEL untuk membersihkan form , dan CLOSE untuk keluar dari menu dan menuju menu utama.

UNIVERSITAS Dinamika

Gambar 4.6 Jadwal Petugas Piket

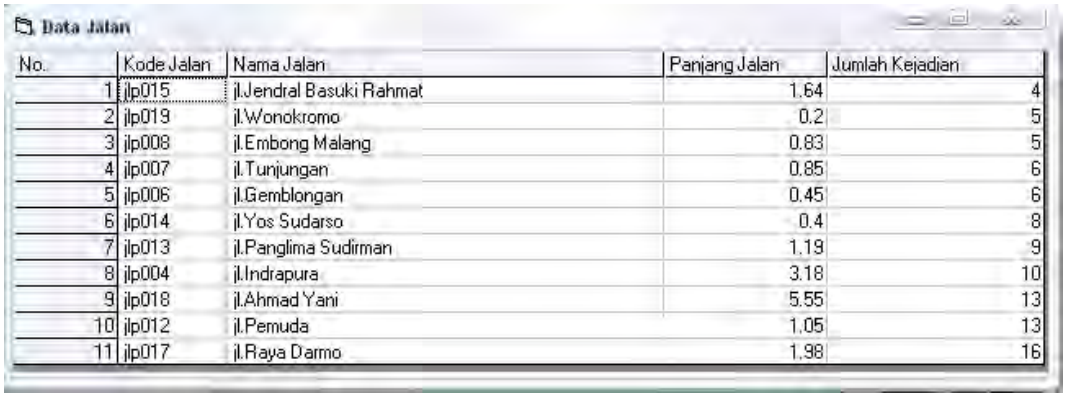
Pada gambar 4.6 menunjukkan form proses jadwal patroli yang berfungsi untuk memasukkan data jadwal petugas. Inputan pada form Jadwal Petugas berisi Kode, Unit Patroli, Petugas I, Petugas II, Hari Mulai Tugas, Tanggal Mulai Tugas, Jam Mulai Tugas, Hari Selesai Tugas, Tanggal Selesai Tugas, dan Jam Selesai Tugas. Pada form Petugas Piket terdapat tombol proses ADD NEW untuk memasukkan data baru, SAVE untuk menyimpan data, EDIT untuk merubah data, CANCEL untuk membersihkan form , dan CLOSE untuk keluar dari menu dan menuju menu utama.

UNIVERSITAS Dinamika

Gambar 4.7 Form Pengaduan Perkara

Pada gambar 4.7 menunjukkan form pengaduan kejadian atau perkara yang berfungsi untuk memasukkan data suatu kejadian di suatu ruas jalan. Inputan pada form Kejadian / Perkara berisi Index, Jenis Kejadian, Hari / Tanggal, Jam, Jenis Laporan, Unit Patroli, Petugas 1, Petugas 2, Petugas 3, Petugas 4, TKP (Tempat Kejadian Perkara), Identitas Pelaku, Identitas Korban, Keterangan. Pada form Pengaduan Kejadian atau Perkara terdapat tombol proses ADD NEW untuk memasukkan data baru, SAVE untuk menyimpan data, EDIT untuk merubah data, CANCEL untuk membersihkan form , dan CLOSE untuk keluar dari menu dan menuju menu utama.

Gambar 4.8 Form Setting Proses Optimaslisasi



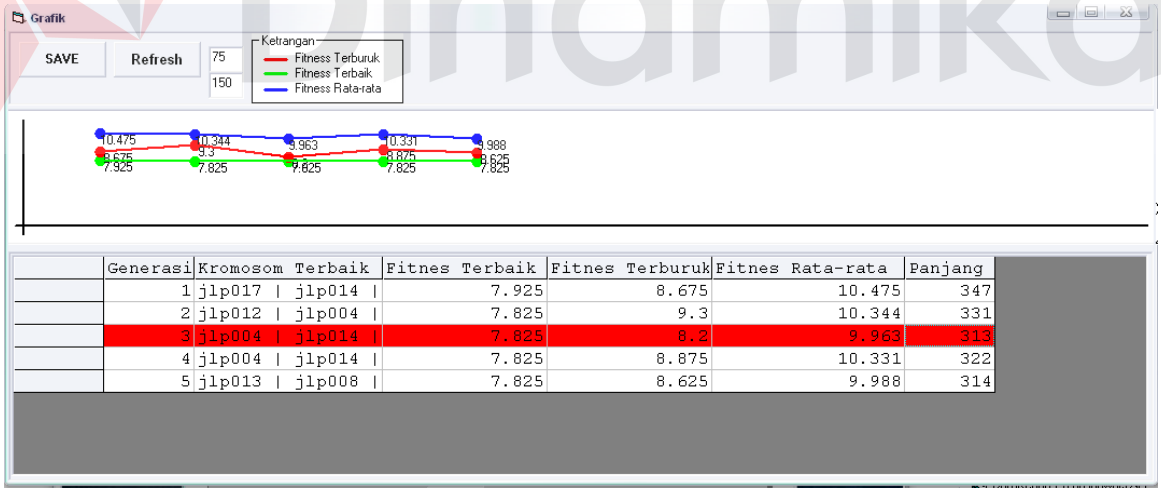
Data Jalan

No.	Kode Jalan	Nama Jalan	Panjang Jalan	Jumlah Kejadian
1	jlp015	jl.Jendral Basuki Rahmat	1,64	4
2	jlp019	jl.Wonokromo	0,2	5
3	jlp008	jl.Embong Malang	0,83	5
4	jlp007	jl.Tunjungan	0,85	6
5	jlp006	jl.Gemblongan	0,45	6
6	jlp014	jl.Yos Sudarso	0,4	8
7	jlp013	jl.Panglima Sudirman	1,19	9
8	jlp004	jl.Indrapura	3,18	10
9	jlp018	jl.Ahmad Yani	5,55	13
10	jlp012	jl.Pemuda	1,05	13
11	jlp017	jl.Raya Darmo	1,98	16

Gambar 4.9 Data Jalan dan Jumlah Kejadian

UNIVERSITAS Dinamika

Gambar 4.8 menunjukan form proses dari optimasi rute. Pada form tersebut terdapat setting parameter input data berupa tombol inputan range waktu kejadian yaitu tombol Mulai Tanggal dan Sampai Tanggal. Tombol Poses 1 akan menampilkan data jalan yang banyak mempunyai kejadian atau perkara berdasarkan range waktunya, data jalan tersebut yang akan menjadi sebuah rute patroli seperti pada gambar 4.9. Inputan setting parameter algoritma yang berupa inputan yaitu ukuran populasi, kecepatan mobil, peluang crossover, peluang mutasi, generasi, serta random seed. Hasil proses dalam bentuk grafik, dimana garis warna hijau menunjukkan fitness terbaik yang diperoleh, garis wana merah menunjukkan fitness terburuk yang diperoleh dari proses, dan garis warna biru menunjukkan fitness rata-rata tiap generasi. Pada form proses terdapat juga tombol Proses 2 untuk memproses, Reset, dan Close. Untuk tombol Proses 2 digunakan untuk memproses data jalan, serta parameter-parameter genetika.

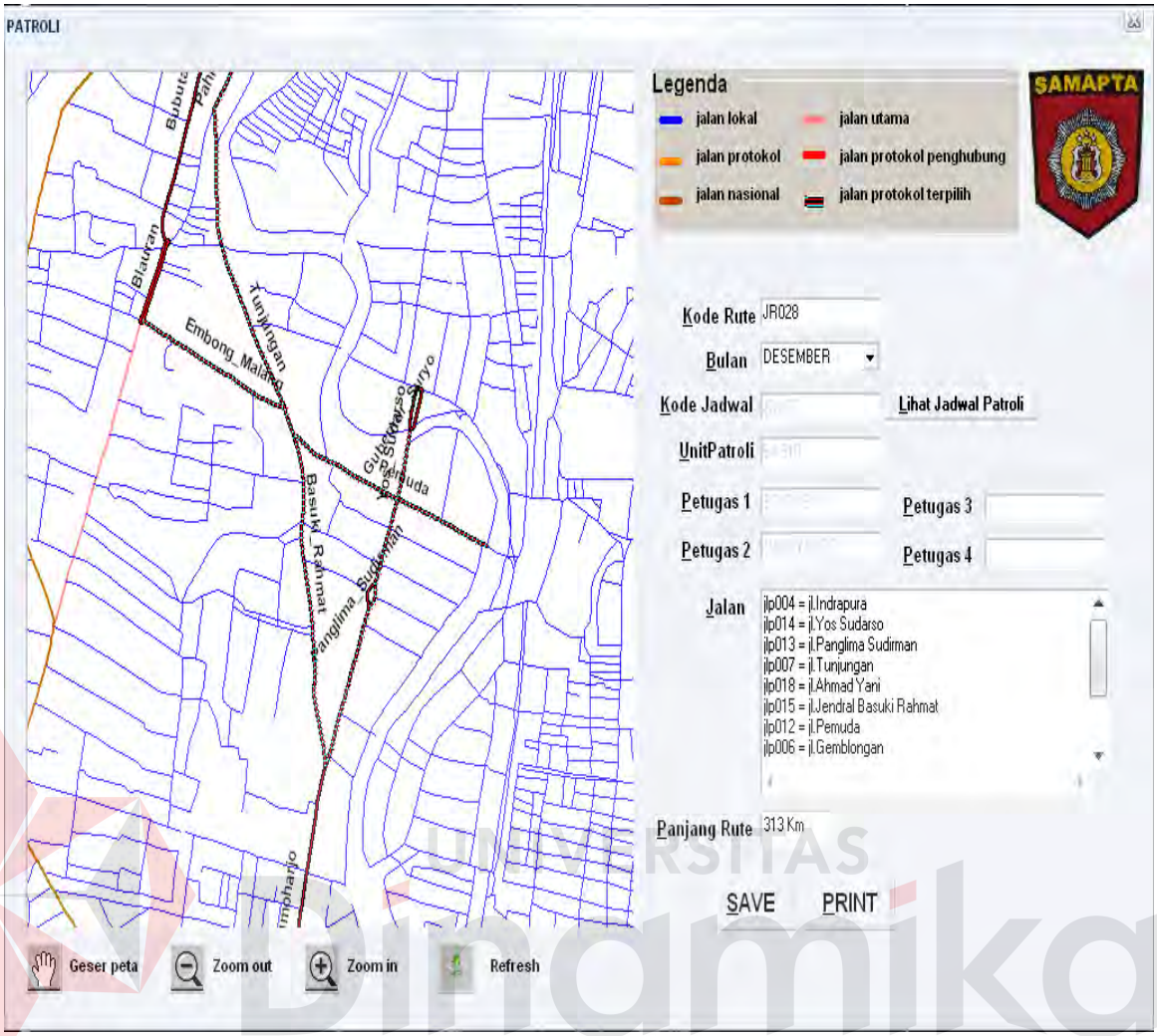


	Generasi	Kromosom Terbaik	Fitnes Terbaik	Fitnes Terburuk	Fitnes Rata-rata	Panjang
	1	jlp017 | jlp014 |	7.925	8.675	10.475	347
	2	jlp012 | jlp004 |	7.825	9.3	10.344	331
	3	jlp004 | jlp014 |	7.825	8.2	9.963	313
	4	jlp004 | jlp014 |	7.825	8.875	10.331	322
	5	jlp013 | jlp008 |	7.825	8.625	9.988	314

Gambar 4.10 Form Proses Optimasi Rute Patroli

Gambar 4.10 adalah hasil pemroses pameter genetika yang masih berupa grafik. Pada form ini terdapat tombol Refresh yang akan mengatur tinggi dan lebar dari grafik, dan tombol Save untuk menyimpan data kromosom yang terbaik.

UNIVERSITAS Dinamika



Gambar 4.11 Form Hasil Akhir

Pada Gambar 4.11 menampilkan hasil akhir dari optimalisasi rute patroli. Pada form ini pada field jalan ditampilkan jalan yang sebagai rute utama yang harus dilewati dimana jalan yang paling atas menampilkan rute yang di tempuh, sedangkan pada peta akan ditampilkan jalan utama yang harus di lewati dan jalan penghubung sebagai jalan penghubung antara jalan sebagai rute yang utama.

4.3 Evaluasi Sistem

Pada bagian eveluasi ini akan dilakukan uji coba aplikasi yang telah dibuat dengan mengacu pada sumber teori yang telah ada. Dalam bagian ini

dilakukan proses membandingkan hasil proses genetika dengan menciptakan bebepara generasi untuk menentukan hasil yang optimal. Hasil optimal dapat dilihat dari nilai fitness terbaik yang paling kecil (minimal) dan jarak yang pendek yang dihasilkan oleh setiap kromosom yang terbentuk dari proses genetika.

Berikut adalah hasil yang diperoleh dari proses genetika yang telah terbentuk berdasar beberapa kriteria yang digunakan :

4.3.1 Uji Coba I Terhadap Generasi

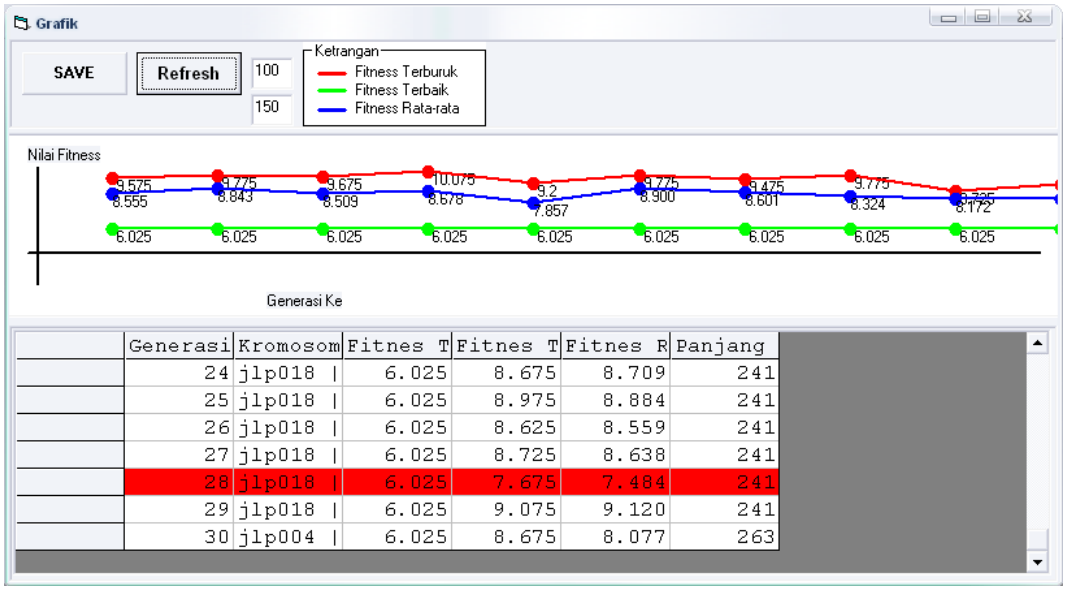
Pada uji coba terhadap generasi ini, akan dilakukan pengujian proses genetika algoritma dengan memberikan nilai generasi yang berbeda-beda dan dengan range waktu yang berbeda yaitu :

Tabel 4.1 Uji Coba I Generasi

	Generasi				
	10	**20**	**30**	**40**	**50**
Populasi	30	30	30	30	30
Prob Crossover	45	45	45	45	45
Prob Mutasi	10	10	10	10	10
Fitness Min	6,025	6,025	6,025	6,025	6,025
Panjang Jalan	328 Km	293 Km	241 Km	241 Km	241 Km
Random Seed	1	1	1	1	1

Dari hasil uji coba I, diperoleh nilai optimal untuk jumlah generasi yang digunakan adalah 30 generasi, dimana pada generasi ke 16 sudah mendapatkan nilai fitness terbaik dan mendapatkan panjang jalan yang paling pendek. Dibawah ini adalah gambar dari grafik perubahan nilai fitness dari hasil uji coba I diatas.

UNIVERSITAS Dinamika



	Generasi	Kromosom	Fitnes T	Fitnes T	Fitnes R	Panjang
	24	jlp018 |	6.025	8.675	8.709	241
	25	jlp018 |	6.025	8.975	8.884	241
	26	jlp018 |	6.025	8.625	8.559	241
	27	jlp018 |	6.025	8.725	8.638	241
	28	jlp018 |	6.025	7.675	7.484	241
	29	jlp018 |	6.025	9.075	9.120	241
	30	jlp004 |	6.025	8.675	8.077	263

Gambar 4.12 Grafik Uji Coba I

4.3.2 Uji Coba II Terhadap Populasi

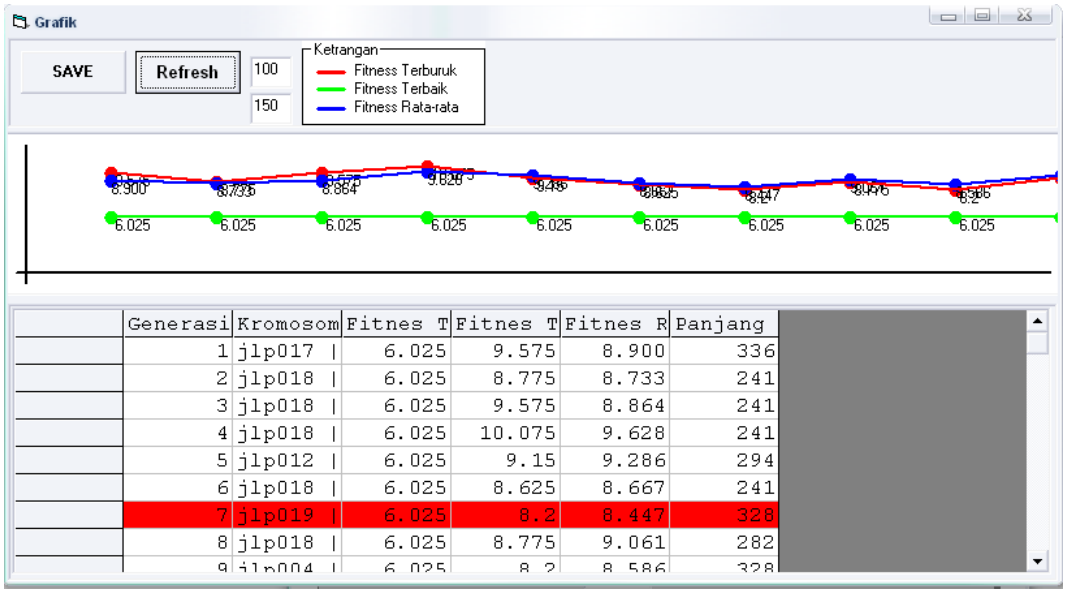
Pada uji coba terhadap jumlah populasi ini, akan dilakukan pengujian proses genetika algoritma dengan melakukan uji coba yang memberikan jumlah populasi yang berbeda-beda dengan range waktu yang sama.

Tabel 4.2 Uji Coba II Populasi

	Populasi				
	10	**20**	**30**	**40**	**50**
Generasi	10	10	10	10	10
Prob Crossover	0,45	0,45	0,45	0,45	0,45
Prob Mutasi	0,01	0,01	0,01	0,01	0,01
Fitness Min	6,025	6,025	6,025	6,025	6,025
Panjang Jalan	328 Km	354 Km	363 Km	336 Km	359 Km
Random Seed	1	1	1	1	1

Dari hasil uji coba II, diperoleh nilai optimal untuk jumlah populasi yang digunakan adalah 10 dimana generasi yang terbaik adalah generasi ke 7. Dibawah

ini adalah gambar dari grafik perubahan nilai fitness dari hasil uji coba II diatas.



Gambar 4.13 Grafik Uji Coba II

4.3.3 Uji Coba III Terhadap Probabilitas Crossover

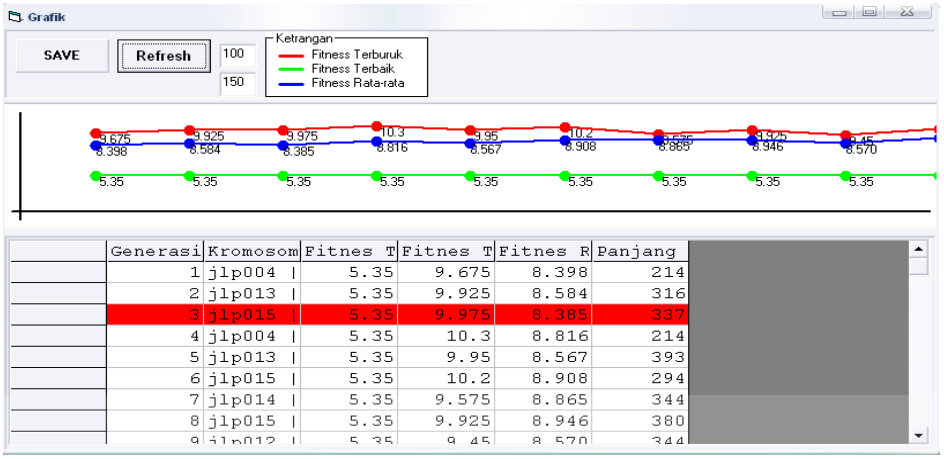
Pada uji coba terhadap probabilitas crossover ini, akan dilakukan pengujian proses genetika algoritma dengan melakukan uji coba yang memberikan nilai probabilitas crossover yang berbeda-beda dengan inputan range waktu yang sama.

Tabel 4.3 Uji Coba III Probabilitas Crossover

	Probabilitas Crossover (%)				
	10	**20**	**30**	**40**	**50**
Generasi	10	10	10	10	10
Populasi	100	100	100	100	100
Prob Mutasi	0,01	0,01	0,01	0,01	0,01
Fitness Min	5,35	5,35	5,35	5,35	5,35
Panjang Jalan	337 Km	337 Km	337 Km	337 Km	337 Km
Random Seed	1	1	1	1	1

UNIVERSITAS Dinamika

Dari hasil uji coba III, diperoleh nilai optimal untuk probabilitas crossover adalah sama tetapi pada probabilitas crossover 30 % dihasilkan generasi yang terbaik pada generasi ke 3. Dibawah ini adalah gambar dari grafik perubahan nilai fitness dari hasil uji coba III diatas.



Gambar 4.14 Grafik Uji Coba III

UNIVERSITAS Dinamika

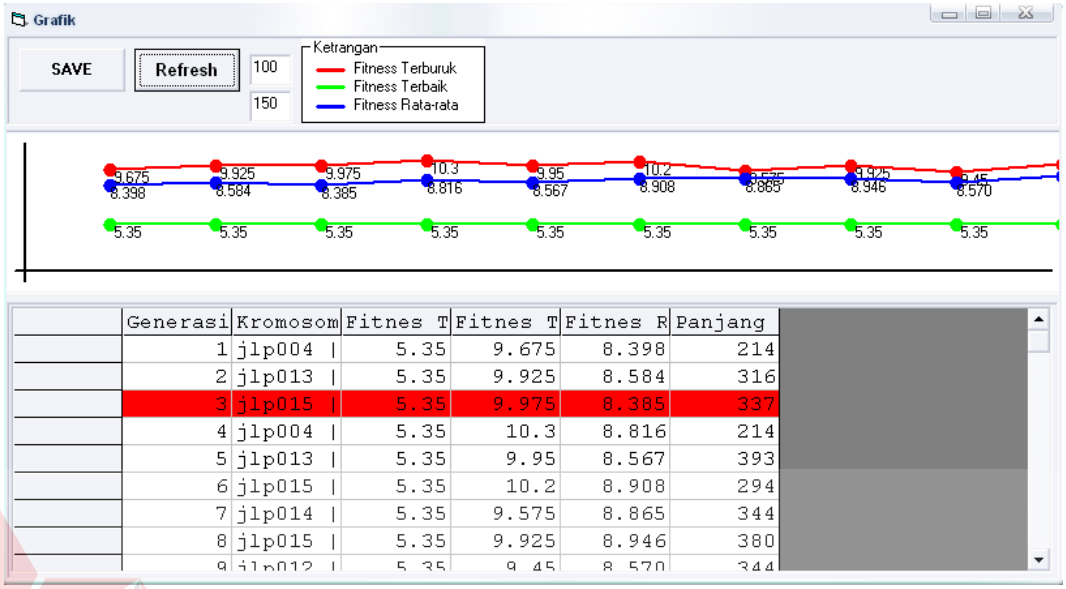
4.3.4 Uji Coba IV Terhadap Probabilitas Mutasi

Pada uji coba terhadap probabilitas mutasi ini, akan dilakukan pengujian proses genetika algoritma dengan melakukan uji coba yang memberikan nilai probabilitas mutasi yang berbeda-beda.

Tabel 4.4 Uji Coba IV Probabilitas Mutasi

	Probabilitas Mutasi (%)				
	10	**20**	**30**	**40**	**50**
Generasi	10	10	10	10	10
Populasi	100	100	100	100	100
Prob Crossover	10	10	10	10	10
Fitness Min	5,35	5,35	5,35	5,35	5,35
Panjang Jalan	337 Km	337 Km	337 Km	337 Km	337 Km
Random Seed	1	1	1	1	1

Dari hasil uji coba IV, diperoleh nilai optimal untuk probabilitas mutasi yang digunakan adalah sama, karena tidak akan merubah hasilnya. Dibawah ini adalah gambar dari grafik perubahan nilai fitness dari hasil uji coba IV diatas.



Gambar 4.15 Grafik Uji Coba IV

Dari beberapa uji coba yang telah dilakukan dalam proses pengujian program dilakukan, sebagai pengukur hasil percobaan digunakan kriteria-kriteria nilai terbaik atau rata-rata. Dan hasil uji coba diatas dapat diperoleh solusi optimal berdasar nilai fitness terkecil (minimal), dengan generasi 30, populasi 30, probabilitas crossover 45%, probabilitas mutasi 10% dan nilai fitness sebesar 6,025 dengan hasil jarak rute 241 Km, dengan penggunaan random seed 1. Hasil dari Jadwal rute patroli dan rutenya terlihat dalam laporan ke KANIT pada Gambar 4.16.

UNIVERSITAS Dinamika

RUTE PATROLI

BULAN	KODE	UNIT PATROLI	RUTE JALAN	PANJANG
DESEMBER	JR035	BJ 810 EDI BASUKI EKO WAHYUDI TONNY H ENDANG K	jlp017 = jl.Raya Darmo jlp014 = jl.Yos Sudarso jlp013 = jl.Panglima Sudirman jlp007 = jl.Tunjungan jlp018 = jl.Ahmad Yani jlp015 = jl.Jendral Basuki Rahmat jlp006 = jl.Gemblongan jlp012 = jl.Pemuda jlp004 = jl.Indrapura jlp008 = jl.Embong Malang jlp019 = jl.Wonokromo	347 Km

Di Setujui
Oleh :

KANIT PATROLI
POLWIILTABES SURABAYA

Gambar 4.16 Jadwal dan Rute Pengiriman Barang

UNIVERSITAS Dinamika

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil uji coba yang diperoleh dari proses yang telah dilakukan dengan menggunakan aplikasi yang telah dibuat maka dapat diambil kesimpulan bahwa penerapan algoritma genetika pada optimalisasi rute patroli dapat memberikan kemudahan bagi petugas untuk mengetahui rute patroli pada bulan selanjutnya secara komputerisasi, dan diharapkan dapat mengurangi jumlah kejahatan.

Sebenarnya dalam pengoptimalisasian rute patroli ini masih terdapat hambatan-hambatan yang antara lain : tidak adanya identifikasi kemacetan lalu lintas terutama pada jam-jam tertentu, tidak mengukur lebar jalan, dan hanya menggunakan jalan protokol.

Dengan penggunaan kombinasi rata-rata parameter genetika yaitu jumlah generasi 30, populasi 30, probabilitas crossover sebesar 45%, probabilitas mutasi 10% maka menhasilkan nilai fitness sebesar 6,025 dengan hasil jarak rute 241 Km, dapat menghasilkan solusi yang lebih optimal dalam menghasilkan rute patroli.

5.2 Saran

Permasalahan masalah optimasi rute patroli pada tugas akhir ini dapat dikembangkan lebih lanjut dengan beberapa saran sebagai berikut :

UNIVERSITAS Dinamika

1. Menambahkan data lebar jalan dan mengidentifikasikan kemacetan lalu lintas terutama pada jam-jam sibuk.
2. Menentukan jalan awal sebagai awal rute patroli dan jalan akhir sebagai akhir dari rute patroli.
3. Memperindah dan memperjelas tampilan terutama pada GIS (*Geographic Information System*)

UNIVERSITAS Dinamika

DAFTAR PUSTAKA

Budiyanto, Eko, 2002, Sistem Informasi Geografis Menggunakan ARC VIEW GIS, Yogyakarta, Indonesia.

Foster, Bob, 2002, Terpadu Fisika,, Jakarta, Indonesia.

Gen, Mutsuo and Cheng, Runwei, 1997. *Generic Algorithm And Engineering Design*,A Shikaga Institute of Techonology A Shikaga, Japan.

Hermanto, Deny, 2006, Tutorial Algoritma Genetika,www.google.com, diakses 24 Agustus 2006, jam 19.00 WIB.

Holland, John, 1975, *Adaption in Natural and Artificial System*, Universitas Michigan,Amerika Serikat.

Kusuwadewi, Sri., 2003, *Artificial Inteligen*, Yogyakarta, Indonesia.

Kusumadewi, Sri, 2005, *Penyelesaian Masalah Optimasi Dengan Teknik-Teknik Heuristik*, Graha Ilmu, Yogyakarta.

Prahasta, Eddy, 2001, Konsep-konsep dasar Sistem Informasi Geografi, Teknik Informatika ITB, CV Informatik, Bandung.

S.Novelianti, 2006. Makalah Ilmiah Solusi Traveliing Problem Menggunakan AlgoritmaGreendy,http://www.comp.nus.edusq/~stevanha/programming/prog_grenndy.html, diakses 02 Agustus 2006, jam 15.00 WIB

UNIVERSITAS Dinamika