

PROTOTYPE SISTEM PENCARIAN REKOMENDASI LOKASI KERAJINAN LOKAL DI YOGYAKARTA MENGGUNAKAN TRAVELING SALESMAN PROBLEM (TSP) DENGAN ALGORITMA GENETIKA

Handy Eka Putra Anwar¹, Deviani Titi Nautami², Dhina Puspari Wijaya³

*Magister Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia^{1,2,3}
Sleman, Yogyakarta, Indonesia*

E-mail : handyeka@gmail.com¹, devianititi.n@gmail.com², dhina.puspa@gmail.com³

ABSTRACT

Local craft industry is now starting to compete with other industries in the process of advancing the region and even the country's economy. Yogyakarta, as a tourism destination known for its traditional cultural values would be more interesting if the culture of local crafts can be more openly introduced. The high interest in the local craft makes tourists want to see, and even to try the process of making of local crafts in Yogyakarta. Traveling Salesman Problem (TSP) with genetic algorithm method works by providing Yogyakarta's local craft location recommendation, from closest to farthest location that can be visited by tourists with the use of Global Positioning System (GPS) on their smartphones. Traveling Salesman Problem (TSP) with genetic algorithm (GA) method provides the best traveling route for tourists who want to find and visit Yogyakarta's local craft location in one hand.

Keywords : Traveling Salesman Problem (TSP); Genetic Algorithm; Local Craft; Traveling Route; Global Positioning System (GPS).

1. PENDAHULUAN

Ekonomi kreatif mampu meningkatkan kualitas kepariwisataan menjadi daya tarik di destinasi pariwisata, dan tentunya bisa menjadi media promosi bagi pariwisata daerah (Seliari, 2015). Sebaliknya, kemajuan pariwisata suatu daerah umumnya diikuti oleh peningkatan permintaan akan karya kreatif lokal. Konsep kegiatan wisata didefinisikan dengan tiga faktor, yaitu *something to see, something to do, dan something to buy* (Yoeti, 1985). Hal tersebut menjelaskan bahwa kegiatan berwisata adalah suatu yang dapat dilihat, dilakukan, dan dibeli. Yogyakarta, sebuah daerah wisata yang memiliki beragam karya kreatif lokal yang diciptakan oleh banyak industri kreatif kecil, seperti kerajinan perak, batik, bambu, kulit, gerabah, dan sebagainya dituntut untuk mampu menciptakan konsep kegiatan berwisata yang lengkap seperti konsep di atas.

Luas wilayah daerah Yogyakarta yang terbilang tidak kecil, membuat banyak wisatawan terkadang kesulitan untuk menemukan apa saja dan di mana lokasi

kerajinan lokal yang ada di Yogyakarta. Sehingga para wisatawan lebih memilih untuk datang ke pusat oleh-oleh atau pasar sentra hanya untuk membeli hasil kerajinan lokal tersebut saja. Hal tersebut tentunya mengurangi 3 konsep kegiatan berwisata yang diharapkan. Dan saat ini, bagaimana solusi supaya para wisatawan dapat langsung mengunjungi langsung lokasi para pengrajin industri kreatif kerajinan lokal di Yogyakarta.

Salah satu metode untuk mencari rute terpendek adalah *Traveling Salesman Problem* (Hardianti, 2013). Metode *Traveling Salesman Problem* (TSP) dengan algoritma genetika bekerja dengan memberikan solusi rekomendasi di mana saja lokasi kerajinan lokal yang ada di Yogyakarta. TSP tidak hanya bertujuan untuk menemukan jarak mana yang paling dekat hingga terjauh untuk dikunjungi oleh wisatawan, tetapi juga memiliki urutan titik yang harus dikunjungi terlebih dahulu sebelum titik yang lain.

Algoritma genetika merupakan metode yang adaptif yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam suatu masalah optimasi (Kusumadewi, 2003).

Algoritma genetika akan bekerja di dalam suatu populasi yang terdiri dari individu-individu yang dilambangkan dengan suatu nilai *fitness* untuk mencari solusi terbaik dari persoalan ini.

TSP dengan algoritma genetika membantu untuk mengetahui apa saja dan di mana saja lokasi kerajinan lokal yang ada di Yogyakarta dalam satu genggam *smartphone* mereka dengan memanfaatkan *Global Positioning System* (GPS) untuk mendeteksi di mana lokasi mereka berada sehingga dapat ditentukan di mana saja titik-titik kerajinan lokal yang telah dikategorikan menurut jenis kerajinan lokal yang ada di Yogyakarta. Hal tersebut tentu diharapkan dapat memberikan kemudahan alur lokasi terbaik berwisata bagi para wisatawan yang ingin berkunjung langsung ke lokasi kerajinan lokal di Yogyakarta.

2. TINJAUAN PUSTAKA

2.1. Greedy Heuristic

Pemilihan lintasan pada *greedy heuristic* dimulai pada lintasan yang memiliki nilai paling minimum (Amin, 2006). Algoritma yang juga awam disebut dengan algoritma *Nearest Neighbour* ini, akan memilih kota selanjutnya yang belum dikunjungi berdasarkan jarak terminimum dari kota pertama yang berhasil dicapai untuk dikunjungi.

Algoritma *greedy* memiliki tingkat kompleksitas yang sangat mengagumkan yaitu $O(n)$, namun hasil yang akhirnya didapat akan sangat jauh dari hasil teroptimal. Semakin banyak kota yang akan dikunjungi maka semakin besar pula perbedaan hasil yang dicapai.

2.2. Traveling Salesman Problem (TSP)

Traveling Salesman Problem (TSP) adalah sebuah persoalan optimasi untuk mencari rute terpendek bagi seorang pedagang keliling (*salesman*) (Lamabelawa, 2006). Persoalan optimasi yang ingin dicapai ialah mendapatkan nilai paling minimum dari rute yang dilalui.

Salah satu metode yang dipastikan bisa mendapatkan solusi paling optimal dari TSP adalah penghitungan menyeluruh dan evaluasi. Prosedur ini dimulai dengan membangkitkan terlebih dahulu berbagai kemungkinan dari semua perjalanan dan mengevaluasinya sesuai panjang perjalanan. Perjalanan dengan panjang terkecil dipilih sebagai yang terbaik, dan dijamin akan optimal (Homaifar, 1992).

2.3. Algoritma Genetika

Algoritma Genetika adalah salah satu cabang *evolutionary algorithms*, yaitu suatu teknik optimasi yang didasarkan pada genetika alami. Untuk menghasilkan suatu solusi optimal, algoritma genetika melakukan proses pencarian di antara sejumlah alternatif titik optimal berdasarkan fungsi probabilitistik (Michalewicz, 1996).

Algoritma genetika merupakan salah satu algoritma yang muncul untuk menyelesaikan persoalan TSP. Masalah utama pada algoritma genetika adalah tentang bagaimana cara untuk memetakan satu masalah menjadi satu string kromosom (Widodo, 2010). Berikut adalah alur proses dari algoritma genetika :

2.3.1. Jarak Antar Kota.

Dalam TSP, permasalahannya adalah mencari jarak terpendek dari lokasi kerajinan lokal yang akan dilalui. Untuk TSP, permasalahannya juga untuk meminimalkan total biaya. Oleh karena itu, nilai *fitness* yang bisa digunakan adalah 1 dibagi dengan total biaya. Dalam hal ini yang dimaksud total biaya adalah jumlah jarak antara satu lokasi kerajinan lokal dengan lokasi kerajinan lokal yang lainnya. Untuk menghitung jarak dari satu lokasi ke lokasi yang lain digunakan rumus jarak kartesian yaitu :

$$\|A - B\| = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2} \dots\dots(1)$$

Rumus di atas menjelaskan di mana (X_A, Y_A) menyatakan koordinat kota A dan (X_B, Y_B) menyatakan koordinat kota B (Rachmayadi, 2008).

2.3.2. Hitung Nilai *Fitness*

TSP bertujuan untuk meminimalkan jarak, maka nilai *fitness*-nya adalah inversi dari total jarak dari jalur yang didapatkan dengan menggunakan rumus berikut :

$$Fitness = \frac{1}{x} \dots\dots\dots(2)$$

Nilai x dalam rumus (2) adalah total jarak dari jalur yang didapatkan. Variabel yang digunakan untuk mencari nilai *fitness* yaitu populasi, jumlah gen, dan jarak antar kota dalam suatu kromosom. Keluaran dari fungsi ini adalah nilai *fitness* dalam suatu populasi.

Algoritma genetika bertujuan mencari individu dengan nilai *fitness* yang paling tinggi. Umumnya kromosom ber-*fitness* tinggi akan bertahan dan berlanjut kegenerasi berikutnya (Arzain, 2009).

2.3.3. Hitung Probabilitas *Fitness*

Probabilitas *fitness* adalah perhitungan masing-masing nilai *fitness* pada setiap kromosom dalam suatu populasi terhadap jumlah total nilai *fitness*-nya. Rumus yang digunakan adalah

$$Probabilitas Fitness = \frac{Nilai Fitness}{Total Fitness} \dots\dots\dots(3)$$

Pada tahap ini juga dapat ditentukan nilai kumulatif dari probabilitasnya seperti pada rumus (3) dimana nilai *fitness* akan dibagikan dengan total *fitness*.

2.3.4. Roulette Wheel Selection

Pada proses seleksi *roulette wheel* akan dicari kromosom terbaik dalam satu generasi, dimana kromosom terbaik dapat dilihat dari nilai *fitness*. Setiap kromosom akan dievaluasi berdasarkan nilai *fitness*-nya. Tahap pertama yang dilakukan adalah nilai *fitness* yang telah diperoleh dijumlahkan, kemudian membangkitkan bilangan random untuk dibandingkan dengan nilai *fitness* yang telah terurut. Jika hasil dari nilai *fitness* dibagi dengan total *fitness* > bilangan random yang telah dibangkitkan, maka kromosom tersebut akan terpilih sebagai induk untuk melakukan proses selanjutnya.

2.3.5. Crossover / Perkawinan Silang

Pada tahap *crossover* akan dipilih dua kromosom induk yang akan mengalami perkawinan silang secara acak, kemudian tentukan titik potongnya. Setelah itu lakukan pertukaran informasi dari kedua kromosom tersebut berdasarkan titik potong yang telah ditentukan.

2.3.6. Global Positioning System (GPS)

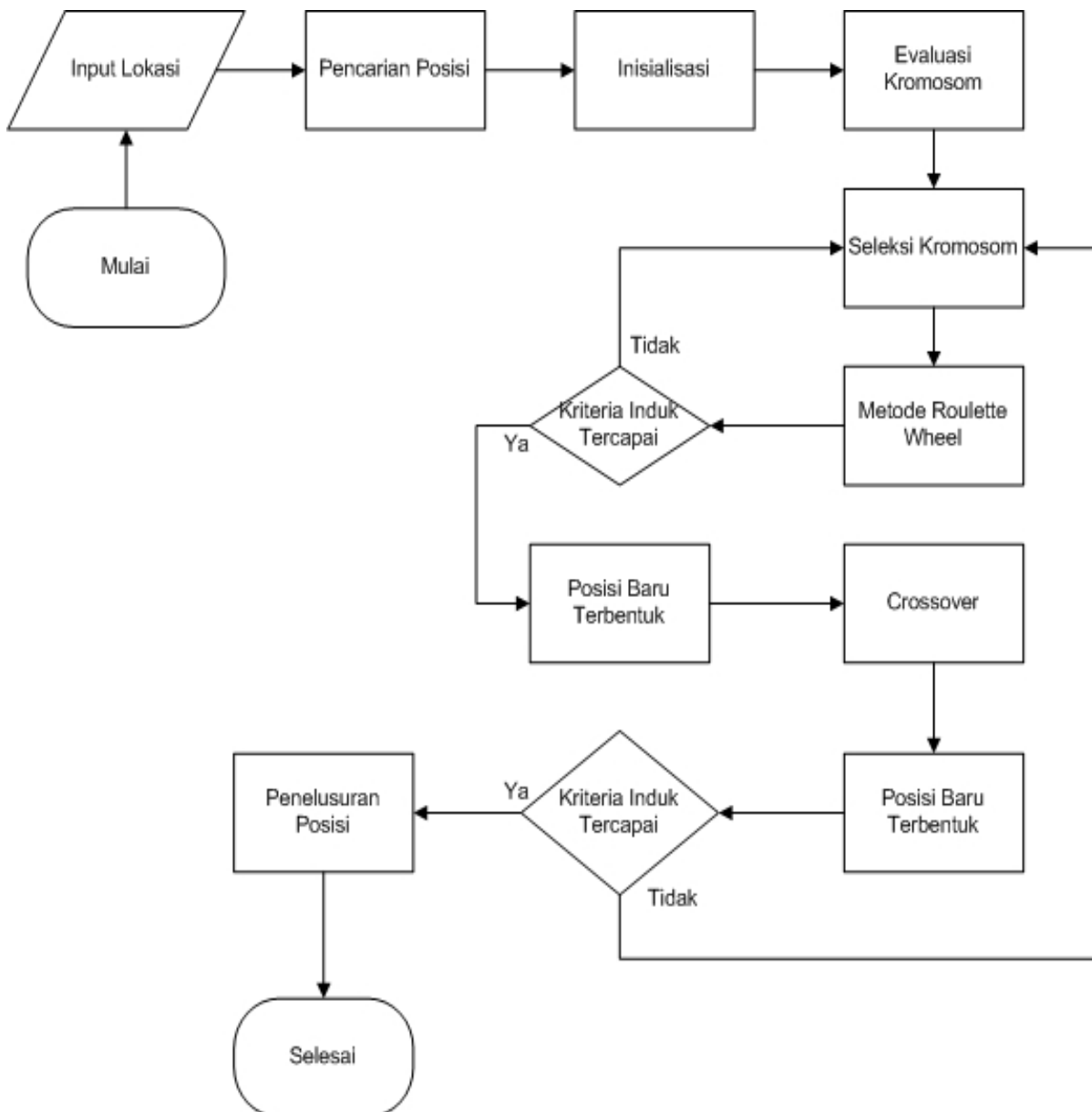
Menurut Wikipedia, *Global Positioning System* (GPS) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. GPS menyediakan posisi dengan ketepatan akurasi hingga 15 meter dalam memberikan koordinat terhadap suatu lokasi tertentu. Ketepatan GPS bergantung pada lokasi GPS *receiver* dan halangan terhadap sinyal satelit GPS (Scot, 2009).

Penggunaan GPS dalam sistem ini begitu vital karena GPS berfungsi untuk mendeteksi posisi atau titik-titik lokasi kerajinan lokal yang ada di Yogyakarta yang perlu dikunjungi oleh wisatawan. *Latitude* dan *longitude* menjadi nilai dari posisi yang akan digunakan untuk menjadi variabel dalam perhitungan algoritma genetika pada prosesnya.

Google Maps digunakan sebagai media pencarian lokasi karena kelebihan dari *Google Maps* yang memiliki fitur - fitur yang mendukung dalam penelitian ini seperti fitur *direction* yang berfungsi sebagai navigator dan memberikan jalur penyelesaian dari satu tempat tujuan ke tempat tujuan lain. Pada permasalahan TSP di penelitian ini, fungsi *direction* digunakan untuk membentuk *path* dari satu lokasi kerajinan lokal ke lokasi kerajinan lokal lainnya.

3. METODE PENELITIAN

Gambar 1 di bawah ini adalah diagram alir metodologi penelitian pada penelitian ini. Metodologi penelitian ini menjelaskan singkat alur kerja yang dilakukan secara menyeluruh melalui tahap - tahap penelitian yang akan dilaksanakan.



Gambar 1. Alur Metodologi Penelitian.

Terdapat dua hak akses dalam aplikasi ini yaitu *admin* dan *user*. *Admin* bertugas menambah, mengubah, dan menghapus data - data kerajinan lokal dengan terlebih dahulu memasukkan *username* dan *password*. Data - data tersebut berupa; jenis kerajinan, nama toko kerajinan lokal, alamat, profil singkat, dan keterangan. *User* yaitu wisatawan yang menggunakan aplikasi ini dapat memilih lokasi tujuan toko kerajinan. *User* pun dapat memberi *rating* secara otomatis terhadap lokasi - lokasi kerajinan lokal sehingga dapat diketahui toko kerajinan mana yang paling sering dikunjungi berdasarkan toko kerajinan

yang dipilih di setiap kali proses TSP berjalan.

Dalam penelitian, hanya digunakan 4 tujuan lokasi kerajinan lokal sebagai contoh proses TSP yang dilakukan dengan menggunakan algoritma genetika. Proses penelitian seperti yang digambarkan pada Gambar 1 di atas dimulai dengan memasukkan lokasi - lokasi kerajinan lokal mana yang akan ditelusuri oleh wisatawan, di mana wisatawan bertindak sebagai *user* dari sistem. Setelah ditentukan lokasi kerajinan mana yang akan dipilih, kemudian akan

Tabel 1. Kromosom Hasil *Encoding*

Kromosom [1]	[kayu1 rotan1 kulit1 perak1]
Kromosom [2]	[rotan1 kayu1 kulit1 perak1]
Kromosom [3]	[perak1 kayu1 rotan1 kulit1]
Kromosom [4]	[kulit1 kayu1 perak1 rotan1]
Kromosom [5]	[kulit1 perak1 kayu1 rotan1]
Kromosom [6]	[perak1 rotan1 kulit1 kayu1]

Tabel 2. Titik Latitude dan *Longitude*

Koordinat	Latitude	Longitude
Titik awal	-7,687671	110,413776
batik1	-7,809620	110,384588
kulit1	-7,812938	110,377473
rotan1	-7,807684	110,385098
kayu1	-7,801786	110,352071
perak1	-7,829187	110,399054

Tabel 3. Nilai *Fitness* Kromosom

<i>Fitness</i> [1]	221,012429
<i>Fitness</i> [2]	441,700088
<i>Fitness</i> [3]	441,675951
<i>Fitness</i> [4]	441,672550
<i>Fitness</i> [5]	441,679736
<i>Fitness</i> [6]	221,005242

dilakukan proses pencarian posisi berupa nilai *latitude* dan *longitude* tiap-tiap lokasi.

Proses algoritma genetika dimulai dengan melakukan inisiasi, mengevaluasi kromosom, seleksi, dan pindah silang sampai menemukan nilai yang paling optimum (terkecil) untuk kemudian dilakukan proses penelusuran posisi - posisi dari lokasi wisatawan.

4. HASIL DAN PEMBAHASAN

Terdapat 4 buah lokasi kerajinan lokal di Yogyakarta yang akan dilalui oleh seorang wisatawan. Persoalan TSP tersebut akan diselesaikan menggunakan algoritma genetika. Kriteria berhenti ditentukan terlebih dahulu yaitu apabila setelah beberapa

generasi berturut-turut diperoleh nilai *fitness* yang terendah tidak berubah. Pemilihan nilai *fitness* yang terendah sebagai syarat karena nilai tersebut yang merepresentasikan jarak terdekat yang dicari pada persoalan TSP ini.

Untuk mempermudah perhitungan algoritma genetika, lokasi yang akan menjadi gen dalam kromosom selain lokasi asal, akan diinisialisasikan seperti berikut :

kayu1 = Kerajinan Kayu Bodroyono.
 kayu2 = Harry Art.
 rotan1 = Industri Kerajinan Ponindah.
 rotan2 = Kerajinan Takir Sudi.
 kulit1 = Tio Handicraft.
 kulit2 = Sagita Bag.
 perak1 = Nugroho Silver.
 perak2 = Tom's Silver.

4.1. Inisialisasi

Tabel 1 adalah proses pemilihan secara *random* 6 buah populasi dalam satu generasi.

4.2. Evaluasi Kromosom

Sebelum melakukan proses evaluasi kromosom, dicari terlebih dahulu titik *latitude* dan *longitude* dari masing - masing lokasi seperti pada tabel 2.

Kemudian dilakukan proses perhitungan nilai *fitness* masing-masing kromosom yang telah dibangkitkan seperti pada tabel 3.

4.3. Seleksi Kromosom (Q)

Pada persoalan TSP, yang diinginkan yaitu mendapatkan kromosom dengan nilai *fitness* yang lebih kecil.

$$Q[i] = 1/ Fitness \dots\dots\dots(2)$$

$$Q[1] = 1/221.012429 = 0,004524632$$

$$Q[2] = 1/441,700088 = 0,002263980$$

$$Q[3] = 1/441,675951 = 0,002264103$$

$$Q[4] = 1/441,672550 = 0,002264121$$

$$Q[5] = 1/441,679736 = 0,002264084$$

$$Q[6] = 1/221,005242 = 0,004524780$$

$$\begin{aligned} \text{Total Fitness} &= 0,004524632 + 0,00226398 \\ &+ 0,002264103 + 0,002264121 \\ &+ 0,002264084 + 0,00452478 \\ &= 0,0181057 \end{aligned}$$

4.4. Probabilitas (P)

Kemudian, untuk mencari nilai probabilitas (P) akan digunakan rumus berikut. Q adalah nilai seleksi kromosom, dan Tq adalah Total *Fitness*.

$$P[i] = Q[i] / Tq \dots \dots \dots (3)$$

P[1] = 0,004524632/ 0,0181057 = 0,2499010
 P[2] = 0,002263980/ 0,0181057 = 0,1250424
 P[3] = 0,002264103/ 0,0181057 = 0,1250492
 P[4] = 0,002264121/ 0,0181057 = 0,1250502
 P[5] = 0,002264084/ 0,0181057 = 0,1250481
 P[6] = 0,004524780/ 0,0181057 = 0,2499091

4.5. Roulette Wheel (R)

Untuk proses seleksi, digunakan metode *roulette wheel*, untuk itu, terlebih dahulu dicari nilai kumulatif (C) dari probabilitas (P).

C[1] = 0,2499010
 C[2] = 0,2499010 + 0,1250424 = 0,3749434
 C[3] = 0,3749434 + 0,1250492 = 0,4999926
 C[4] = 0,4999926 + 0,1250502 = 0,6250427
 C[5] = 0,6250427 + 0,1250481 = 0,7500909
 C[6] = 0,7500909 + 0,2499091 = 1,0000000

Proses *roulette wheel* adalah membangkitkan nilai acak R antara 0-1, di mana jika R[k]<C[k] maka kromosom ke-k sebagai induk. Selain itu pilih kromosom ke-k sebagai induk dengan syarat C[k-1]<R[k]<C[k]. Metode *roulette wheel* akan berputar sebanyak jumlah kromosom, yaitu 6 kali (membangkitkan bilangan acak R). Hasil dari *roulette wheel* adalah sebagai berikut :

R[1] 0,87135159
 R[2] 0,453641842
 R[3] 0,07105522
 R[4] 0,132659835
 R[5] 0,234372073
 R[6] 0,598647236

Setelah proses *roulette wheel* selesai, maka populasi (K) baru akan terbentuk, yaitu :

K [1] = [2] = [rotan1 kayu1 kulit1 perak1]
 K [2] = [3] = [perak1 kayu1 rotan1 kulit1]
 K [3] = [1] = [kayu1 rotan1 kulit1 perak1]
 K [4] = [4] = [kulit1 kayu1 perak1 rotan1]
 K [5] = [5] = [kulit1 perak1 kayu1 rotan1]
 K [6] = [6] = [perak1 rotan1 kulit1 kayu1]

4.6. Crossover (Pindah Silang)

Pada skema ini, satu bagian kromosom dipertukarkan dengan tetap menjaga urutan lokasi yang bukan bagian dari kromosom tersebut. Kromosom yang dijadikan induk dipilih secara acak dan jumlah kromosom yang dipindahsilangkan dipengaruhi oleh parameter *crossover probability* (pc).

Misal ditentukan pc = 25%, kemudian kita bangkitkan bilangan random R sebanyak jumlah populasi yaitu 6 kali.

R[1] = 0,60228
 R[2] = 0,49958
 R[3] = 0,59255
 R[4] = 0,020096
 R[5] = 0,40415
 R[6] = 0,18409

Kromosom ke - k dipilih sebagai induk jika R[k]<pc. Maka yang dijadikan induk adalah kromosom[4] dan kromosom[6]. Setelah melakukan pemilihan induk, proses selanjutnya adalah menentukan posisi *crossover*.

Kromosom[4]=Kromosom[6]xKromosom[4]
 = [perak1 rotan1 kulit1 kayu1]
 x [kulit1 kayu1 perak1 rotan1]
 = [perak1 kayu1 kulit1 rotan1]

Kromosom[6]=Kromosom[4]xKromosom[6]
 =[kulit1 kayu1 perak1 rotan1]
 x [perak1 rotan1 kulit1 kayu1]
 = [kulit1 perak1 rotan1 kayu1]

Kemudian hasil populasi terbaru setelah dilakukan proses *crossover* :

Kromosom[1]=[rotan1 kayu1 kulit1 perak1]
 Kromosom[2]=[perak1 kayu1 rotan1 kulit1]
 Kromosom[3]=[kayu1 rotan1 kulit1 perak1]
 Kromosom[4]=[perak1 kayu1 kulit1 rotan1]

Kromosom[5]=[kulit1 perak1 kayu1 rotan1]
 Kromosom[6]=[kulit1 perak1 rotan1 kayu1]

Proses algoritma genetika untuk generasi pertama telah selesai. Maka nilai *fitness* setelah 1 generasi adalah :

$$\begin{aligned} \text{Fitness}[1] &= (\text{lokasi} + \text{rotan1}) + (\text{rotan1} + \text{kayu1}) + \\ & (\text{kayu1} + \text{kulit1}) + (\text{kulit1} + \text{perak1}) + \\ & (\text{perak1} + \text{lokasi}) \\ &= 220,7989066 + 0,033549503 + \\ & 0,027742183 + 0,027014247 + \\ & 220,8128753 \\ &= \mathbf{441,7000879} \end{aligned}$$

$$\begin{aligned} \text{Fitness}[2] &= (\text{lokasi} + \text{perak1}) + (\text{perak1} + \text{kayu1}) \\ & + (\text{kayu1} + \text{rotan1}) + (\text{rotan1} + \text{kulit1}) \\ & + (\text{kulit1} + \text{lokasi}) \\ &= 220,8128753 + 0,054389494 + \\ & 0,033549503 + 0,009259867 + \\ & 220,7658765 \\ &= \mathbf{441,6759507} \end{aligned}$$

$$\begin{aligned} \text{Fitness}[3] &= (\text{lokasi} + \text{kayu1}) + (\text{kayu1} + \text{rotan1}) + \\ & (\text{rotan1} + \text{kulit1}) + (\text{kulit1} + \text{perak1}) + \\ & (\text{perak1} + \text{lokasi}) \\ &= 0,129729489 + 0,033549503 + \\ & 0,009259867 + 0,027014247 + \\ & 220,8128753 \\ &= \mathbf{221,0124285} \end{aligned}$$

$$\begin{aligned} \text{Fitness}[4] &= (\text{lokasi} + \text{perak1}) + (\text{perak1} + \text{kayu1}) \\ & + (\text{kayu1} + \text{kulit1}) + (\text{kulit1} + \text{rotan1}) \\ & + (\text{rotan1} + \text{lokasi}) \\ &= 220,8128753 + 0,054389494 + \\ & 0,033549503 + 0,009259867 + \\ & 220,7658765 \\ &= \mathbf{441,6759507} \end{aligned}$$

$$\begin{aligned} \text{Fitness}[5] &= (\text{lokasi} + \text{kulit1}) + (\text{kulit1} + \text{perak1}) + \\ & (\text{perak1} + \text{kayu1}) + (\text{kayu1} + \text{rotan1}) + \\ & (\text{rotan1} + \text{lokasi}) \\ &= 220,7658765 + 0,027014247 + \\ & 0,054389494 + 0,033549503 + \\ & 220,7989066 \\ &= \mathbf{441,6797364} \end{aligned}$$

$$\begin{aligned} \text{Fitness}[6] &= (\text{lokasi} + \text{kulit1}) + (\text{kulit1} + \text{perak1}) + \\ & (\text{perak1} + \text{rotan1}) + (\text{rotan1} + \text{kayu1}) + \\ & (\text{kayu1} + \text{lokasi}) \\ &= 220,7658765 + 0,027014247 + \\ & 0,025634917 + 0,033549503 + \\ & 0,129729489 \\ &= \mathbf{220,9818046} \end{aligned}$$

Sebelumnya telah ditentukan kriteria berhenti, yaitu jika setelah dalam beberapa generasi berturut - turut diperoleh nilai *fitness* yang terendah tidak berubah. Pada 1 generasi telah terlihat bahwa terdapat nilai *fitness* terkecil yang tidak berubah. Apabila dilanjutkan hingga generasi ke-N maka diyakinkan bahwa nilai *fitness* yang terendah tetap tidak akan berubah. Walaupun perhitungan cukup dijabarkan hingga generasi ke-1 saja namun solusi yang mendekati optimal telah didapatkan (Rachmayadi, 2008).

Dari proses perhitungan dengan menggunakan algoritma genetika, proses perhitungan berhenti pada proses *crossover* di mana didapatkan solusi terbaik dengan nilai terkecil yaitu pada nilai *fitness*[6]= 220,9818046, sehingga kromosom[6] akan menjadi induk dari runtutan jalur yang harus dilalui.

Jadi, rute terbaik ketika wisatawan ingin mengunjungi 4 lokasi pilihannya tersebut adalah Tio Handicraft → Nugroho Silver → Industri Kerajinan Ponindah → Kerajinan Kayu Bodroyono.

4.7. Prototype Sistem

Prototype dari Sistem Pencarian Rekomendasi Lokasi Kerajinan Lokal di Yogyakarta menggunakan *Traveling Salesman Problem* (TSP) dengan Algoritma Genetika ini dirancang khusus untuk sistem aplikasi android seperti berikut :

4.8. Halaman Muka

Gambar 2 di bawah adalah *prototype* halaman muka aplikasi sistem kerajinan lokal Yogyakarta.



Gambar 2. Halaman Muka.

4.9. Halaman Kategori

Pada Gambar 3, 4 adalah *prototype* dari halaman kategori jenis kerajinan lokal di Yogyakarta, di mana wisatawan dapat menentukan kerajinan apa yang diinginkan sesuai kategori yang telah ditentukan sebanyak maksimal 4 lokasi kerajinan.



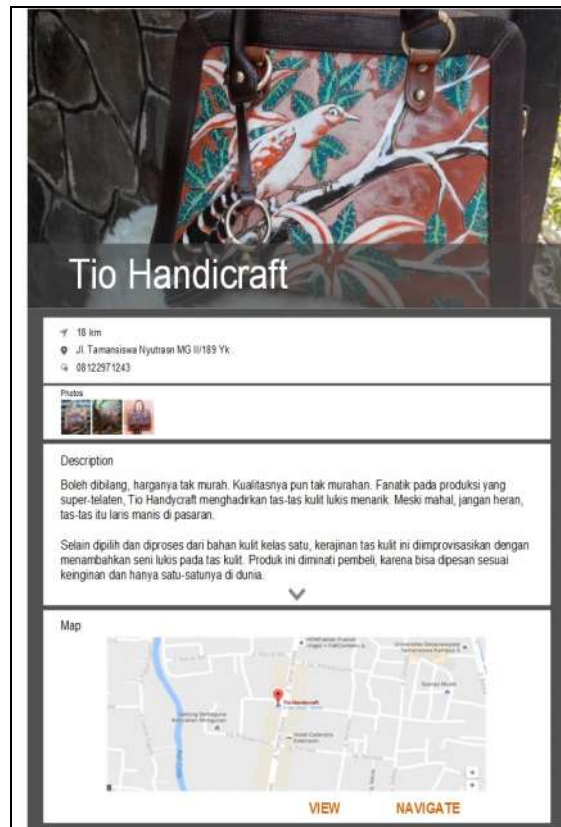
Gambar 3. Halaman Kategori.



Gambar 4. Halaman Isi Kategori.

4.10. Halaman Profil Tempat Kerajinan Lokal

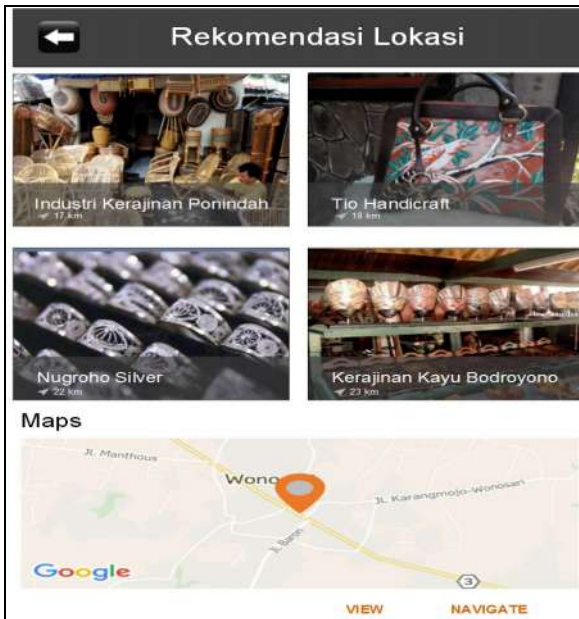
Gambar 5 berikut adalah *prototype* halaman profil dari tempat kerajinan yang bisa dilihat oleh wisatawan untuk mengetahui secara singkat informasi tentang tempat-tempat kerajinan lokal di Yogyakarta.



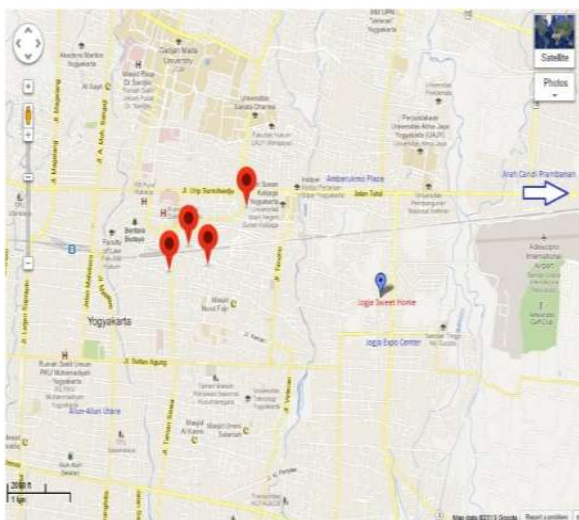
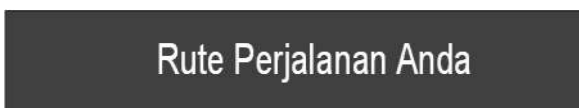
Gambar 5 Halaman Profil Tempat Kerajinan Lokal.

4.11. Halaman Rekomendasi Lokasi

Hasil akhir dari sistem ini adalah seperti yang ada pada Gambar 6, 7 di mana terdapat rekomendasi terbaik untuk melakukan perjalanan wisata kerajinan lokal. Dengan bantuan GPS yang telah diaktifkan pada *smartphone user*, maka sistem akan memberikan rute perjalanan sesuai lokasi di mana *user* berada.



Gambar 6. Halaman Rekomendasi Lokasi.



Gambar 7. Halaman Rute Perjalanan.

5. KESIMPULAN

Kesimpulan yang didapat dari proses pencarian rekomendasi lokasi kerajinan lokal di Yogyakarta menggunakan *Traveling Salesman Problem (TSP)* dengan algoritma genetika beserta hasil *prototype* dari implementasi metode pada sistem ini adalah :

1. Algoritma genetika memperoleh jalur terbaik pada proses *crossover* dengan nilai *fitness* sebagai hasil terendah yaitu 220,9818046.
2. Rekomendasi jalur terbaik yang dapat dilalui oleh wisatawan adalah Tio Handicraft → Nugroho Silver → Industri Kerajinan Ponindah → Kerajinan Kayu Bodroyono.
3. Algoritma Genetika dapat di implementasikan di dalam sistem *smartphone* yang dilengkapi dengan fasilitas *Global Positioning System (GPS)* yang diaktifkan.

DAFTAR PUSTAKA

- A. Homaifar, S. Guan, dan G.E. Liepins, "Schema analysis of the traveling salesman problem using genetic algorithms," *Complex Systems*, 1992, 6920, pp 183-217.
- A, Yoeti, "Pengantar Ilmu Pariwisata," Bandung, Penerbit Angkasa, 1985, p 64.
- A.W. Widodo, dan W.F. Mahmudy, 2010 "Penerapan Algoritma Genetika pada Sistem Rekomendasi Wisata Kuliner," FMIPA, Universitas Brawijaya, Malang, Indonesia, pp 205-211.
- D. Scot, 2009. "Google Maps API V2," Texas: Rayleigh, pp 50-88.
- I. Lamabelawa, 2006. "Penyelesaian TSP dengan Algoritma Genetik," jawawuan.web.ugm.ac.id/tspalgen.pptx.
- J. Arzain, 2009. "Implementasi Algoritma Genetika untuk *Travelling Salesman Problem (TSP)*," FMIPA, Universitas Sumatera Utara, Medan, pp 8-17.

- S. Kusumadewi, 2003. "Artificial Intelligence: Teknik dan Aplikasinya," Yogyakarta: Graha Ilmu, p 278.
- T. Seliari, 2015. "Perubahan Pola Ruang di Kawasan Sentra Industri Kreatif Kerajinan Gerabah Kasongan," Magister Arsitektur Pariwisata, Universitas Gadjah Mada, Indonesia, p 1.
- T. Rachmayadi, 2008. "Pencarian Solusi TSP (*Traveling Salesman Problem*) menggunakan Algoritma Genetik," Scientific Jurnal, Institut Teknologi Bandung, Bandung, Indonesia, pp 1-5.
- Y. Hardianti, dan Purwanto, 2013. "Penerapan Algoritma Genetika dalam Penyelesaian Travelling Salesman Problem with Precedence Constraints (TSPPC)," Universitas Malang, Indonesia, pp 1-2.
- Z. Michalewicz, 1996. "Genetic Algorithms + Data Structures = Evolution Programs" Heidelberg : Springer - Velag, pp 11-88.