

BAB III

METODOLOGI

3.1 Pengumpulan Data

Data yang dibutuhkan untuk pengembangan sistem berupa jarak dari lokasi satu ke lokasi yang lain dan koordinat semua lokasi. Data tersebut diambil dari google dengan memanfaatkan layanan Google API. Jarak dari lokasi satu ke lokasi lainnya diperlukan untuk mencari solusi TSP sehingga aplikasi dapat membuat *itinerary wisata*. Sedangkan koordinat lokasi digunakan untuk proses klustering. Data baru akan diperoleh setelah sistem menerima masukan.

3.2 Analisis Kebutuhan

Analisis kebutuhan terdiri dari apa saja yang dibutuhkan untuk mengembangkan aplikasi berdasarkan kebutuhan masukan, kebutuhan proses, kebutuhan keluaran, kebutuhan perangkat keras, dan kebutuhan perangkat lunak.

3.3 Analisis Kebutuhan Masukan

Berdasarkan hasil analisis dari kebutuhan masukan, maka dapat diketahui apa saja masukan yang diperlukan oleh aplikasi sebagai berikut :

- a. Pengguna memasukan jumlah hari untuk berwisata
- b. Pengguna memasukan lokasi awal untuk memulai wisata
- c. Pengguna memasukan destinasi wisata yang ingin dikunjungi

3.3.1 Analisis Kebutuhan Proses

Berdasarkan hasil analisis kebutuhan proses, maka aplikasi memerlukan proses sebagai berikut :

- a. Proses mengakses halaman pada aplikasi.
- b. Proses mengakses lokasi awal yang diinginkan pengguna.
- c. Proses mengakses lokasi destinasi wisata.
- d. Proses membuat itinerary wisata.
- e. Proses menampilkan itinerary wisata.

3.3.2 Analisis Kebutuhan Keluaran

Berdasarkan hasil analisis kebutuhan keluaran, maka aplikasi diharapkan akan memberikan hasil keluaran sebagai berikut :

- a. Halaman pertama dapat menampilkan dan menjalankan semua fitur yang tersedia.
- b. Halaman kedua dapat memberikan dan menampilkan hasil pembuatan itinerary wisata.

3.3.3 Analisis Kebutuhan Perangkat Keras

Kebutuhan Perangkat keras bertujuan untuk pengujian performa aplikasi. Pengujian performa aplikasi akan dilakukan pada 3 perangkat komputer yang memiliki spesifikasi yang berbeda-beda. Kebutuhan perangkat keras setiap komputer meliputi prosesor dan RAM.

3.3.4 Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang dibutuhkan untuk mengembangkan aplikasi adalah sebagai berikut :

- a. Text Editor
Text editor untuk menulis kode.
- b. Web Browser
Browser untuk mengakses aplikasi.
- c. Django
Web Framework untuk bahasa python sekaligus menjadi web server.
- d. Python 3.7
Bahasa pemrograman python.

3.4 Perancangan dan Pemodelan Sistem

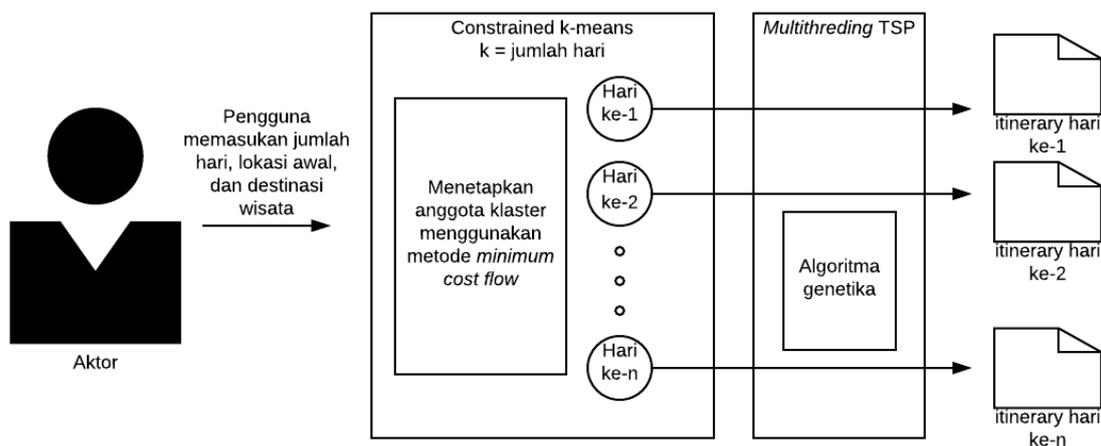
Setelah melakukan analisis sistem tahap berikutnya adalah perancangan dan pemodelan sistem. Pemodelan dan perancangan dalam sistem ini berupa pemodelan *framework* sistem, *flowchart*, dan perancangan antarmuka.

3.4.1 Kerangka Kerja Sistem Itinerary Wisata

Kerangka kerja berfungsi untuk memodelkan sistem. Kerangka ini dibutuhkan untuk membantu dalam pembuatan aplikasi sesuai dengan analisis kebutuhan sistem. Dari analisis yang dilakukan terdapat 3 hal yang menjadi kebutuhan dalam pembuatan sistem, kebutuhan tersebut adalah kebutuhan masukan, kebutuhan proses, dan kebutuhan keluaran. Kebutuhan masukan berupa jumlah hari wisatawan ingin berwisata, lokasi awal, dan destinasi wisata yang ingin dikunjungi oleh wisatawan.

Lokasi awal akan menjadi *starting point* pada itinerary di setiap harinya. Lokasi awal dan destinasi wisata dapat diisi dengan memasukkan alamat yang benar pada field yang disediakan. Serta destinasi wisata akan di klaster berdasarkan jumlah hari. Sistem ini tidak menggunakan database, semua data yang dibutuhkan diambil dari google dengan menggunakan Google API. Data yang dibutuhkan berupa matriks jarak untuk menghitung jarak antar kota dan koordinat setiap lokasi untuk proses klastering.

Terdapat 2 proses utama yang dilakukan oleh sistem berdasarkan data yang dimasukan yaitu proses klastering dan proses pembuatan itinerary. Proses pembuatan itinerary akan dijalankan menggunakan teknik *multithreading* dimana jumlah *thread* sama dengan jumlah hari. Kerangka kerja dari sistem dapat dilihat pada Gambar 3. 1.



Gambar 3. 1 Kerangka Kerja Sistem

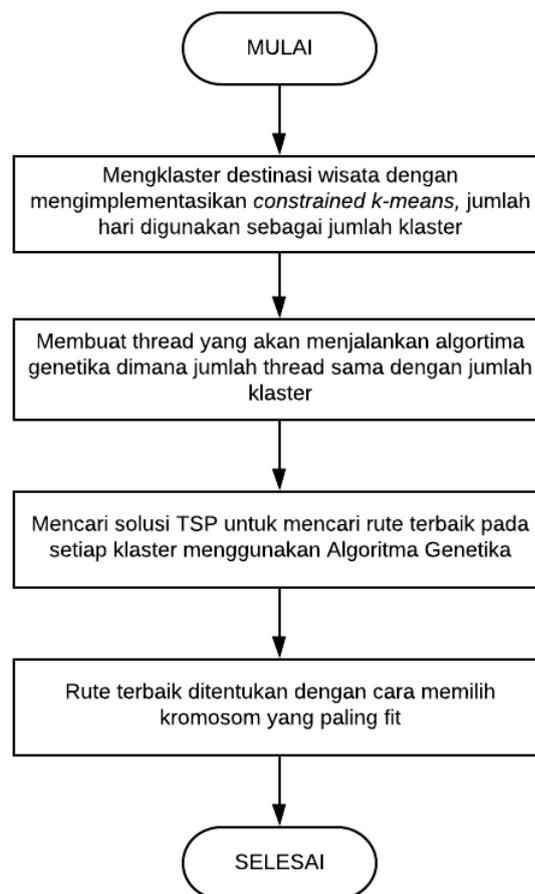
Dari gambar 4 dapat dilihat bahwa terdapat 2 modul utama yaitu proses klastering dan proses pembuatan itinerary. Proses klastering dilakukan dengan *constrained k-means* untuk menghindari klaster tidak memiliki anggota atau klaster memiliki anggota yang sangat sedikit atau tidak seimbang. Kemudian untuk proses pembuatan itinerary menggunakan metode TSP

yang diselesaikan dengan Algoritma Genetika. Proses pembuatan itinerary ini juga diimplementasikan teknik *multithreading* untuk mengurangi waktu proses.

3.4.2 Flowchart Sistem Itinerary Wisata

Flowchart merupakan diagram yang menggambarkan langkah-langkah proses atau alur algoritma penyelesaian pada suatu masalah sehingga mudah untuk dipahami. Terdapat empat *flowchart* yang menggambarkan proses pada sistem yaitu *flowchart* detail pembuatan itinerary, *flowchart* proses detail pada proses klustering, dan *flowchart* proses detail pada proses mencari solusi TSP menggunakan Algoritma Genetika

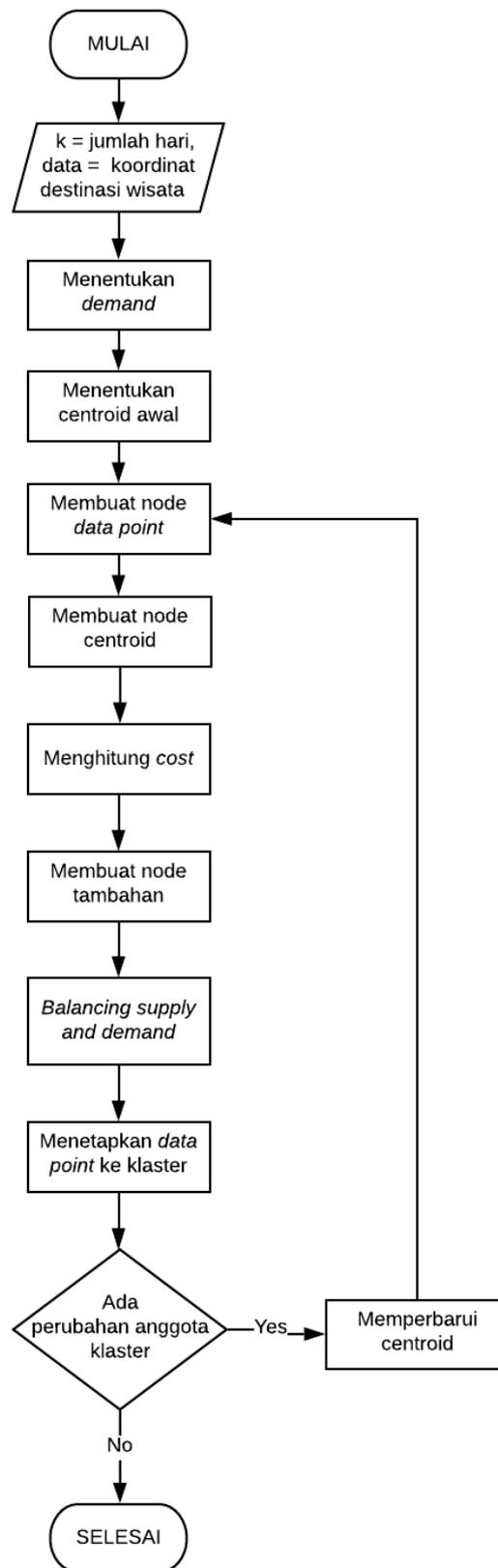
Flowchart pertama menggambarkan proses detail pembuatan rekomendasi itinerary dengan mencari solusi TSP menggunakan Algoritma Genetika. Proses tersebut dapat dilihat pada Gambar 3. 2.



Gambar 3. 2 Flowchart Proses Pembuatan Itinerary

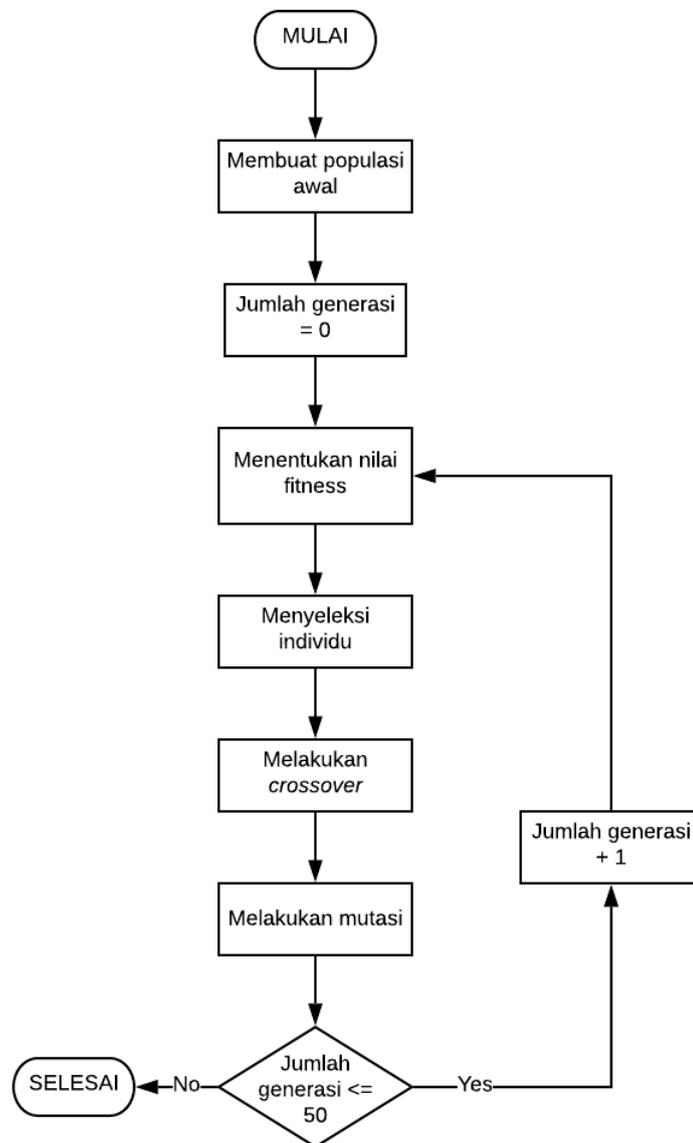
Dapat dilihat pada Gambar 3. 2, proses dimulai dari mengklaster destinasi wisata dengan mengimplementasikan *constrained k-means*, jumlah hari digunakan sebagai jumlah klaster. Setelah destinasi wisata diklasterkan, proses berikutnya adalah membuat *thread* dimana jumlah *thread* sama dengan jumlah hari. Masing-masing *thread* akan menjalankan Algoritma Genetika untuk mencari solusi TSP pada setiap klaster.

Flowchart selanjutnya menggambarkan proses klustering menggunakan *constrained k-means*. Proses yang dilakukan adalah mengelompokkan destinasi wisata ke dalam klaster k . *Constraint* dalam algoritma ini adalah jumlah minimum anggota pada setiap klaster. Penggunaan *constraint* pada *k-means* bertujuan untuk menyeimbangkan anggota klaster. Proses dimulai dengan menentukan jumlah k klaster dimana k adalah jumlah hari, lalu menentukan *demand* yang merupakan jumlah minimum anggota klaster, menentukan centroid awal, kemudian membuat node data point dan node centroid. Semua node data point terhubung dengan semua node centroid yang memiliki nilai *cost* dimana *cost* merupakan jarak *Euclidean* antara data point dengan semua centroid. Anggota klaster ditetapkan berdasarkan *cost* dari node data point ke node centroid. Karena proses klustering ini menggunakan metode *minimum cost flow*, maka kebutuhan *supply* dan *demand* harus seimbang dimana node dengan *demand* tidak diperbolehkan menerima *supply* lebih dari kebutuhan *demand*. Node tambahan diperlukan untuk menyeimbangkan kebutuhan *supply* dan *demand*. Proses dari klustering ini dapat dilihat pada Gambar 3. 3.



Gambar 3. 3 Flowchart Proses Klustering

Flowchart terakhir menggambarkan proses mencari solusi TSP menggunakan Algoritma Genetika. Proses tersebut dapat dilihat pada Gambar 3. 4.



Gambar 3. 4 Flowchart Algoritma Genetikaa

Dalam mencari solusi TSP, langkah awal yang dilakukan adalah membuat populasi awal. Populasi awal akan menghasilkan kumpulan kromosom yang kemudian akan diseleksi untuk dilakukan proses *crossover* dan mutasi. Proses *crossover* akan menghasilkan populasi baru yang dinamakan *children*. Kemudian dengan probabilitas yang kecil, akan dilakukan proses mutasi pada *children*. Proses ini akan diulang sejumlah 50 generasi dimana setiap generasi akan menghasilkan *children* sebagai populasi baru untuk generasi berikutnya.

Contoh Kasus

Untuk memperjelas bagaimana cara kerja sistem itinerary wisata berjalan, maka disajikan contoh kasus sebagai berikut:

Jumlah hari = 2.

Lokasi awal = Hotel Ambarukmo.

Destinasi wisata = Candi Prambanan, Kraton Yogyakarta, Malioboro Yogyakarta, Pasar Beringharjo Yogyakarta, Boko, Gembira Loka Zoo.

Langkah awal penyelesaiannya adalah mengklaster destinasi wisata sejumlah k klaster dimana k adalah jumlah hari = 2. Kemudian destinasi wisata yang sudah diklasterkan akan dicari solusi TSP-nya menggunakan Algoritma Genetika. Untuk contoh kasus, ditentukan jumlah populasi dan jumlah generasi pada Algoritma Genetika yaitu masing-masing 5.

Keterangan :

A : Hotel Ambarukmo (lokasi awal)

B : Malioboro Yogyakarta

C : Gembira Loka Zoo

D : Candi Prambanan

E : Pasar Beringharjo Yogyakarta

F : Kraton Yogyakarta

Koordinat destinasi wisata untuk proses klaster :

Tabel 3. 1 Koordinat Destinasi Wisata

Kota	Latitude	Longitude
B	-7.7926306	110.3658442
C	-7.8041252	110.3980215
D	-7.7520206	110.4914674
E	-7.7987265	110.3653401
F	-7.8052845	110.3642031

Proses Klastering

Langkah proses klastering menggunakan *constrained k-means* :

- a. Diketahui jumlah klaster adalah 2.
- b. Menentukan *demand* (jumlah minimum anggota klaster)

Demand harus berupa bilangan *integer*, ditentukan dengan cara menghitung jumlah destinasi dibagi jumlah hari. Pada kasus ini jumlah destinasi = 5 dan jumlah hari = 2.

Maka, dengan menggunakan rumus (2.2) sehingga $demand = 2$.

- c. Membuat node data dengan $supply = -1$

Supply artinya node akan memberikan nilai *supply*.

- d. Membuat node centroid.

Node centroid masing-masing akan memiliki nilai $demand = 2$.

- e. Menentukan centroid awal.

Centroid 1 dipilih dari koordinat C dan Centroid 2 dipilih dari koordinat D

Tabel 3.3 Centroid awal

Centroid	Latitude	longitude
Centroid 1	-7.8041252	110.3980215
Centroid 2	-7.8052845	110.3642031

- f. Menghitung *cost* dimana *cost* adalah jarak *Euclidean* semua data terhadap masing-masing centroid. Perhitungan *cost* akan disajikan dalam bentuk matriks sebagai berikut:

Tabel 3. 2 Nilai *Cost* Semua Data ke Setiap Centroid

Lokasi	Cost	
	Ke Centroid 1	Ke Centroid 2
B	0.03416876	0.01275987
C	0	0.03383826
D	0.10699077	0.13796103
E	0.03312431	0.00665583
F	0.03383826	0

- g. Membuat node tambahan untuk menyemibangkan kebutuhan *supply* dan *demand*
Node tambahan dibuat dengan nilai $cost = 0$ dan memiliki nilai $demand =$ jumlah destinasi wisata – sum *demand* pada centroid . Node tambahan diberikan label Z

- h. Menetapkan anggota klaster dengan minimum cost flow

Berdasarkan tabel 3.4 didapati minimum cost flow-nya yaitu

B : {Centroid 1 : 0, Centroid 2 : 1}

C : {Centroid 1 : 1, Centroid 2 : 0}

D : {Centroid 1 : 0, Centroid 2 : 1}

E : {Centroid 1 : 1, Centroid 2 : 0}

F : {Centroid 1 : 1, Centroid 2 : 0}

Centroid 1 : { Z : 1}

Centroid 2 : { Z : 0}

Maksud dari *flow* di atas adalah node B, C, D, E, dan F masing-masing memberikan *supply* ke Centroid 1 dan Centroid 2 berdasarkan nilai *cost*-nya. Dapat dilihat pada Centroid 1 mendapatkan *supply* yang berlebih yaitu 3, masing-masing 1 dari node C, E, dan F. Oleh karena itu Centroid 1 memberikan nilai 1 ke node Z sehingga kebutuhan *supply* dan *demand* menjadi seimbang. Berdasarkan minimum *cost flow* didapati anggota klaster pada setiap centroid yaitu :

Centroid 1 : {C, E, F}

Centroid 2 : {B, D}

i. Memperbarui centroid

Centroid diperbarui dengan cara menghitung rata-rata dari semua data point yang tergabung dalam setiap klaster.

Perhitungan centroid baru dapat dilihat pada Tabel 3. 3 dan Tabel 3. 4.

Centroid 1 :

Tabel 3. 3 Perhitungan Centroid 1 Baru

Lokasi	Latitude	Longitude
C	-7.8052845	110.3642031
E	-7.8041252	110.3980215
F	-7.7926306	110.3658442
Mean	-7.8006801	110.37602293

Centroid 2 :

Tabel 3. 4 Perhitungan Centroid 2 Baru

Lokasi	Latitude	Longitude
B	-7.7520206	110.4914674

D	-7.7705416	110.4894158
Mean	-7.7612811	110.4904416

Centroid baru dapat dilihat pada Tabel 3. 5.

Tabel 3. 5 Koordinat Centroid Baru

Centroid	Latitude	longitude
Centroid 1	-7.8006801	110.37602293
Centroid 2	-7.7612811	110.4904416

Iterasi ke-1

Centroid iterasi 1 dapat dilihat pada Tabel 3. 6.

Tabel 3. 6 Centroid Iterasi 1

Centroid	Latitude	Longitude
Centroid 1	-7.8006801	110.37602293
Centroid 2	-7.7612811	110.4904416

Cost iterasi 1 dapat dilihat pada Tabel 3. 7.

Tabel 3. 7 Cost Iterasi 1

Lokasi	Cost	
	Ke Centroid 1	Ke Centroid 2
B	0.12528037	0.00931714
C	0.01268499	0.13368791
D	0.11732975	0.00931714
E	0.02226669	0.10186801
F	0.01297694	0.12848075

Minimum Cost Flow :

B : {Centroid 1 : 0, Centroid 2 : 1}

C : {Centroid 1 : 1, Centroid 2 : 0}

D : {Centroid 1 : 0, Centroid 2 : 1}

E : {Centroid 1 : 1, Centroid 2 : 0}

F : {Centroid 1 : 1, Centroid 2 : 0}

Centroid 1 : { Z : 1}

Centroid 2 : { Z : 0}

Karena tidak ada perubahan pada anggota kluster maka iterasi berhenti. Didapati hasil kluster sebagai berikut :

Kluster 1 : {C, E, F}

Kluster 2 : {B, D}

Proses TSP

Setelah mendapatkan destinasi pada masing-masing kluster, tahap berikutnya adalah mencari solusi TSP setiap rute dari setiap kluster. Langkah pertama pada Algoritma Genetika adalah membuat populasi yang berisi kumpulan kromosom atau individu. Kromosom tersebut berupa rute yang terdiri dari kumpulan destinasi wisata. Model kromosom dapat dilihat pada Gambar 3. 5 dan Gambar 3. 6.

Hotel Ambarukmo	Kraton Yogyakarta	Malioboro Yogyakarta	Gembira Loka Zoo
-----------------	-------------------	-------------------------	------------------

Gambar 3. 5 Representasi Kromosom Kluster 1

Hotel Ambarukmo	Candi Prambanan	Istana Ratu Boko
-----------------	-----------------	------------------

Gambar 3. 6 Representasi Kromosom Kluster 2

Kromosom dapat direpresentasikan sebagai rute perjalanan wisata. Populasi dibuat dengan cara membuat kromosom sebesar jumlah populasi yaitu 5. Masing-masing kromosom akan memiliki urutan destinasi wisata secara acak. Untuk menghitung jarak antar lokasi diperlukan matriks jarak dalam satuan meter yang dapat dilihat pada Tabel 3. 8.

Tabel 3. 8 Matriks Jarak

Index i, j	A	B	C	D	E	F
A	0	11561	7511	11958	4461	5622
B	12813	0	18334	5560	14328	16804

C	6320	18145	0	18541	4322	2674
D	13004	5520	18525	0	14519	16995
E	3947	14258	4441	14654	0	5524
F	5064	16583	1506	16980	4518	0

Proses penyelesaian TSP disajikan dalam bentuk tabel sebagai berikut :

Solusi TSP klaster 1 :

Jarak dihitung dalam satuan Meter

Populasi awal klaster 1 dapat dilihat pada Tabel 3. 9.

Tabel 3. 9 Populasi Awal Klaster 1

Kromosom	Jarak	Fitness
Kromosom 1 (A, C, E, F)	15279	6,54493E-05
Kromosom 2 (A, C, F, E)	13350	7,49064E-05
Kromosom 3 (A, C, E, F)	15279	6,54493E-05
Kromosom 4 (A, E, F, C)	11139	8,97747E-05
Kromosom 5 (A, E, F, C)	11139	8,97747E-05

Di setiap populasi akan dilakukan proses seleksi dengan metode *roulette wheel selection*. Dari 5 kromosom akan diseleksi hanya 2 kromosom sebagai parent 1 dan parent 2. Akan dilakukan proses *crossover* menggunakan metode *ordered crossover* untuk kromosom yang telah diseleksi. Metode *ordered crossover* dilakukan dengan cara mengambil subset dari parent 1 secara acak, lalu mengisi sisa kromosom dengan gen (destinasi wisata) yang ada pada parent 2 dimana gen tersebut tidak ada dalam subset yang diambil dari parent 1. Proses *crossover* akan menghasilkan *children* sebagai populasi baru. Setelah proses *crossover*, dengan probabilitas rendah yaitu 0.01 *children* akan dilakukan proses mutasi dengan metode *swap mutation*. *Swap mutation* dilakukan dengan cara menukar gen di dalam kromosom. *Children*

akan menjadi populasi baru untuk generasi berikutnya. Jarak dan fitness setiap kromosom pada setiap generasi dapat dilihat pada Tabel 3. 10, Tabel 3. 11, Tabel 3. 12, Tabel 3. 13, dan Tabel 3. 14 untuk klaster 1 dan pada Tabel 3. 16, Tabel 3. 17, Tabel 3. 18, Tabel 3. 19, dan Tabel 3. 20 untuk klaster 2. Tabel 3. 10 Generasi 1 Klaster 1

Generasi 1 :

Tabel 3. 10 Generasi 1 Klaster 1

Kromosom	Jarak	Fitness
Kromosom 1 (A, E, F, C)	11139	8,97747E-05
Kromosom 2 (A, E, F, C)	11139	8,97747E-05
Kromosom 3 (A, E, F, C)	11139	8,97747E-05
Kromosom 4 (A, C, E, F)	15279	6,54493E-05
Kromosom 5 (A, C, E, F)	15279	6,54493E-05

Generasi 2 :

Tabel 3. 11 Generasi 2 Klaster 1

Kromosom	Jarak	Fitness
Kromosom 1 (A, E, F, C)	11139	8,97747E-05
Kromosom 2 (A, E, F, C)	11139	8,97747E-05
Kromosom 3 (A, C, E, F)	15279	6,54493E-05
Kromosom 4 (A, E, F, C)	11139	8,97747E-05
Kromosom 5 (A, C, E, F)	13350	7,49064E-05

(A, C, F, E)		
--------------	--	--

Generasi 3 :

Tabel 3. 12 Generasi 3 Klaster 1

Kromosom	Jarak	Fitness
Kromosom 1 (A, E, F, C)	11139	8,97747E-05
Kromosom 2 (A, E, F, C)	11139	8,97747E-05
Kromosom 3 (A, E, C, F)	9775	0,000102302
Kromosom 4 (A, F, C, E)	12179	8,21085E-05
Kromosom 5 (A, C, E, F)	15279	6,54493E-05

Generasi 4 :

Tabel 3. 13 Generasi 4 Klaster 1

Kromosom	Jarak	Fitness
Kromosom 1 (A, E, C, F)	9775	0,0001023
Kromosom 2 (A, E, F, C)	11139	8,9775E-05
Kromosom 3 (A, F, E, C)	14910	6,7069E-05
Kromosom 4 (A, E, C, F)	9775	0,0001023
Kromosom 5 (A, E, F, C)	11139	8,9775E-05

Generasi 5 :

Tabel 3. 14 Generasi 5 Klaster 1

Kromosom	Jarak	Fitness
Kromosom 1 (A, E, C, F)	9775	0,000102302
Kromosom 2 (A, E, C, F)	9775	0,000102302
Kromosom 3 (A, F, C, E)	12179	8,21085E-05
Kromosom 4 (A, C, F, E)	13350	7,49064E-05
Kromosom 5 (A, F, E, C)	14910	6,70691E-05

Dilihat dari generasi ke – 5, kromosom yang paling fit adalah kromosom 1 dan 2 dengan total jarak 9,775 KM. Karena rutenya sama maka didapati rute terbaik adalah Hotel Ambarukmo, Gembira Loka Zoo, Kraton Yogyakarta, dan Malioboro Yogyakarta.

Solusi TSP pada klaster 2 :

Populasi awal klaster 2 dapat dilihat pada Tabel 3. 15.

Tabel 3. 15 Populasi Awal Klaster 2

Kromosom	Jarak	Fitness
Kromosom 1 (A, B, D)	18333	5,45464E-05
Kromosom 2 (A, B, D)	18333	5,45464E-05
Kromosom 3 (A, B, D)	18333	5,45464E-05
Kromosom 4 (A, B, D)	18333	5,45464E-05
Kromosom 5 (A, B, D)	18333	5,45464E-05

(A, B, D)		
-----------	--	--

Generasi 1 :

Tabel 3. 16 Generasi 1 Klaster 2

Kromosom	Jarak	Fitness
Kromosom 1 (A, B, D)	18333	5,45464E-05
Kromosom 2 (A, B, D)	18333	5,45464E-05
Kromosom 3 (A, D, B)	18564	5,38677E-05
Kromosom 4 (A, B, D)	18333	5,45464E-05
Kromosom 5 (A, D, B)	18564	5,38677E-05

Generasi 2 :

Tabel 3. 17 Generasi 2 Klaster 2

Kromosom	Jarak	Fitness
Kromosom 1 (A, B, D)	18333	5,45464E-05
Kromosom 2 (A, B, D)	18333	5,45464E-05
Kromosom 3 (A, D, B)	18564	5,38677E-05
Kromosom 4 (A, D, B)	18564	5,38677E-05

Kromosom 5 (A, B, D)	18333	5,45464E-05
-------------------------	-------	-------------

Generasi 3 :

Tabel 3. 18 Generasi 3 Klaster 2

Kromosom	Jarak	Fitness
Kromosom 1 (A, B, D)	18333	5,45464E-05
Kromosom 2 (A, B, D)	18333	5,45464E-05
Kromosom 3 (A, D, B)	18564	5,38677E-05
Kromosom 4 (A, D, B)	18564	5,38677E-05
Kromosom 5 (A, D, B)	18564	5,38677E-05

Generasi 4 :

Tabel 3. 19 Generasi 4 Klaster 2

Kromosom	Jarak	Fitness
Kromosom 1 (A, B, D)	18333	5,4546E-05
Kromosom 2 (A, B, D)	18333	5,4546E-05
Kromosom 3 (A, B, D)	18333	5,4546E-05
Kromosom 4 (A, B, D)	18333	5,4546E-05
Kromosom 5 (A, D, B)	18564	5,3868E-05

Generasi 5 :

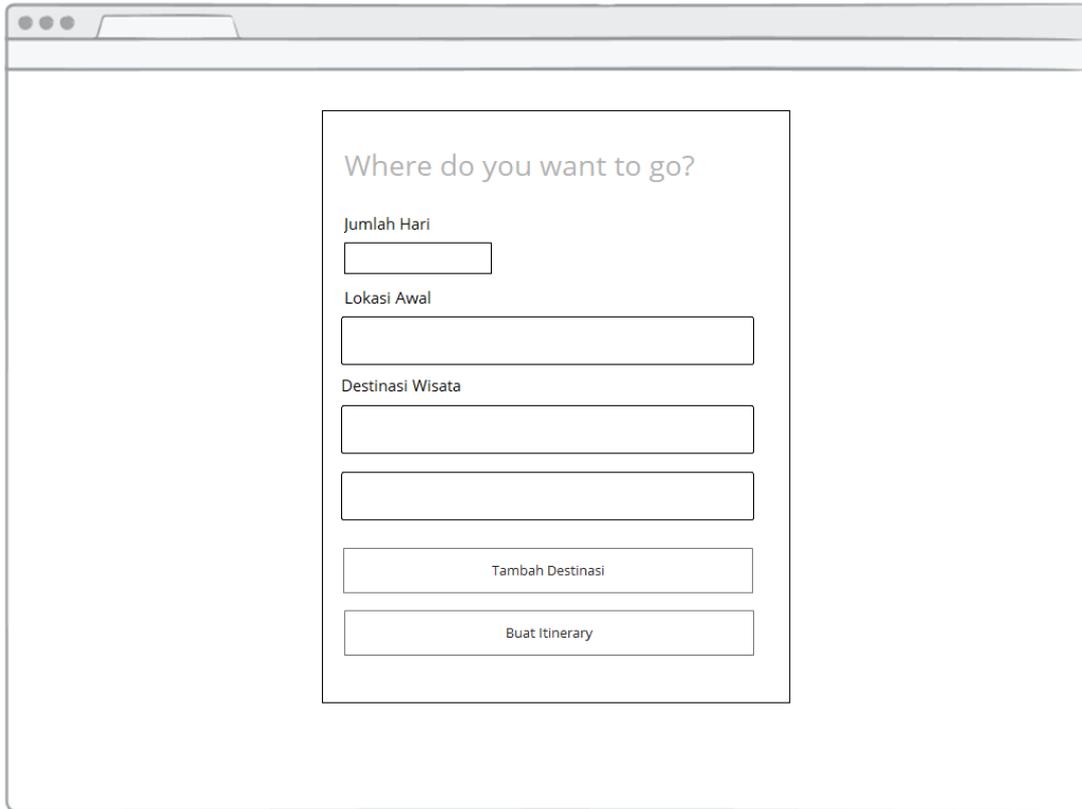
Tabel 3. 20 Generasi 5 Klaster 2

Kromosom	Jarak	Fitness
Kromosom 1 (A, B, D)	18333	5,45464E-05
Kromosom 2 (A, B, D)	18333	5,45464E-05
Kromosom 3 (A, B, D)	18333	5,45464E-05
Kromosom 4 (A, B, D)	18333	5,45464E-05
Kromosom 5 (A, B, D)	18333	5,45464E-05

Dilihat dari generasi ke – 5, semua kromosom memiliki nilai fitness yang sama dengan jarak 18,33 KM. Maka dapat disimpulkan bahwa rute terbaik adalah Hotel Ambarukmo, Candi Prambanan, Istana Ratu Boko.

3.5 Perancangan Antar Muka

Rancangan antar muka pada aplikasi itinerary wisata terdiri dari 2 halaman. Halaman pertama untuk memasukan jumlah hari, lokasi awal, dan destinasi wisata. Halaman kedua untuk menampilkan hasil itinerary wisata. Rancangan antar muka dari aplikasi itinerary wisata dapat dilihat pada Gambar 3. 7 dan Gambar 3. 8.



Where do you want to go?

Jumlah Hari

Lokasi Awal

Destinasi Wisata

Gambar 3. 7 Halaman Pertama



Hari ke - 1

MAP

Rute Hari ke - 1

Lokasi	Total Jarak Perjalanan
Lokasi	Estimasi Waktu Perjalanan
Lokasi	

Gambar 3. 8 Halaman Kedua

Pada Gambar 3. 7, pengguna dapat mengisikan field jumlah hari, lokasi awal perjalanan, dan destinasi wisata yang ingin dikunjungi. Kemudian, pada Gambar 3. 8, pengguna dapat melihat hasil itinerary wisata. Halaman tersebut terdiri dari canvas map untuk memudahkan pengguna dalam melihat petunjuk arah setiap lokasi. Kemudian terdapat urutan destinasi serta total jarak perjalanan dan estimasi waktu perjalanan pada masing-masing hari.

3.6 Skenario Pengujian

Terdapat 3 skenario pengujian yang akan dilakukan terhadap sistem yaitu pengujian *black box*, pengujian performa sistem, dan *usability testing*. Pengujian performa sistem dilakukan dengan cara menganalisis waktu pemrosesan sistem. *Usability sistem* dilakukan dengan cara memberikan kuisioner kepada calon pengguna.

3.6.1 Rancangan Black Box Testing

Pengujian Black Box adalah pengujian dimana struktur / desain / implementasi sistem yang sedang diuji tidak diketahui oleh *tester*. Pengujiannya dapat berupa fungsional dan non-fungsional. Untuk sistem itinerary wisata dilakukan pengujian fungsional yaitu dengan cara mengamati hasil eksekusi sistem. Skenario pengujian dapat dilihat pada Tabel 3. 21.

Tabel 3. 21 Skenario Pengujian *Black Box*

Kelas uji	Butir Uji
Jumlah hari	Menentukan jumlah hari
Lokasi awal	Memasukan lokasi awal
Destinasi wisata	Memasukan destinasi wisata
	Menambah destinasi wisata
Proses klastering	Menentukan anggota klaster
Proses membuat <i>thread</i>	Membuat <i>thread</i>
Proses mencari solusi TSP	Membuat rute perjalanan

3.6.2 Rancangan Pengujian Performa

Dalam pengujian performa dilakukan analisis waktu pemrosesan sistem. Pengujian ini akan dilakukan dengan 3 perangkat komputer yang berbeda. Terdapat 2 jenis skenario pengujian yaitu pengujian dengan *multithreading* dan tanpa *multithreading*. Beberapa faktor

yang mempengaruhi waktu pemrosesan antara lain kecepatan prosesor, kapasitas RAM, koneksi internet, jumlah destinasi wisata dan jumlah hari. Rancangan dari pengujian ini dapat dilihat pada Tabel 3. 22.

Tabel 3. 22 Rancangan Pengujian Performa

No	Jumlah hari	Jumlah lokasi	Rata-rata waktu pemrosesan (tanpa <i>multithreading</i>) (<i>second</i>)	Rata-rata waktu pemrosesan (dengan <i>multithreading</i>) (<i>second</i>)
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

3.6.3 Rancangan Usability Testing

Pengujian ini dilakukan dengan cara meminta responden untuk mengisi kuisioner. Pengujian ini bertujuan untuk mengevaluasi sistem apakah semua fitur di dalamnya dapat berjalan dengan baik dan sesuai dengan tujuan dari aplikasi.

Tabel 3.24 Rancangan *Usability Testing*

No	Pertanyaan	Jawaban				
		STS	TS	N	S	SS
1	Apakah aplikasi dapat memberikan itinerary wisata?					
2	Apakah aplikasi dapat menampilkan lokasi wisata dalam map?					
3	Apakah aplikasi dapat memberikan petunjuk arah setiap lokasi?					

4	Apakah aplikasi dapat menampilkan rute perjalanan wisata?					
5	Apakah aplikasi dapat menampilkan informasi detail setiap lokasi?					