

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi Sistem

Aplikasi Itinerary Wisata yang dibangun berbasis website ini akan dibangun dengan pendekatan *Traveling Salesman Problem*, *Constrained k-means*, dan *Multithreading*. Sistem akan menerima masukan dari pengguna berupa jumlah hari, lokasi awal, dan destinasi wisata. Masukan tersebut akan diproses oleh sistem sehingga sistem dapat memberikan rekomendasi rute perjalanan wisata. Akan dilakukan proses klustering menggunakan metode *Constrained K-Means* pada masukan destinasi wisata dengan jumlah kluster adalah jumlah hari. Setiap destinasi yang telah dikluster akan dicari solusi TSP-nya menggunakan Algoritma Genetika dengan jumlah populasi dan jumlah generasi = 50. Solusi TSP tersebut menjadi rekomendasi rute perjalanan wisata. Masukan lokasi awal akan menjadi *starting point* untuk masing-masing rute perjalanan wisata. Proses mencari solusi TSP akan diimplementasikan teknik *Multithreading* dimana jumlah *thread* sama dengan jumlah hari.

#### 4.2 Batasan Implementasi

Dalam pengembangan aplikasi Itinerary Wisata, ditetapkan batasan-batasan implementasi berupa asumsi-asumsi sebagai berikut :

- a. Menggunakan Google API untuk mendapatkan data berupa jarak, durasi, dan detail lokasi.
- b. Menggunakan metode *constrained k-means* untuk proses klustering.
- c. Menggunakan Algoritma Genetika untuk solusi TSP.
- d. Sistem akan dibangun dengan berbasis website.
- e. Pengguna telah menentukan jumlah hari berwisata.
- f. Pengguna telah menentukan lokasi awal.
- g. Pengguna telah menentukan destinasi wisata yang ingin diikunjungi.
- h. Lokasi awal dan destinasi wisata berada di lingkup Daerah Istimewa Yogyakarta.
- i. Jumlah jumlah hari harus kurang dari jumlah destinasi wisata.
- j. Maksimal jumlah destinasi wisata adalah 20.
- k. Estimasi waktu perjalanan merupakan waktu perjalanan mengunjungi setiap destinasi wisata tanpa berhenti.

### 4.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak dalam pengembangan aplikasi Itinerary Wisata adalah sebagai berikut :

a. Web Browser

*Web browser* digunakan untuk mengakses halaman web aplikasi. Halaman tersebut terdiri dari halaman masukan dan halaman hasil. Halaman masukan diakses pengguna untuk mendapatkan data masukan dan halaman hasil akan menampilkan rute perjalanan wisata.

b. Django

Django merupakan *framework* yang menggunakan bahasa python untuk mengembangkan aplikasi berbasis web. Django digunakan untuk membantu dalam mengembangkan aplikasi Itinerary Wisata.

c. Python 3.7

Python digunakan sebagai bahasa pemrograman untuk mengembangkan aplikasi Itinerary Wisata.

d. Text Editor

Text digunakan untuk menulis kode.

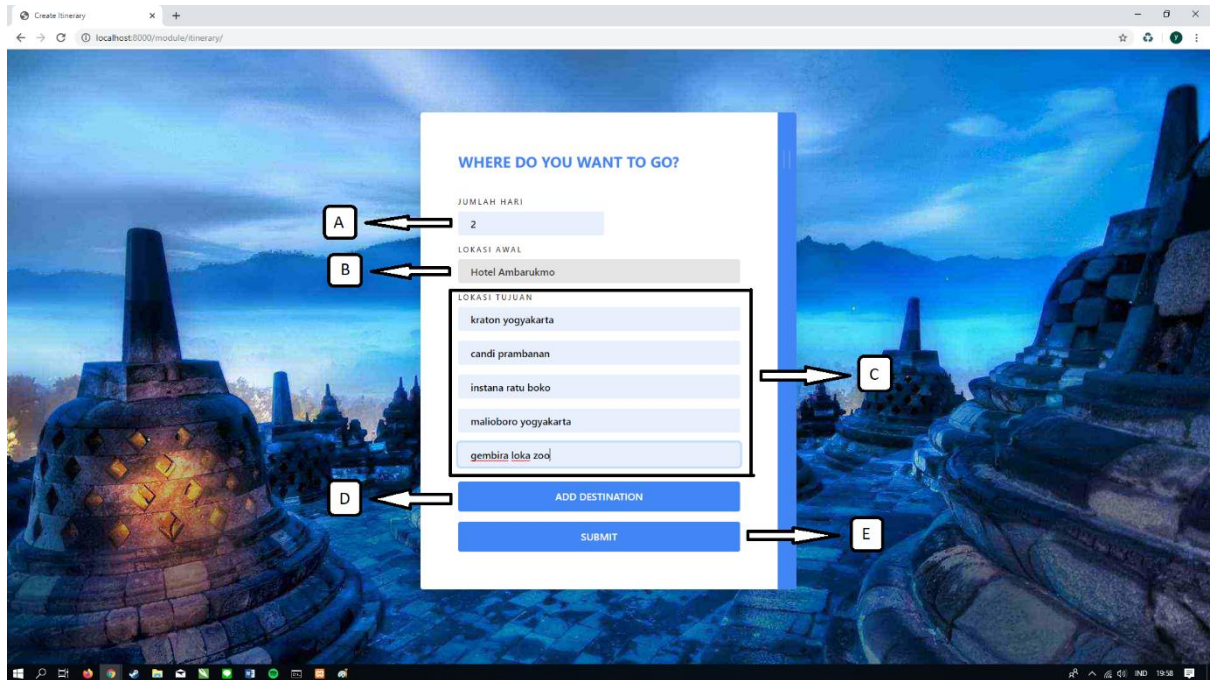
e. Justinmind

Justinmind digunakan untuk membuat rancangan antar muka aplikasi.

### 4.4 Hasil Implementasi

#### 4.4.1 Halaman Masukan

Pada halaman masukan terdapat beberapa fitur yaitu field untuk memasukan jumlah hari, field untuk memasukan lokasi awal, field untuk memasukan destinasi wisata, tombol menambah destinasi, dan tombol submit untuk membuat *itinerary*. Semua *field* tidak boleh kosong. Hasil dari halaman ini dapat dilihat pada Gambar 4. 1 Halaman Masukan Gambar 4. 1.



Gambar 4. 1 Halaman Masukan

Fungsi dari masing-masing fitur antara lain :

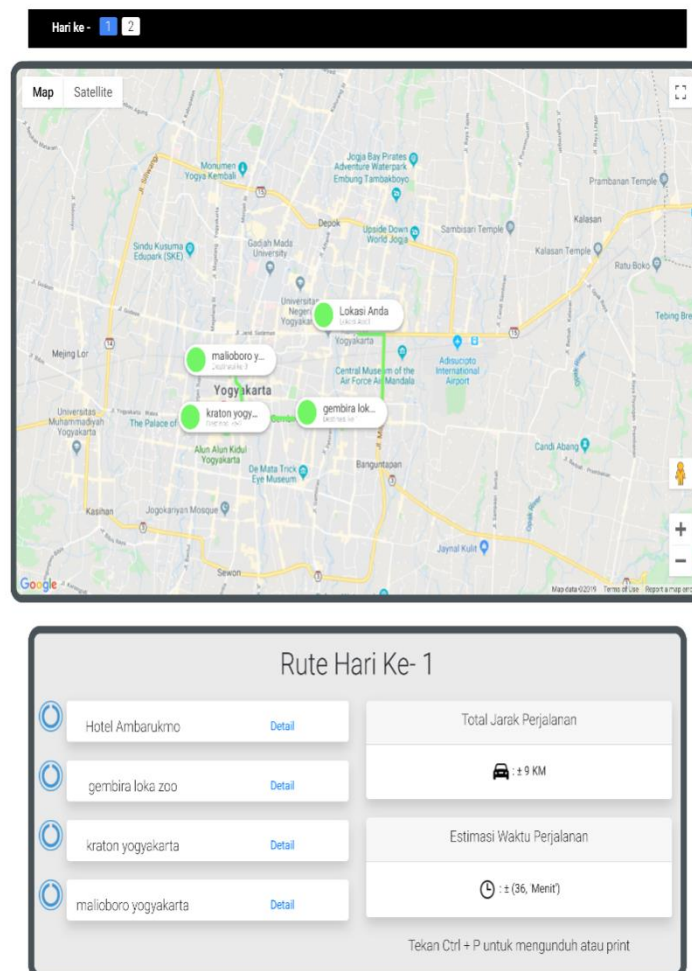
- a. Field jumlah hari  
 Pada field jumlah hari, pengguna dapat memasukkan jumlah hari yang diinginkan untuk berwisata. Masukkan hanya dapat berupa angka. Tampilan fitur ini dapat dilihat pada Gambar 4. 1 yang diinisialkan dengan huruf “A”.
- b. Field Lokasi awal  
 Pada field ini pengguna dapat memasukkan lokasi awal sebagai *starting point* rute perjalanan wisata. Lokasi awal diasumsikan berada pada lingkup Daerah Istimewa Yogyakarta. Field ini menggunakan layanan *autocomplete* Google API untuk mempermudah pengguna dalam memasukan alamat dari lokasi awal. Tampilan fitur ini dapat dilihat pada Gambar 4. 1 yang diinisialkan dengan huruf “B”.
- c. Field Destinasi Wisata  
 Pada field ini pengguna dapat memasukkan alamat destinasi wisata yang ingin dikunjungi dengan asumsi destinasi wisata berada pada lingkup Daerah Istimewa Yogyakarta. Tampilan fitur ini dapat dilihat pada Gambar 4. 1 yang diinisialkan dengan huruf “C”.
- d. Tombol “ADD DESTINATION”  
 Tombol ini berfungsi untuk menambah *field* destinasi wisata yang ingin dikunjungi. Tampilan fitur ini dapat dilihat pada Gambar 4. 1 yang diinisialkan dengan huruf “D”.

e. Tombol “SUBMIT”

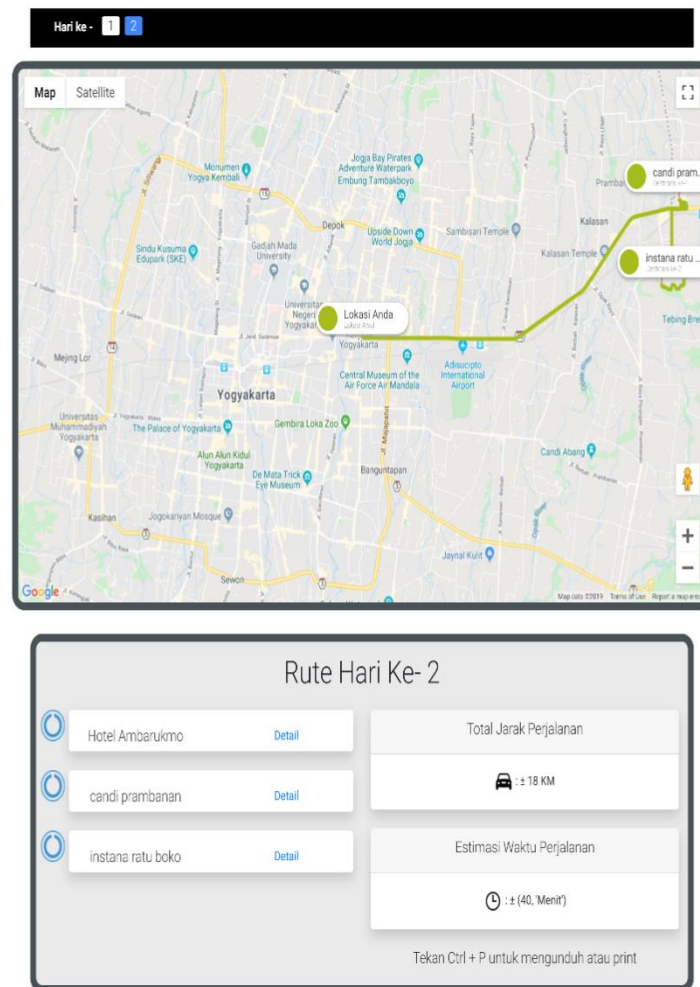
Tombol ini berfungsi untuk mengeksekusi semua data masukan dari pengguna yang kemudian data tersebut akan diproses sehingga sistem dapat memberikan hasil berupa rute perjalanan wisata. Tampilan fitur ini dapat dilihat pada Gambar 4. 1 yang diinisialkan dengan huruf “E”.

#### 4.4.2 Halaman Hasil

Halaman hasil menampilkan hasil dari proses pembuatan itinerary berdasarkan masukan dari halaman masukan. Halaman ini memberikan beberapa informasi yaitu lokasi setiap destinasi beserta petunjuk arah perjalanan yang ditampilkan pada *map canvas*, rute perjalanan wisata, total jarak, dan estimasi waktu pada setiap harinya. Gambar 4. 2 dan Gambar 4. 3 menunjukkan hasil dari halaman hasil.



Gambar 4. 2 Halaman Hasil Hari Ke-1



Gambar 4. 3 Halaman Hasil Hari Ke-2

Terdapat 3 bagian yang terdapat pada halaman hasil. Bagian pertama adalah *header*, bagian kedua adalah *canvas map*, dan bagian ketiga adalah bagian informasi rute. Berikut penjelasan dari setiap bagian pada halaman hasil :

f. Header

*Header* terletak pada bagian paling atas halaman. Pada bagian ini pengguna dapat memilih rekomendasi itinerary pada hari yang telah ditentukan pengguna.

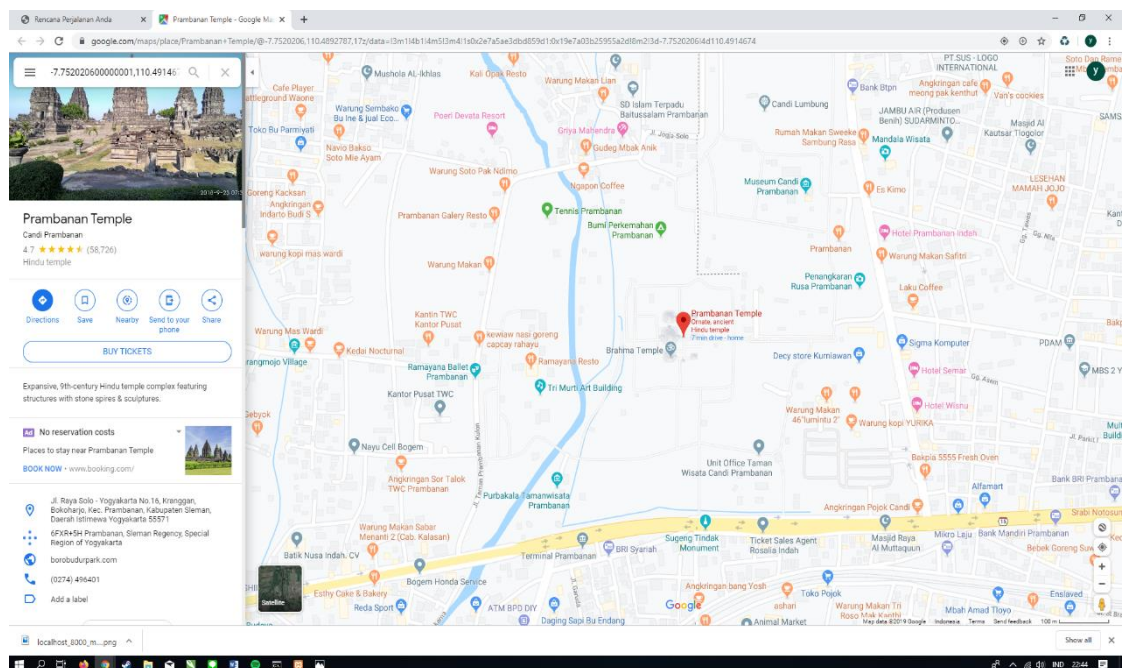
g. Map Canvas

*Map Canvas* memberikan informasi kepada pengguna berupa lokasi awal dan semua lokasi destinasi wisata yang ada pada hari yang telah ditentukan pengguna. Tidak hanya itu, bagian ini juga memberikan petunjuk arah perjalanan ke masing-masing destinasi wisata. Gambar 4.2 menampilkan *canvas map* pada hari ke -1 dengan petunjuk arah dari lokasi awal yaitu Hotel Ambarukmo, lalu menuju ke Gembira Loka Zoo sebagai destinasi

ke-1, lalu menuju ke Kraton Yogyakarta sebagai destinasi ke -2, dan yang terakhir menuju ke Malioboro Yogyakarta sebagai destinasi ke -3. Sedangkan gambar 4.3 menampilkan *canvas map* pada hari ke -2 dengan petunjuk arah dari Hotel Ambarukmo, menuju ke Candi Prambanan sebagai destinasi ke -1, kemudian yang terakhir menuju ke Istana ratu Boko sebagai destinasi ke -2.

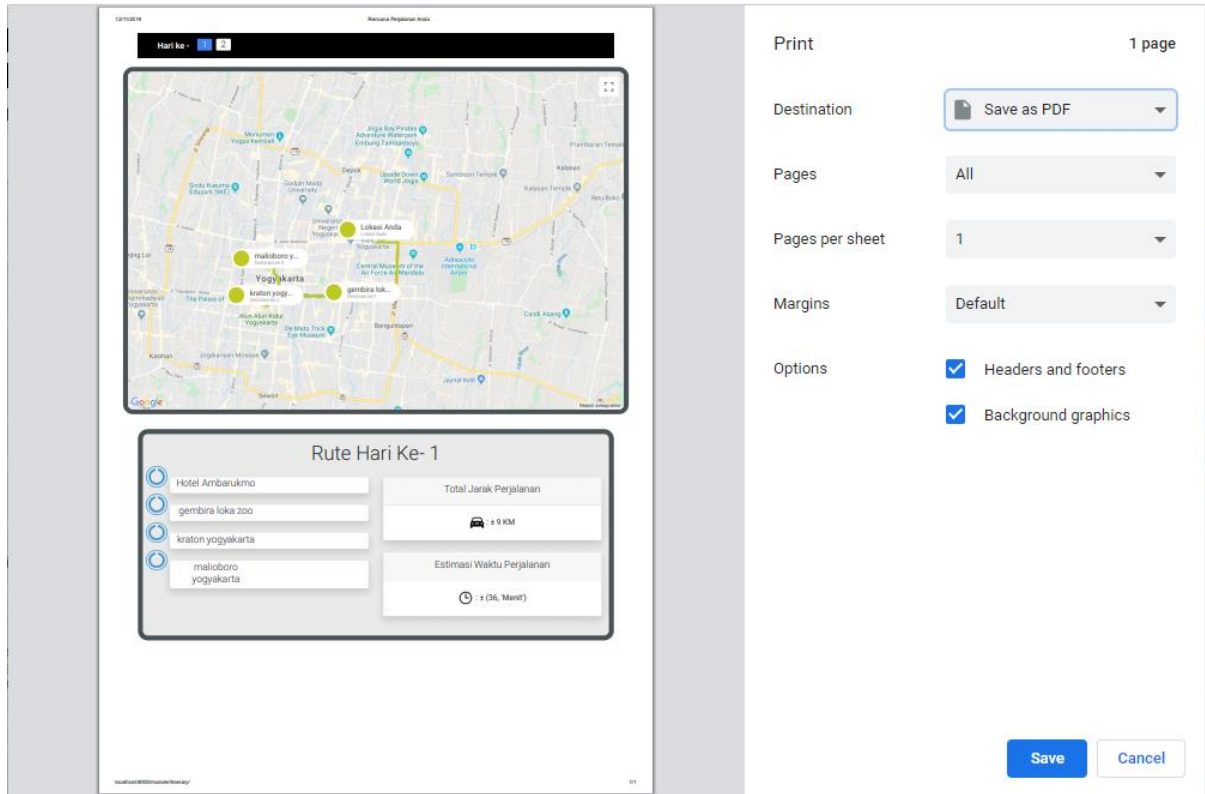
#### h. Informasi Rute

Bagian Informasi rute berada di paling bawah halaman hasil yaitu di bawah *Map Canvas*. Bagian ini memberikan informasi kepada pengguna berupa urutan rute perjalanan wisata, total jarak perjalanan, dan estimasi waktu perjalanan pada hari yang telah ditentukan pengguna. Gambar 4. 2 menampilkan urutan rute perjalanan pada hari ke -1 yaitu Hotel Ambarukmo, Gembira Loka Zoo, Kraton Yogyakarta, dan Malioboro Yogyakarta dengan total jarak kurang lebih 9 KM dan estimasi waktu perjalanan 36 menit. Sedangkan Gambar 4. 3 menampilkan rute perjalanan pada hari ke -2 yaitu Hotel Ambarukmo, Candi Prambanan, dan Istana Ratu Boko dengan total jarak kurang lebih 18 KM dan estimasi waktu perjalanan 40 menit. Pengguna dapat melihat informasi detail masing-masing lokasi dengan menekan tombol “Detail”. Tombol tersebut akan mengarahkan pengguna menuju halaman informasi detail lokasi yang disediakan oleh google. Halaman tersebut dapat dilihat pada Gambar 4. 4

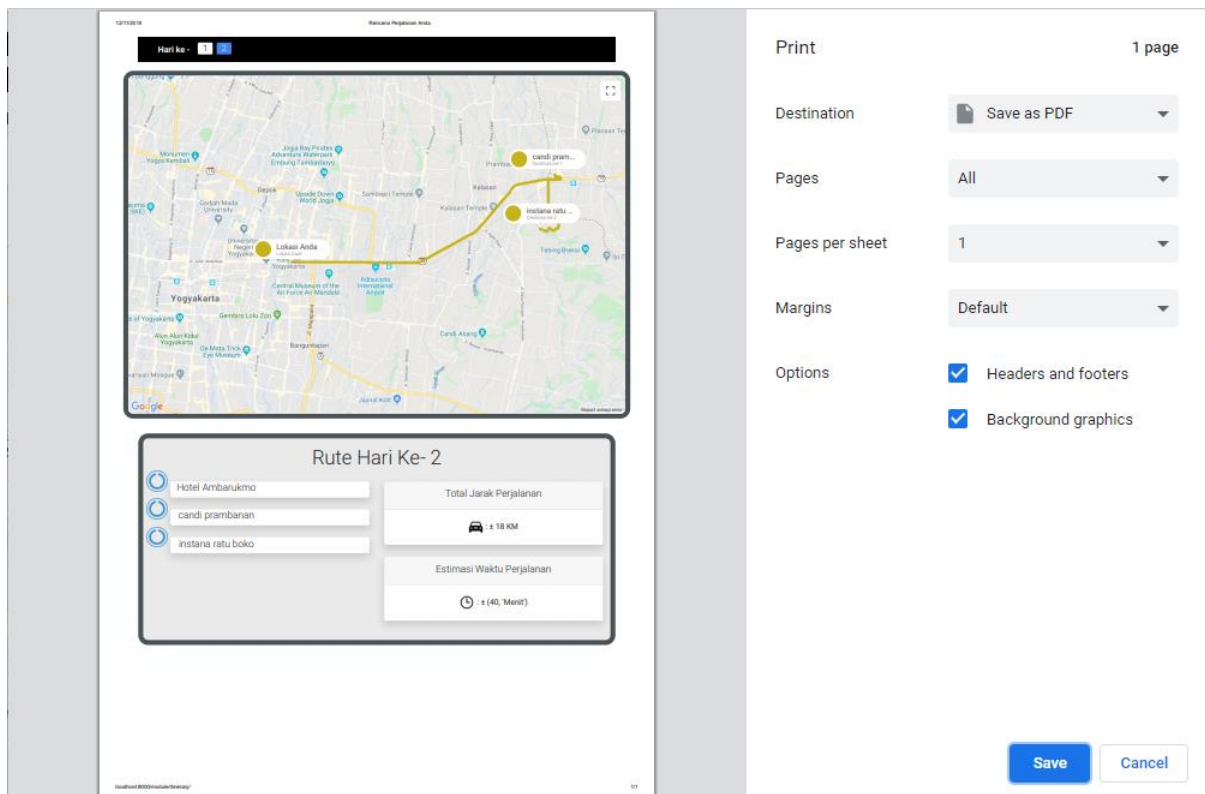


Gambar 4. 4 Halaman Google Untuk Detail Lokasi





Gambar 4. 5 Unduh *Itinerary* Hari ke-1



Gambar 4. 6 Unduh *Itinerary* Hari ke-2

Rekomendasi itinerary wisata dapat dicetak dan diunduh dengan menekan tombol CTRL + P. Pengguna dapat mengunduh atau mencetak itinerary wisata pada masing-masing hari. Gambar 4. 5 dan Gambar 4. 6 menunjukkan hasil cetak atau unduh rekomendasi itinerary.

## 4.5 Hasil Pengujian

### 4.5.1 Hasil Black Box Testing

Hasil pengujian *black box* terhadap fungsionalitas aplikasi Itinerary Wisata dapat dilihat pada Tabel 4.1 berikut :

Tabel 4. 1 Hasil *Black Box Testing*

Kelas Uji	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
Pengujian fitur jumlah hari.	Sistem dapat menerima dan menyimpan masukkan.	True	<ul style="list-style-type: none"> <li>Masukkan berupa angka.</li> <li>Data tersimpan dalam <i>session</i>.</li> </ul>
Pengujian fitur lokasi awal.	Sistem dapat memberikan <i>autocomplete</i> pada field lokasi awal.	True	
	Sistem dapat menerima dan menyimpan masukkan.	True	<ul style="list-style-type: none"> <li>Masukkan berupa alamat lokasi.</li> <li>Data tersimpan dalam <i>session</i>.</li> </ul>
Pengujian fitur destinasi wisata.	Sistem dapat menerima dan menyimpan masukkan.	True	<ul style="list-style-type: none"> <li>Masukkan berupa alamat lokasi.</li> <li>Data tersimpan dalam <i>session</i>.</li> </ul>
	Sistem dapat menambah field destinasi wisata.	True	
Pengujian fitur tombol submit	Sistem dapat memberikan rekomendasi itinerary wisata	True	



#### 4.5.2 Hasil Pengujian Performa

Pengujian performa yang dilakukan adalah dengan menganalisis waktu pemrosesan sistem dalam mengeksekusi program. Terdapat 2 jenis pengujian yang dilakukan, pertama melakukan percobaan dengan menjalankan aplikasi tanpa teknik *multithreading* dan kedua menjalankan program dengan *multithreading*. Teknik *multithreading* diimplementasikan dengan cara membuat *thread* dimana jumlah *thread* sama dengan jumlah hari. Setiap percobaan diulangi sebanyak tiga kali lalu menghitung rata-rata waktu pemrosesan. Walaupun data masukannya sama, waktu pemrosesan sistem mungkin akan bervariasi. Hal ini bergantung pada beberapa faktor yaitu koneksi internet, memory pada komputer, dan kecepatan prosesor. Pengujian ini juga dilakukan pada tiga perangkat komputer yang berbeda. Spesifikasi masing-masing komputer sebagai berikut:

a. Komputer 1 :

Prosesor : intel core i5-5200 2,3 GHz

RAM : 2 GB

b. Komputer 2 :

Prosesor : Intel Core i5-2310 2,9 GHz

RAM : 8 GB

c. Komputer 3 :

Prosesor : Intel Core i5-6200U 2,4 GHz

RAM : 8 GB

Hasil dari analisis tersebut adalah sebagai berikut :

Pengujian pada komputer 1 :

Tabel 4. 2 Hasil Pengujian Performa Komputer 1

No	Jumlah hari	Jumlah destinasi	Rata-rata waktu pemrosesan (tanpa <i>multithreading</i> )	Rata-rata waktu pemrosesan (dengan <i>multithreading</i> )
1	3	6	3,33250	2,47143
2	3	8	3,64223	2,66786
3	3	10	4,11850	2,82763
4	3	15	4,62256	3,45716
5	3	20	6,82060	3,73590

6	4	10	5,00375	3,43716
7	4	15	5,25993	3,87843
8	4	20	5,97670	4,20263
9	5	15	6,12256	4,38730
10	5	20	7,33790	4,84593

Pengujian pada komputer 2 :

Tabel 4. 3 Hasil Pengujian Performa Komputer 2

No	Jumlah hari	Jumlah destinasi	Rata-rata waktu pemrosesan (tanpa <i>multithreading</i> ) ( <i>second</i> )	Rata-rata waktu pemrosesan (dengan <i>multithreading</i> ) ( <i>second</i> )
1	3	6	3,53610	2,73660
2	3	8	3,68483	2,88796
3	3	10	3,93946	2,97303
4	3	15	5,18196	3,47096
5	3	20	5,95610	4,00426
6	4	10	5,01906	3,59646
7	4	15	5,76740	4,15273
8	4	20	6,54130	4,56730
9	5	15	6,76160	4,77233
10	5	20	7,39270	5,31116

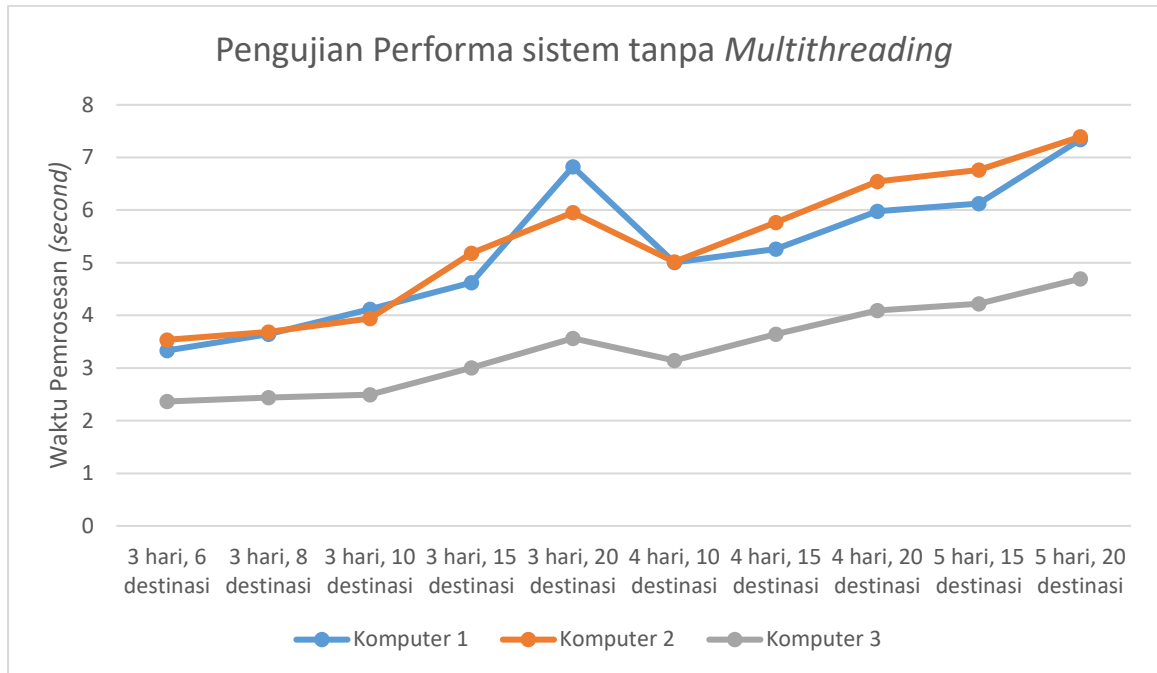
Pengujian pada komputer 3 :

Tabel 4. 4 Hasil Pengujian Performa Komputer 3

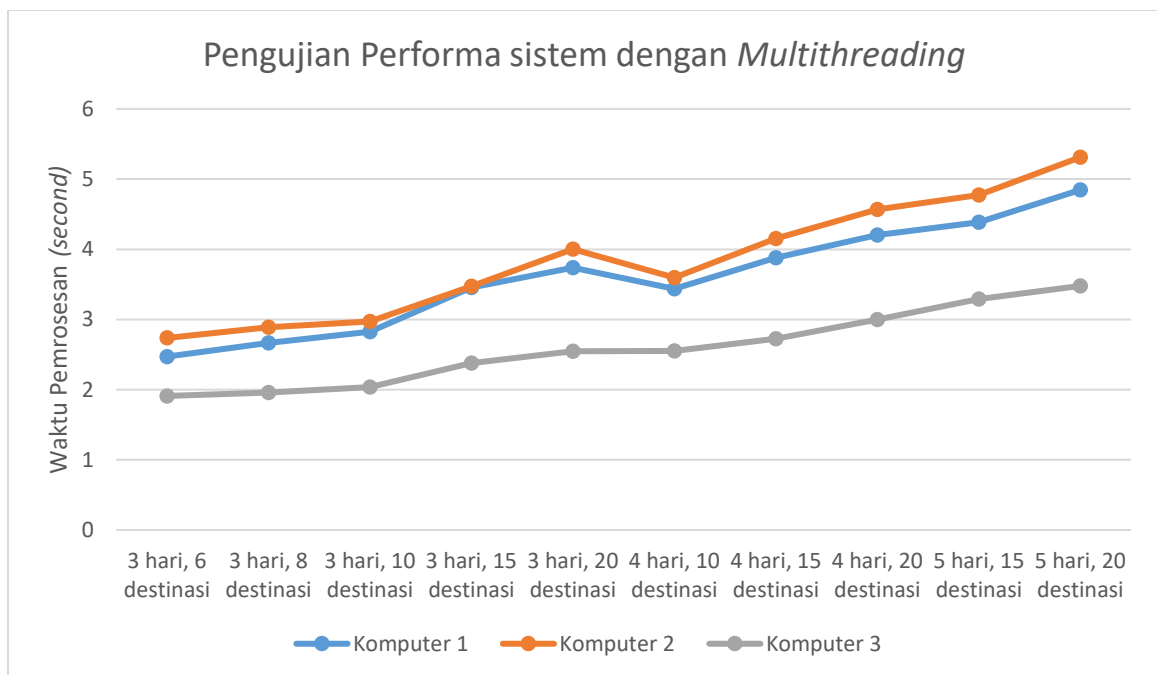
No	Jumlah hari	Jumlah destinasi	Rata-rata waktu pemrosesan (tanpa <i>multithreading</i> ) ( <i>second</i> )	Rata-rata waktu pemrosesan (dengan <i>multithreading</i> ) ( <i>second</i> )
1	3	6	2,36360	1,91003
2	3	8	2,43576	1,95806
3	3	10	2,49366	2,03816

4	3	15	3,00376	2,37710
5	3	20	3,56290	2,54513
6	4	10	3,14523	2,55276
7	4	15	3,64320	2,72696
8	4	20	4,09033	2,99660
9	5	15	4,22296	3,29083
10	5	20	4,69573	3,47830

Dapat dilihat pada Tabel 4. 2, Tabel 4. 3, dan Tabel 4. 4, waktu pemrosesannya bervariasi. Hal ini dipengaruhi oleh jumlah hari, jumlah destinasi, spesifikasi komputer, dan koneksi internet. Seperti yang diketahui pada kasus TSP bahwa semakin banyak jumlah kota maka waktu pemrosesannya akan bertambah lama. Tabel 4.5 menunjukkan perbandingan antara penelitian sebelumnya dengan metode *Brute Force* (Rani, Kholidah, & Huda, 2018). Dapat dilihat pada Tabel 4.5, untuk jumlah destinasi yang sedikit, metode *Brute Force* lebih cepat waktu pemrosesannya. Namun, untuk jumlah destinasi yang banyak, Algoritma Genetika lebih cepat daripada metode *Brute Force*. Hal ini membuktikan bahwa Algoritma Genetika lebih efisien dalam menyelesaikan TSP dengan jumlah kota yang banyak dari segi waktu pemrosesan. Dari hasil yang didapat, implementasi *multithreading* juga dapat mengurangi waktu pemrosesan. Sedangkan, pengujian aplikasi pada perangkat yang berbeda memberikan hasil yang bervariasi. Gambar 4. 7 dan Gambar 4. 8 menunjukkan grafik garis dari performa ketiga komputer.



Gambar 4. 7 Grafik Pengujian Performa Pada 3 Komputer Tanpa *Multithreading*



Gambar 4. 8 Grafik Pengujian Performa Pada 3 Komputer Dengan *Multithreading*

Tabel 4. 5 Rata-Rata Waktu Pemrosesan 3 Komputer

No	Rata-rata waktu pemrosesan dari 3 komputer (tanpa <i>multithreading</i> ) ( <i>second</i> )	Rata-rata waktu pemrosesan dari 3 komputer (dengan <i>multithreading</i> ) ( <i>second</i> )
1	3,0774	2,37269
2	3,25427	2,50463
3	3,51721	2,61294
4	4,26943	3,10174
5	5,44653	3,42843
6	4,38935	3,19546
7	4,89018	3,58604
8	5,53611	3,92218
9	5,70237	4,15015
10	6,47544	4,54513

Tabel 4. 6 Perbandingan Waktu Pemrosesan dengan Penelitian Sebelumnya

No	Jumlah hari	Jumlah destinasi	Rata-rata waktu pemrosesan ( <i>second</i> )		
			<i>Brute force</i>	Algoritma Genetika tanpa <i>Multithreading</i>	Algoritma Genetika dengan <i>Multithreading</i>
1	3	6	2.0817	3,0774	2,37269
3	3	8	3.5128	3,25427	2,50463
5	3	10	4.6360	3,51721	2,61294
10	3	15	5.7449	4,26943	3,10174

#### 4.5.3 Usability Testing

Pengujian ini dilakukan dengan cara meminta responden untuk menjalankan aplikasi, kemudian responden diminta untuk mengisi kuisioner. Dalam pengujian ini, responden merupakan calon wisatawan. Berikut merupakan daftar responden :

Tabel 4. 7 Daftar Responden

Nama
Anfika Sigma
Aditya Mahavira
Dian Ramadhan
Nassharieh Abdulloh
Firza Halim

Berikut merupakan hasil dari usability testing :

Tabel 4. 8 Hasil *Usability Testing*

No	Pertanyaan	Jawaban					Total
		STS	TS	N	S	SS	
1	Apakah aplikasi dapat memberikan itinerary wisata?			3	2		17
2	Apakah aplikasi dapat menampilkan lokasi wisata dalam map?				1	4	24
3	Apakah aplikasi dapat memberikan petunjuk arah setiap lokasi?				4	1	21
4	Apakah aplikasi dapat menampilkan rute perjalanan wisata?				3	2	22
5	Apakah aplikasi dapat menampilkan informasi detail setiap lokasi?			1	3	1	20

Dengan menggunakan formula ( 2.3 ) didapati index keberhasilan dari *Usability Testing* di atas yaitu 83,2%. Dapat disimpulkan aplikasi termasuk pada kategori baik.



