

Implementasi Algoritma Genetika dan Google Maps API Dalam Penyelesaian Traveling Salesman Problem with Time Window (TSP-TW) Pada Penjadwalan Rute Perjalanan Divisi Pemasaran STMIK El Rahma

Herdiesel Santoso
Program Studi Sistem Informasi
STMIK El Rahma
herdiesel.santoso@stmikelrahma.ac.id

Rachmad Sanuri
Program Studi Sistem Informasi
STMIK El Rahma
nicesanuri@gmail.com

Abstrak – Divisi pemasaran STMIK El Rahma memiliki permasalahan dengan penjadwalan rute kunjungan ketika harus melakukan perjalanan multi destinasi ke sekolah-sekolah untuk melakukan promosi. Perjalanan multi destinasi dengan mempertimbangkan waktu kunjungan merupakan permasalahan *Travelling Salesman Problem with Time Windows (TSP-TW)*. Algoritma Genetika merupakan salah satu metode pencarian yang dapat digunakan untuk memberikan rute perjalanan yang optimal. Rekomendasi yang diberikan tidak hanya mempertimbangkan jarak tetapi juga waktu tempuh didapatkan menggunakan Google Maps API. Skenario pengujian yang dilakukan adalah pengujian banyak generasi optimal, pengujian banyak populasi optimal, pengujian kombinasi probabilitas *crossover* (Pc) dan probabilitas mutasi (Pm), serta pengujian konsistensi solusi yang dihasilkan Algoritma Genetika. Hasil pengujian menunjukkan bahwa jumlah individu terbaik adalah 150 individu dalam satu populasi. Kriteria berhenti jika setelah 127 generasi berturut-turut didapatkan nilai *fitness* tertinggi yang tidak berubah dan kombinasi probabilitas *crossover* dan probabilitas mutasi yang paling optimal adalah {0.3 : 0.7}.

Kata Kunci: Algoritma Genetika, Google Maps API, Optimasi, Time Windows, Travelling Salesman Problem.

I. PENDAHULUAN

Divisi pemasaran merupakan ujung tombak kampus STMIK ELRAHMA dalam aktivitas penerimaan mahasiswa baru. Kegiatan pemasaran menjadi fungsi integratif organisasi kampus STMIK ELRAHMA, yang langsung berhubungan dengan pihak eksternal, khususnya mahasiswa dan calon mahasiswa [1]. Divisi pemasaran harus mengunjungi lebih dari satu sekolah dengan waktu yang sudah ditentukan. Sekolah-sekolah tersebut dapat berada

dalam satu kota atau berada di luar kota. Biaya yang terbatas, waktu kunjungan yang sudah ditentukan dan banyaknya destinasi yang ingin dikunjungi, membuat divisi pemasaran harus menjadwalkan perjalanan multi destinasi seefektif mungkin.

Perjalanan multi destinasi merupakan permasalahan *Traveling Salesman Problem (TSP)*. TSP termasuk dalam bentuk permasalahan *Vehicle Routing Problem (VRP)* yaitu salah satu bentuk permasalahan transportasi yang melibatkan pendistribusian barang maupun orang kepada pelanggan dengan menggunakan kendaraan. TSP sendiri dapat didefinisikan sebagai suatu kondisi di mana tiap daerah harus dikunjungi tepat satu kali dan kembali lagi ke tempat semula dengan total biaya seminimal mungkin [2]. TSP dapat berupa TSP-simetris dan TSP-asimetris. Pada TSP-simetris biaya perjalanan dari titik A ke titik B adalah sama dengan biaya perjalanan dari titik B ke titik A. Sedangkan pada TSP-asimetris, biaya perjalanan dari titik A ke titik B tidak sama dengan biaya perjalanan dari titik B ke titik A [3]. *Travelling Salesman Problem with Time Windows (TSP-TW)* merupakan pengembangan TSP yang mempertimbangkan waktu kunjungan (*Time Windows*). Waktu kunjungan meliputi total waktu perjalanan, waktu pengiriman, waktu pelayanan, dan waktu kedatangan [4].

Sebagai algoritma pencarian, Algoritma Genetika dapat mencari penyelesaian dari titik-titik optimum populasi dan bukan dari sebuah titik saja [2]. Tetapi dalam pencarian titik-titik optimum populasi tersebut, proses pencarian Algoritma Genetika tidak terjebak dalam daerah optimum lokal, karena itu Algoritma Genetika berbeda dengan algoritma pencarian yang lain [5]. Pada penelitian ini Algoritma Genetika digunakan untuk menyelesaikan permasalahan TSP-TW pada kasus rekomendasi perjalanan divisi pemasaran di STMIK El Rahma Yogyakarta. Rekomendasi yang diberikan dengan mempertimbangkan jarak dan waktu tempuh yang simetris maupun asimetris juga kepadatan lalu lintas yang didapatkan dari Google Maps menggunakan Google Maps

API. Destinasi yang memiliki jarak minimal belum tentu memiliki waktu tempuh yang minimal juga, dikarenakan kepadatan lalu lintas, jalan satu arah dan dua arah, portal jalan, lampu lalu lintas, dan halangan sejenisnya [6]. Google Maps API dapat memberikan jarak terbaik berdasarkan keadaan lalu lintas saat ini dan waktu perjalanan prediktif berdasarkan data waktu historis setiap harinya [7]. Sehingga solusi yang dihasilkan Algoritma Genetika diharapkan mampu mendekati kondisi sebenarnya.

II. PENELITIAN PENDUKUNG

Penelitian yang dilakukan [8], menerapkan algoritma *best-first search* untuk menyelesaikan masalah TSP pada pencarian rute terpendek perjalanan wisata di kota Yogyakarta. Parameter biaya yang digunakan dalam penelitian tersebut hanya jarak dengan jumlah simpul tujuan sebanyak 8 (delapan). *Best-first search* memiliki kelemahan, jika simpulnya semakin banyak maka algoritma ini membutuhkan memori yang cukup banyak dan waktu pemrosesan yang cukup lama. Penelitian algoritma optimasi lain menggunakan metode *Nearest Insertion Heuristik* untuk menyelesaikan permasalahan optimasi *Vehicle Routing Problem* (VRP) dengan studi kasus kendaraan pengangkut sampah [9]. Penelitian tersebut membandingkan antara metode *Nearest Insertion Heuristic* dengan metode modifikasinya, menentukan titik untuk disisipkan dengan mencari tempat titik bebas yang terletak paling dekat dengan suatu titik pada rute. Hasil penelitian menunjukkan bahwa metode *Nearest Insertion Heuristic* menghasilkan jarak yang lebih pendek karena jarak total yang dihasilkan jauh lebih singkat begitupula dengan waktu yang dibutuhkan. Akan tetapi, metode ini mempunyai kekurangan yaitu ruang pencarian yang sempit dan proses pencarian solusi hanya dilakukan satu kali.

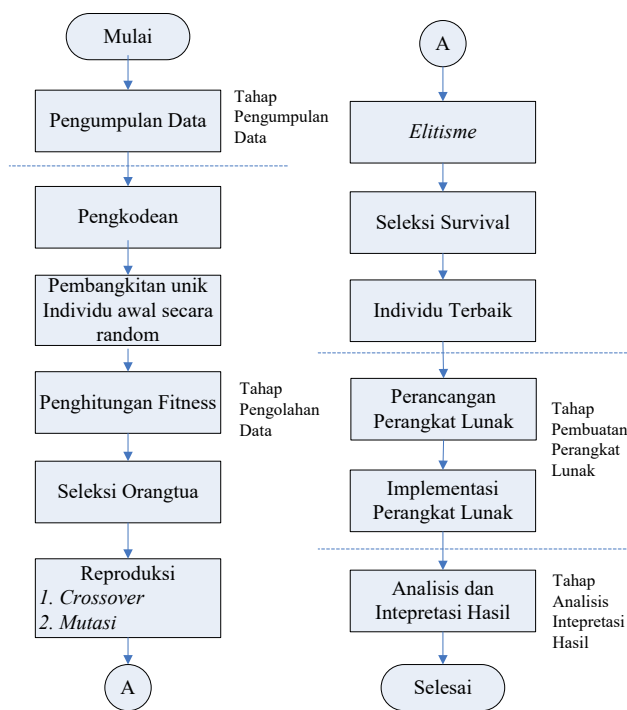
Penelitian lain menggunakan algoritma *heuristic* yaitu *simulated annealing* untuk optimasi *Vehicle Routing Problem with Time Windows* (VRPTW) [10]. VRPTW merupakan pengembangan dari VRP yang mempertimbangkan waktu kunjungan (*Time Windows*) sebagai batasan. Fungsi-fungsi khusus ditambahkan untuk memecahkan persoalan berkaitan dengan ruang pencarian besar dari VRPTW. Serangkaian eksperimen komputasi membuktikan bahwa *simulated annealing* yang ditingkatkan dapat menghasilkan solusi yang baik dalam waktu rata-rata 82.29 detik. Selain penelitian [10], [11] menyelesaikan masalah TSP dengan menggunakan algoritma *simulated annealing*. Parameter yang digunakan adalah biaya, jarak tempuh dan waktu perjalanan. Langkah pertama yang dilakukan adalah rute awal ditentukan secara *random* dan dihitung biaya, jarak dan waktu perjalanan *salesman*. Hasil penelitian adalah rute dengan biaya, jarak tempuh dan waktu perjalanan yang minimum. *Simulated annealing* memilih simpul baru yang memiliki biaya terkecil diantara suksesor-suksesor saudaranya (*sibling*) sehingga bisa menghasilkan solusi yang tidak optimal. Selain itu karena bergantung pada

suatu nilai probabilitas, maka *simulated annealing* tidak selalu menemukan solusi terbaik.

Penelitian [4], menyelesaikan masalah TSP-TW dengan menggunakan Algoritma Genetika penjadwalan perjalanan wisata di Pulau Bali. Ukuran generasi, ukuran populasi, probabilitas *crossover* dan mutasi dan metode seleksi yang tepat memungkinkan untuk mengoptimalkan pemilihan jadwal rute wisata dengan *time window* masing-masing. Sedangkan [12], membandingkan antara Algoritma Genetika dan Algoritma Greedy dalam penentuan rute terpendek perjalanan *marketing* dengan parameter menggunakan jarak tempuh. Objek penelitian hanya mencakup satu kota saja diwilayah *marketing* promosi BSI Tasikmalaya. Penelitian tersebut menunjukkan bahwa Algoritma Genetika lebih baik dari Algoritma Greedy dalam menentukan solusi yang optimum.

III. METODOLOGI PENELITIAN

Tahapan-tahapan yang dilakukan dalam penelitian ini mengikuti alur pada Gambar 1.



Gambar 1. Tahapan Penelitian

Tahapan penelitian terdiri dari 4 tahapan utama yaitu pengumpulan data, pengolahan data dengan algoritma genetika, pembuatan perangkat lunak serta analisis dan interpretasi hasil. Penjelasan setiap tahapan adalah sebagai berikut:

A. Pengumpulan Data

Data yang digunakan untuk penelitian adalah lokasi-lokasi yang menjadi tujuan perjalanan divisi pemasaran

STMIK El Rahma Yogyakarta seperti sekolah negeri atau swasta setaraf SMA/SMK/MA dan pesantren di Provinsi Yogyakarta. Data tersebut didapatkan dari Google Maps menggunakan Google Maps API yang meliputi data jarak dan waktu tempuh dan koordinat (lintang dan bujur). Teknik pengumpulan data yang digunakan adalah :

1. Observasi, yaitu pengumpulan data dengan melakukan pengamatan langsung terhadap data yang diperlukan.
2. *Interview*, yaitu pengumpulan data dengan melakukan wawancara atau tanya jawab secara langsung dengan divisi pemasaran STMIK El Rahma Yogyakarta.

Penelitian ini menggunakan 10 objek, dengan 9 objek SMA/SMK/MA dan pesantren yang menjadi lokasi tujuan dan STMIK El Rahma yang menjadi lokasi awal perjalanan.

Tabel 1. Objek Penelitian Beserta Kodenya

Kode	Nama Objek	Lat	Long	Ketersediaan Waktu
1	STMIK El Rahma	-7,823	110,372	
2	SMK Muh. 3 Yogyakarta	-7,822	110,389	08:00 - 14:00
3	SMK Piri 1 Yogyakarta	-7,794	110,383	09:00 - 13:00
4	MA Muh. 1 Yogyakarta	-7,806	110,356	07:00 - 14:30
5	SMK Negeri 2 Yogyakarta	-7,777	110,367	08:00 - 14:00
6	SMK Negeri 5 Yogyakarta	-7,799	110,395	09:00 - 14:00
7	SMK Muh. 1 Yogyakarta	-7,819	110,382	10:00 - 14:00
8	MA Ali Maksum	-7,829	110,360	07:00 - 13:00
9	MAN 4 Bantul	-7,771	110,362	08:00 - 14:00
10	SMK Negeri 7 Yogyakarta	-7,786	110,365	07:00 - 13:00

Data perbandingan jarak 10 objek disajikan dalam Tabel 2, sedangkan data perbandingan waktu tempuh setiap objek disajikan pada Tabel 3. Pada Tabel tersebut terlihat jarak dan waktu tempuh dari STMIK El-Rahma ke SMK Negeri 2 Yogyakarta dan dari SMK Negeri 2 Yogyakarta ke STMIK El-Rahma tidak sama (*asimetris*). Karena antara STMIK El-Rahma dan SMK Negeri 2 Yogyakarta menerapkan jalan satu arah, sehingga dari STMIK El-Rahma ke SMK Negeri 2 Yogyakarta harus mencari rute yang berbeda.

B. Pengolahan Data

Pengolahan data pada penelitian ini bertujuan untuk mencari rute paling optimal menggunakan Algoritma Genetika. Adapun langkah-langkahnya adalah sebagai berikut [13].

1. Pengkodean

Suatu permasalahan dalam Algoritma Genetika harus dikonversi dulu ke dalam bentuk individu yang diwakili oleh satu atau lebih kromosom dengan kode tertentu. Pada permasalahan perjalanan atau kunjungan, solusi yang ingin dicari adalah urutan objek yang haru dikunjungi. Skema pengkodean yang biasa digunakan adalah *permutation encoding* (pengkodean permutasi) yaitu setiap objek direpresentasikan dengan bilangan bulat bertipe integer. Jumlah gen dalam setiap individu adalah sebanyak jumlah objek destinasi perjalan yang dipilih.

2. Inisiasi populasi

Inisiasi adalah membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Dengan menggunakan fungsi *random* akan dipilih suatu individu dengan memilih secara acak antara bilangan 1 sampai dengan jumlah gen hingga terbentuk suatu kromosom. Kromosom yang sudah dibangkitkan akan menjadi sebuah populasi.

3. Fungsi evaluasi

Setelah terbentuk satu populasi, tahap selanjutnya adalah menghitung nilai *fitness* setiap individu. Oleh karena masalah perjalanan atau kunjungan adalah permasalahan TSP dan tujuannya adalah meminimalkan total biaya, maka fungsi *fitness* yang bisa digunakan adalah 1 dibagi dengan total biaya ditambahkan dengan pinalti.

4. Seleksi orangtua

Proses penyeleksian yang digunakan dalam penelitian ini adalah *roulette-wheel selection* [13]. Metode *roulette-wheel selection* sangat mudah diimplementasikan dalam pemrograman. Berikut ini adalah langkah-langkah dari *roulette-wheel selection*.

- a. Menghitung nilai *fitness* dari masing-masing individu (fi dimana I adalah individu ke-1 s/d ke-n).
- b. Menghitung total *fitness* semua individu.
- c. Menghitung probabilitas setiap individu tersebut. Nilai *fitness* setiap individu diperoleh (langkah a), kemudian nilai *fitness* tersebut dijumlah menghasilkan nilai total *fitness* (langkah b). Probabilitas dicari dengan membagi nilai *fitness* masing-masing individu dengan nilai total *fitness* semua individu.
- d. Dengan $P_s[i]$ adalah peluang individu ke-i dan $f(i)$ adalah nilai *fitness* individu ke-I, $i = 1,2,3,4,\dots,n$.
- e. Menghitung probabilitas kumulatif.
- f. Membangkitkan bilangan *random* $R \in (0, 1]$, berdasarkan banyaknya populasi pada generasi.
- g. Menentukan individu yang terpilih sebagai induk berdasarkan letak bilangan *random* yang dihasilkan, dengan ketentuan, jika bilangan acak $R[i]$ kurang dari *fitnes* kumulatif pertama $q[1]$, maka kromosom ke-i diganti dengan kromosom pertama (kromosom pertama sebagai induk), selain itu pilih kromosom ke-i sebagai induk dengan syarat $q[i-1] < R \leq q[i]$ (kromosom yang mempunyai kumulatif *fitness* $q[i]$ lebih besar atau sama dengan dari $R[i]$). Misal $R[1]$ kurang dari $q[4]$ maka kromosom $K[1]$ diganti dengan kromosom $K[4]$ pada populasi awal.

5. Reproduksi

Pada penelitian ini menggunakan *cycle crossover* sebagai metode *crossover* untuk representasi permutasi [3]. Proses selanjutnya adalah mutasi menggunakan skema mutasi *swap*

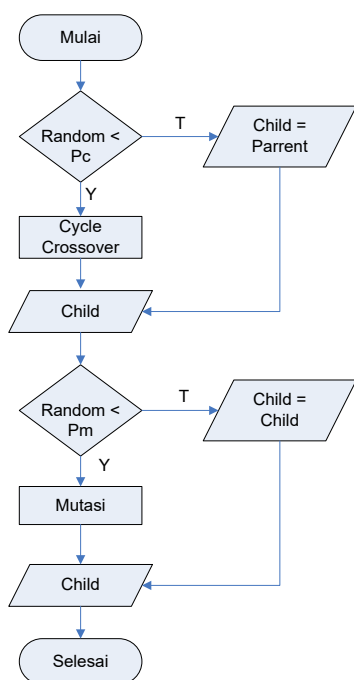
mutation. Proses *crossover* dan mutasi mengikuti skema pada Gambar 2.

Tabel 2. Data Jarak Tempuh Antar Objek (Km)

Asal/Tujuan	1	2	3	4	5	6	7	8	9	10
1	0	2,97	4,27	3,48	6,81	4,67	2,01	2,05	11,39	7,01
2	2,63	0	4,19	4,74	6,88	2,97	1,32	4,96	14,18	7,08
3	5,42	4,13	0	4,4	3,65	1,79	3,65	6,2	15,63	3,85
4	3,58	4,74	4,4	0	4,22	5,05	4,93	3,18	11,05	4,04
5	6,18	7,02	3,79	4,22	0	5,58	6,54	7,05	16,31	1,48
6	4,81	2,97	1,81	5,14	5,46	0	3,04	6,68	16,09	5,66
7	1,76	2	4,19	4,1	6,76	4,55	0	3,57	13,04	6,96
8	2,05	5,43	6,24	3,18	7,4	6,67	4,89	0	9,96	7,23
9	11,38	13,58	15,65	11,06	15,28	16,13	13,04	9,64	0	15,1
10	6,25	7,58	4,33	3,08	2,07	6,13	7,08	7,42	15,17	0

Tabel 3. Data Waktu Tempuh Antar Objek (Menit)

Asal/Tujuan	1	2	3	4	5	6	7	8	9	10
1	0	8,7	12,93	11,25	20,52	12,83	6,95	7,32	25,32	20,35
2	8,08	0	10,83	14,7	19,62	7,85	3,5	14,57	30,42	19,45
3	14,73	10,22	0	14,73	10,68	4,9	9,23	19,32	38,43	10,53
4	12,57	14,98	15,15	0	13,65	18,07	15,87	10,52	25,63	12,3
5	21,22	19,92	11,3	14,32	0	16,22	18,93	22,6	39,12	5,58
6	13,78	7,83	5,32	16,62	16	0	8,28	19,35	36,82	15,83
7	5,48	5,72	12,6	13,35	20,17	12,5	0	11,633	28,52	20,01
8	7,28	12,17	19,05	10,4	24,05	18,93	12,28	0	21,78	22,7
9	25,05	28,63	37,85	25,82	39,47	35,42	28,73	20,37	0	38,12
10	21,08	20,83	12,22	10,37	7,25	17,13	19,85	20,13	35,17	0



Gambar 2. Skema *Crossover* dan *Mutasi*

6. Elitisme

Untuk menjaga agar individu bernilai *fitness* tertinggi tidak hilang selama evolusi, perlu dibuat skema elitisme. Individu hasil proses elitisme akan menjadi salah satu individu baru pada generasi selanjutnya.

7. Seleksi *survivor*

Seleksi *survivor* akan menggantikan N individu dari pada satu generasi digantikan sekaligus oleh N individu baru. Sejumlah N individu tersebut harus unik artinya tidak ada individu yang sama dalam satu populasi.

8. Kriteria berhenti

Iterasi berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi yang tidak berubah. Sehingga apabila perhitungan dilanjutkan hingga ke generasi ke-N maka diyakinkan bahwa nilai *fitness* yang terendah tetap tidak akan berubah.

C. Pembuatan Perangkat Lunak

Tahap selanjutnya adalah mengubah spesifikasi sistem menjadi perangkat lunak dimana pemrograman menulis program dengan bahasa pemrograman yang telah dipilih. Bahasa pemrograman yang digunakan untuk

mengimplementasikan menggunakan HTML, CSS dan PHP. Sedangkan fungsi-fungsi Google Maps API digunakan untuk mengenali objek (*geocoding*), mendapatkan jarak, waktu tempuh (*distance matrix*) dan menampilkan rute atau petunjuk menampilkan petunjuk arah (*direction*) [14]. Untuk dapat menggunakan fungsi-fungsi dari Google Maps API diperlukan bahasa pemrograman Javascript.

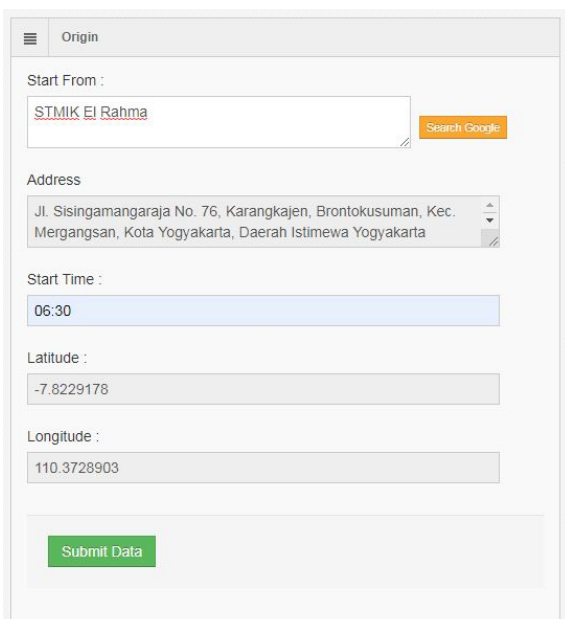
D. Analisis dan Interpretasi Hasil

Pada tahap ini dilakukan pembahasan dan analisis model Algoritma Genetika yang telah dirancang sesuai tujuan penelitian. Analisis meliputi analisis perangkat lunak, analisis banyak generasi, analisis banyak populasi dan analisis kombinasi probabilitas *crossover* dan probabilitas mutasi yang optimal. Hasil analisis diinterpretasikan dengan jelas untuk membantu penarikan kesimpulan.

IV. HASIL DAN PEMBAHASAN

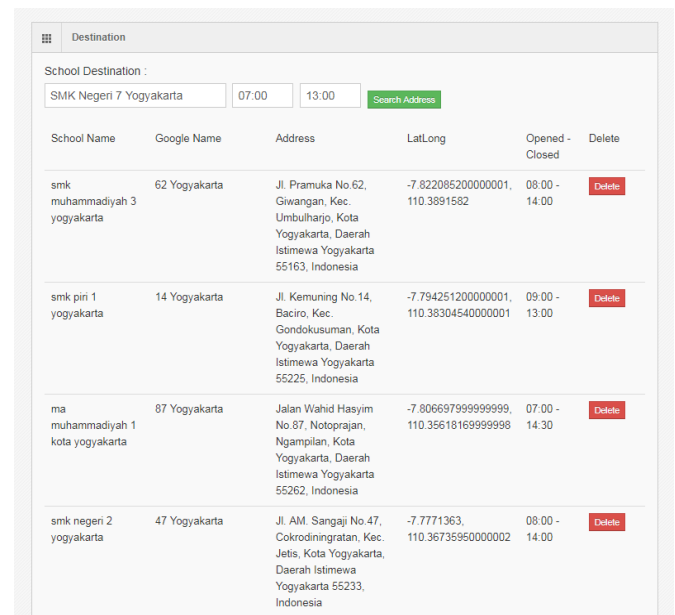
A. Implementasi Perangkat Lunak

Pembuatan perangkat lunak bertujuan untuk memudahkan divisi pemasaran STMIK El Rahma mendapatkan rekomendasi rute perjalanan. Halaman depan aplikasi terdiri dari menu *input origin* yaitu menu untuk memasukan lokasi awal perjalanan dan menu *input destination* yaitu menu untuk memasukan lokasi yang menjadi tujuan perjalanan dari divisi pemasaran STMIK El Rahma. Pada menu *input origin* pengguna memasukan objek yang menjadi lokasi awal perjalanan dan waktu mulai perjalanan (*start time*). Jika objek tersebut dikenali oleh aplikasi, maka akan muncul nama lokasi, alamat beserta koordinatnya. Selanjutnya pengguna memasukan waktu memulai perjalanan.



Gambar 3. Implementasi Menu Input Lokasi Awal

Pada menu *input destination* pengguna perangkat lunak memasukan objek yang menjadi tujuan perjalanan promosi serta waktu ketersediaan kunjungan. Jika objek tersebut dikenali oleh aplikasi, maka akan muncul nama lokasi, alamat beserta koordinat lintang dan bujur. Gambar 4 merupakan implementasi menu input lokasi tujuan. Google Maps API menghitung jarak dan waktu tempuh dari lokasi awal ke masing-masing lokasi tujuan. Selanjutnya aplikasi melakukan perhitungan menggunakan Algoritma Genetika dengan parameter waktu kedatangan, waktu ketersediaan, jarak, waktu tempuh, jumlah populasi, jumlah generasi, nilai probabilitas *crossover* (Pc) dan probabilitas mutasi (Pm) yang sudah ditentukan. Hasil perhitungan adalah solusi rute terbaik, nilai *fitness*, rekomendasi rute perjalanan lengkap dengan estimasi jarak dan waktu tempuhnya. Implementasi menu hasil perhitungan dengan Algoritma Genetika ditampilkan pada Gambar 5.



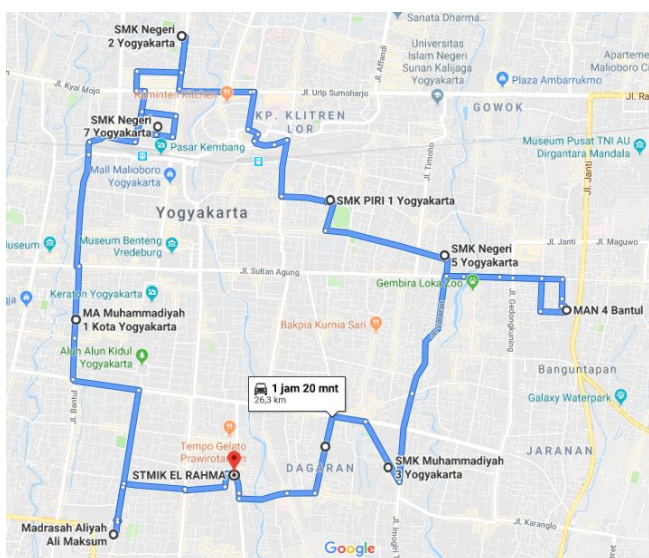
Gambar 4. Implementasi Menu Input Lokasi Tujuan

Pengguna juga dapat melihat petunjuk arah menuju lokasi tujuan. Rute yang ditampilkan adalah rute tercepat saat pengguna menggunakan aplikasi berdasarkan kondisi *real time* lalu lintas yang direkomendasikan dengan bantuan Google Maps API. Rute tersebut dapat berubah sesuai dengan kondisi lalu lintas saat pengguna menggunakan aplikasi. Implementasi petunjuk arah ditampilkan pada gambar 6.

Solusi Terbaik Yang Ditemukan Adalah 1-8-4-10-5-3-6-9-2-7-1 dengan fitness 0.0040983606557377 dengan 165 generations.

Perjalan Ke -	Rekomendasi Tujuan	Estimasi Jarak	Estimasi Waktu	Rute
1	STMIK El Rahma ke madrasah aliyah ali maksum	2.05 Km	7.3 menit	Rute
2	madrasah aliyah ali maksum ke ma muhammadiyah 1 kota yogyakarta	3.183 Km	10.18 menit	Rute
3	ma muhammadiyah 1 kota yogyakarta ke smk negeri 7 yogyakarta	4.03 Km	12.1 menit	Rute
4	smk negeri 7 yogyakarta ke smk negeri 2 yogyakarta	2.074 Km	7.22 menit	Rute
5	smk negeri 2 yogyakarta ke smk piri 1 yogyakarta	3.791 Km	11.22 menit	Rute
6	smk piri 1 yogyakarta ke smk negeri 5 yogyakarta	1.792 Km	4.97 menit	Rute
7	smk negeri 5 yogyakarta ke man 4 bantul	2.051 Km	6.45 menit	Rute
8	man 4 bantul ke smk muhammadiyah 3 yogyakarta	4.477 Km	13.28 menit	Rute
9	smk muhammadiyah 3 yogyakarta ke smk muhammadiyah 1 yogyakarta	1.322 Km	3.42 menit	Rute
10	smk muhammadiyah 1 yogyakarta ke STMIK El Rahma	1.763 Km	5.58 menit	Rute
Total		26.533 Km	81.72 menit	

Gambar 5. Implementasi Menu Hasil Perhitungan



Gambar 6. Implementasi Menu Petunjuk Arah

B. Pengujian Generasi

Pengujian generasi adalah pengujian untuk mencari nilai *threshold* optimal sebagai kriteria berhenti. Pengujian ini menggunakan metode seleksi elitis, banyaknya individu dalam satu populasi adalah 100 dan nilai kombinasi probabilitas *crossover* (Pc) dan probabilitas mutasi (Pm) masing-masing adalah 0,5 [4]. Jumlah gen yang digunakan 6 hingga 10 gen pada Tabel 1, dengan 5 kali percobaan setiap gen. Jumlah generasi awal adalah 500 generasi. Setiap kali percobaan dicatat maksimum interval perubahan nilai *fitness* sebagai *threshold*. Misalkan dalam satu kali percobaan, nilai *fitness* pertama adalah 0,333 pada generasi ke-1. Pada generasi ke-5 nilai *fitness* berubah 0,343, maka interval yang tercatat adalah 4. Selanjutnya pada generasi ke-12 nilai *fitness* berubah 0,351, maka interval yang tercatat 7 dan sampai generasi ke 500 tidak ditemukan nilai *fitness* yang lebih tinggi maka iterasi akan dihentikan. Ketika akhir percobaan maka akan tercatat nilai *threshold* adalah maksimum interval perubahan nilai *fitness* yaitu 7. Hasil percobaan pengujian generasi dapat dilihat pada Tabel 4.

Tabel 4. Pengujian Generasi

Percobaan Ke	Jumlah Gen/Objek				
	6	7	8	9	10
1	12	13	24	20	126
2	14	13	63	40	41
3	11	14	76	94	71
4	11	16	23	66	50
5	13	17	44	44	41
Maksimum Threshold	14	17	76	94	126

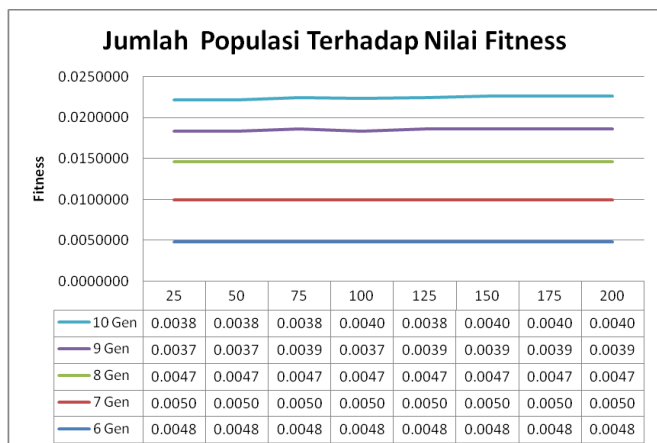
Nilai jangkauan terjauh perubahan nilai *fitness* adalah 126. Nilai tersebut akan menjadi *threshold* optimal sebagai kriteria berhenti. Setelah 126 generasi berturut-turut didapatkan nilai *fitness* tertinggi yang tidak berubah maka iterasi akan dihentikan dan nilai *fitness* terakhir akan dijadikan solusi optimal.

C. Hasil dan Analisis Pengujian Populasi

Tujuan pengujian populasi adalah untuk mencari banyaknya individu dalam satu populasi untuk satu generasi. Nilai kombinasi probabilitas *Crossover* (Pc) dan probabilitas mutasi (Pm) masing-masing adalah 0,5. Populasi yang ditentukan dimulai dari 25 hingga maksimal 200 dengan kelipatan 25. Jumlah objek atau gen yang diuji 6 hingga 10. Nilai *threshold* menggunakan hasil dari pengujian generasi yaitu 126, artinya jika setelah 126 generasi berturut-turut didapatkan nilai *fitness* tertinggi yang tidak berubah maka iterasi akan dihentikan dan nilai *fitness* terakhir dijadikan solusi optimal.

Gambar 7 menunjukkan grafik jumlah populasi terhadap nilai *fitness*. Jumlah objek atau gen lebih kecil sama dengan 8 (gen ≤ 8), nilai *fitness* tertinggi diperoleh saat jumlah individu 25. Karena setelah 25 individu tidak ada perubahan yang signifikan terhadap nilai *fitness*, jumlah individu optimal untuk jumlah gen atau objek lebih kecil atau sama dengan 8 adalah 30 individu pada setiap populasi. Jumlah gen atau objek lebih besar sama dengan 9 (gen ≥ 9), nilai *fitness* tertinggi mulai beragam, tetapi stabil saat jumlah

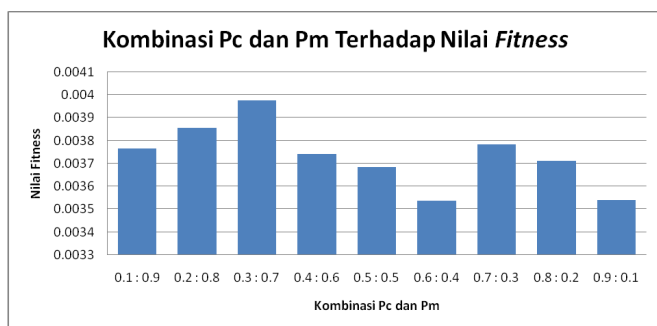
individu 150. Karena setelah 150 individu tidak ada perubahan yang signifikan terhadap nilai *fitness*, jumlah individu optimal untuk jumlah gen atau objek lebih besar sama dengan dari 9 adalah 150 individu pada setiap populasi.



Gambar 7. Grafik Nilai *Fitness* Terhadap Populasi

D. Pengujian *Probabilitas Crossover* dan *Probabilitas Mutasi*

Pengujian kombinasi Pc dan Pm ditentukan nilai Pc dan Pm dengan kelipatan 0,1 pada nilai 0,1 : 0,9 sampai dengan 0,9 : 0,1 dimana jumlah Pc dan Pm adalah 1. Pada pengujian penelitian ini, jumlah gen yang dipakai adalah 10 dan jumlah populasi yang dipakai adalah 150 populasi yang didapatkan dari pengujian populasi. *Threshold* yang digunakan berdasarkan hasil dari pengujian generasi yaitu 126 artinya jika setelah 126 generasi berturut-turut didapatkan nilai *fitness* tertinggi yang tidak berubah maka iterasi akan dihentikan dan nilai *fitness* terakhir akan dijadikan solusi optimal. Setiap percobaan diulang 5 kali kemudian diambil nilai rata-ratanya.



Gambar 8. Grafik Nilai *Fitness* Terhadap Pc dan Pm

Gambar 8 diketahui rata-rata nilai *fitness* terbaik adalah 0,003976 yaitu kombinasi Pc dan Pm masing-masing {0,3 :

0,7}. Rata-rata nilai *fitness* terburuk adalah pada kombinasi Pc dan Pm {0,6 : 0,4} dengan rata-rata nilai *fitness*-nya adalah 0,003538. Hal ini menunjukkan bahwa kasus rekomendasi rute perjalanan multi destinasi paling optimal menggunakan kombinasi Pc dan Pm {0,3 : 0,7}.

E. Pengujian Konsistensi Solusi

Algoritma Genetika sarat akan kebolehjadian (probabilitas) sehingga berbagai macam kemungkinan bisa berpengaruh dalam menghasilkan solusi. Pengujian konsistensi dilakukan untuk melihat keandalan Algoritma Genetika yang sudah diterapkan pada sistem penjadwal perjalanan divisi pemasaran STMIK El Rahma. Parameter Algoritma Genetika yang dipilih harus konsisten memberikan solusi yang optimal ataupun mendekati optimal. Jumlah objek atau gen yang akan diuji 6 hingga 10 gen dan jumlah populasi yang dipakai adalah 150 populasi. Kombinasi Pc dan Pm masing-masing {0,3 : 0,7}. Jika setelah 126 generasi berturut-turut didapatkan nilai *fitness* tertinggi yang tidak berubah maka iterasi akan dihentikan dan nilai *fitness* terakhir akan dijadikan solusi optimal. Setiap percobaan diulang 5 kali kemudian diambil solusi dengan nilai *fitness* terbesar. Tabel 5 menunjukkan hasil pengujian konsistensi Algoritma Genetika untuk memberikan rekomendasi rute perjalanan divisi pemasaran STMIK El Rahma. Sistem yang dikembangkan mampu memberikan rekomendasi rute perjalanan sesuai dengan input yang diberikan. Jika dilihat dari hasil pengujian dari 6 gen hingga 10 gen, Algoritma Genetika mampu memberikan solusi yang konsisten. Hal ini dapat dilihat dari rute dan estimasi jarak dan waktu tempuh yang direkomendasikan.

V. KESIMPULAN DAN SARAN

Berdasarkan penelitian dan hasil pengujian yang telah dilakukan, maka dapat disimpulkan bahwa penelitian ini menghasilkan model dan aplikasi Algoritma Genetika menggunakan parameter jarak dan waktu tempuh, mempertimbangkan biaya perjalanan simetris maupun asimetris serta waktu kunjungan yang diterapkan untuk memberikan rekomendasi rute perjalanan divisi pemasaran STMIK El Rahma Yogyakarta. Hasil pengujian menunjukkan bahwa jumlah individu terbaik adalah 150 individu dalam satu populasi. Kriteria berhenti jika setelah 126 generasi berturut-turut didapatkan nilai *fitness* tertinggi yang tidak berubah dan kombinasi probabilitas crossover dan probabilitas mutasi yang paling optimal adalah {0,3 : 0,7}. Parameter-parameter Algoritma Genetika yang dihasilkan dari pengujian konsisten memberikan solusi yang optimal ataupun mendekati optimal.

Tabel 5. Hasil Pengujian Konsistensi Algoritma Genetika

No.	Banyak Objek / Gen	Solusi	Fitness	Estimasi Jarak dan Waktu
	6 Objek / Gen	1-4-5-3-6-2-1		
1	1,2,3,4,5,6	STMIK El Rahma - MA Muhammadiyah 1 Kota Yogyakarta - SMK Negeri 2 Yogyakarta - SMK Piri 1 Yogyakarta - SMK Negeri 5 Yogyakarta - SMK Muhammadiyah 3 - STMIK El Rahma	0,0048543	18,95 Km 56,45 menit
	7 Objek / Gen	1-4-5-3-6-2-7-1		
2	1,2,3,4,5,6,7	STMIK El Rahma - MA Muhammadiyah 1 Kota Yogyakarta - SMK Negeri 2 Yogyakarta - SMK Piri 1 Yogyakarta - SMK Negeri 5 Yogyakarta - SMK Muhammadiyah 3 - SMK Muhammadiyah 1 Yogyakarta - STMIK El Rahma	0,0050761	19,37 Km 57,28 menit
	8 Objek / Gen	1-8-4-5-3-6-2-7-1		
3	1,2,3,4,5,6,7,8	STMIK El Rahma - Madrasah Aliyah Ali Maksum - MA Muhammadiyah 1 Kota Yogyakarta - SMK Negeri 2 Yogyakarta - SMK Piri 1 Yogyakarta - SMK Negeri 5 Yogyakarta - SMK Muhammadiyah 3 - SMK Muhammadiyah 1 Yogyakarta - STMIK El Rahma	0,0047169	21,15 Km 63,82 menit
	9 Objek / Gen	1-8-4-5-3-6-9-2-7-1		
4	1,2,3,4,5,6,7,8,9	STMIK El Rahma - Madrasah Aliyah Ali Maksum - MA Muhammadiyah 1 Kota Yogyakarta - SMK Negeri 2 Yogyakarta - SMK Piri 1 Yogyakarta - SMK Negeri 5 Yogyakarta - MA Negeri 4 Bantul - SMK Muhammadiyah 3 - SMK Muhammadiyah 1 Yogyakarta - STMIK El Rahma	0,0039682	24,71 Km 75,77 menit
	10 Objek / Gen	1-8-4-10-5-3-6-9-2-7-1		
5	1,2,3,4,5,6,7,8,9,10	STMIK El Rahma - Madrasah Aliyah Ali Maksum - MA Muhammadiyah 1 Kota Yogyakarta - SMK Negeri 7 Yogyakarta - SMK Negeri 2 Yogyakarta - SMK Piri 1 Yogyakarta - SMK Negeri 5 Yogyakarta - MA Negeri 4 Bantul - SMK Muhammadiyah 3 Yogyakarta - SMK Muhammadiyah 1 Yogyakarta - STMIK El Rahma	0,0040983	26,53 Km 81,72 menit

REFERENSI

- [1] Risdiyanto, A. & Kurniyati, Y. (2015). Strategi Pemasaran Perguruan Tinggi Swasta di Kabupaten Sleman Yogyakarta Berbasis Rangsangan Pemasaran. *Jurnal Maksipreneur*, Vol. 5, No. 1, pp. 1–23.
- [2] Kramer, O. (2017). *Genetic Algorithm Essentials*. Cham, Switzerland: Springer International Publishing.
- [3] Jacobson, L. & Kanber, B. (2015). *Genetic Algorithms in Java Basics*. Berkeley, California: Apress Media.
- [4] Priandani, N.D. & Mahmudy, W.F. (2015). Optimasi Travelling Salesman Problem With Time Windows (TSP-TW) pada Penjadwalan Paket Rute Wisata di Pulau Bali Menggunakan Algoritma Genetika. *Prosiding Seminar Nasional Sistem Informasi Indonesia*, pp. 259–266.
- [5] Kumari, J. & Dubey, A.K. (2016). A Review Paper on Genetic Algorithm. *International Journal of Advance Research in Computer Science and Management Studies*. Vol. 4, No. 7, pp. 122–125.
- [6] Shita, R.T. & Subandi. (2017). Implementasi Algoritma Genetika Pada Aplikasi Pemetaan Distribusi Barang Berbasis Web. *Jurnal Telematika MKOM*, Vol. 9, No. 3, pp. 114–118.
- [7] Sholeh, M., Widyastuti, N. & Pratama, M. (2017). Google Map for Implementation of Geographic Information System Development Search Location SMEs. *International Journal of Engineering Research & Technology*, Vol. 6, No. 2, pp. 501–504.
- [8] Abrori, M. & Setiyani, N. (2015). Implementasi Algoritma Best-First Search (BeFS) Pada Penyelesaian Traveling Salesman Problem (TSP) (Studi Kasus: Perjalanan Wisata di Kota Yogyakarta). *Jurnal Fourier*, Vol. 4, No. 2, pp. 93–111.
- [9] Hutami, D.W. & Mahmudy, W.F. (2017). Implementasi Algoritma Nearest Insertion Heuristic dan Modified Nearest Insertion Heuristic Pada

- Optimasi Rute Kendaraan Pengangkut Sampah (Studi Kasus: Dinas Kebersihan dan Pertamanan Kota Malang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 1, No. 2, pp. 95–99.
- [10] Mahmudy, W.F. (2014). Improved Simulated Annealing for Optimization of Vehicle Routing Problem with Time Windows (VRPTW). *Kursor Journal*, Vol. 7, No. 3, pp. 109–116.
- [11] Samana, E., Prihandono, B. & Noviani, E. (2015). Aplikasi Simulated Annealing Untuk Menyelesaikan Travelling Salesman Problem. *Bimaster*, Vol. 03, No. 1, pp. 25–32.
- [12] Purnia, D.S. & Riana, D. (2016). Pencarian Rute Terpendek Perjalan Promosi *Marketing* Menggunakan Algoritma Genetika dan Algoritma Greedy. *Informatika*, Vol. 3, No. 2, pp. 299–313.
- [13] Sivanandam, S.N. & Deepa, S.N. (2008). *Introduction to Genetic Algorithms*, 1ed. Berlin: Springer-Verlag Berlin Heidelberg.
- [14] Dincer, A. & Uraz, B. (2013). *Google Maps JavaScript API Cookbook*. Birmingham: Packt Publishing.