

Aplikasi Perencanaan Wisata di Malang Raya dengan Algoritma *Greedy*

Akhmad Eriq Ghozali¹, Budi Darma Setiawan², M. Tanzil Furqon³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹ericghozali@gmail.com, ²s.budidarma@ub.ac.id, ³m.tanzil.furqon@gmail.com

Abstrak

Malang Raya merupakan salah satu wilayah yang menjadi tempat tujuan utama karena memiliki banyak tempat wisata. Dalam berwisata biasanya hal yang perlu diperhatikan adalah menentukan jadwal pariwisata, setiap orang yang melakukan perjalanan pariwisata pasti memilih jarak dan waktu terpendek untuk dapat mencapai tujuan karena dapat menghemat waktu. Untuk mewujudkan tujuan tersebut digunakan algoritma *greedy* dengan knapsack problem untuk membantu proses optimasi terhadap pencarian waktu tempuh terpendek dan banyaknya jumlah wisata yang dapat dikunjungi dari waktu yang dimiliki. Alokasi waktu yang dimiliki pengguna untuk berwisata digunakan sebagai bobot dalam perhitungan aplikasi ini, sedangkan waktu tempuh pada setiap titik lokasi wisata yang digunakan juga sebagai bobot adalah data waktu yang didapatkan dari google maps. Dengan data tersebut, aplikasi dengan algoritma *greedy* akan menghitung lokasi yang paling optimal untuk dikunjungi dengan waktu yang dimiliki pengguna. Berdasarkan hasil pengujian aplikasi dengan 10 contoh kasus permasalahan mendapatkan hasil akurasi sebesar 90% pada kedua model perhitungan algoritma *greedy* dalam mencari jumlah lokasi yang dapat dikunjungi dengan alokasi waktu yang dimiliki. Sedangkan hasil akurasi wisata optimal yang dikunjungi yaitu 0% pada model perhitungan pertama dan 80% pada model perhitungan kedua.

Kata kunci : *algoritma greedy, malang raya, perencanaan wisata.*

Abstract

Malang raya is one of regions which becomes the main objective place to visit because it has many tourism places. The thing which has to be noticed is determining the tourism schedule, every tourist must choose the shortest distance and time to be able to reach that destination because they can save the time. To reach that destination, it is used greedy algorithm with knapsack problem to assist the optimization process against searching the shortest traveling time and how many tourism places which can be visited from the possessed time. Time allocation which is possessed by the user to tour is used as an integrity in calculating this application, while the traveling time at each tourism locations which are also used as an integrity is time data which is gotten from google maps. With thats data, the application with greedy algorithm will calculate the most optimal location to be visited with the time which belongs to the user. According to the result of testing application with ten sample of problem cases gets accuracy result 90% from two models of greedy algorithm calculation in searching location which can be visited by the allocation time which is owned. While the result of optimal tour accuracy that is visited is 0% from the first model of calculation and 80% from the second calculation.

Keywords : *greedy algorithm, malang raya, tour planning.*

1. PENDAHULUAN

Malang Raya adalah wilayah metropolitan yang merupakan gabungan dari tiga wilayah seperti Kota Batu, Kota Malang, dan Kabupaten Malang, Provinsi Jawa Timur. Wilayah Malang Raya dikenal sebagai salah satu tujuan wisata utama di Indonesia (Malang, 2015). Wisata yang ada di Malang Raya beraneka ragam, mulai dari

wisata kota, wisata bahari, wisata kuliner dan wisata-wisata lainnya. Hal ini menyebabkan banyak wisatawan yang tertarik untuk mengunjungi Malang Raya.

Dalam berwisata biasanya hal yang perlu diperhatikan adalah menentukan jadwal pariwisata dan mengatur waktu sebaik mungkin. Termasuk dalam melakukan perjalanan pariwisata, setiap orang juga pasti memilih waktu tempuh terpendek untuk dapat mencapai

tujuan karena dapat menghemat waktu, tenaga dan biaya (Priatmoko, 2014).

Salah satu algoritma yang dapat mencari waktu tempuh terpendek adalah algoritma *Greedy*. Algoritma *Greedy* adalah algoritma yang dapat memecahkan masalah langkah demi langkah dan merupakan salah satu metode dalam masalah optimasi (Kurnia, 2006).

Pada penelitian terkait sebelumnya yang menggunakan metode *greedy* menyimpulkan bahwa pencarian lintasan terpendek sangat berguna untuk menentukan jalan tersingkat untuk menuju suatu tempat (Hayati, E.N. & Yohanes, 2014).

Knapsack problem merupakan masalah di mana orang dihadapkan pada persoalan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam sebuah wadah yang memiliki keterbatasan ruang atau daya tampung. Dengan adanya optimasi dalam pemilihan benda yang akan dimasukkan ke dalam wadah tersebut diharapkan dapat menghasilkan keuntungan yang maksimum. Wadah yang dimaksud di sini juga memiliki nilai konstanta yang merupakan nilai pembatas untuk benda-benda yang akan dimasukkan ke dalam wadah tersebut sehingga harus diambil sebuah cara memasukkan benda-benda tersebut ke dalam wadah sehingga menghasilkan hasil optimum tetapi tidak melebihi kemampuan wadah untuk menampungnya (Kartina, 2010).

Dalam penelitian ini Knapsack Problem digunakan untuk mengoptimasi jumlah tempat wisata yang dapat dikunjungi dari waktu yang dimiliki oleh wisatawan. Benda atau beban dalam Knapsack diasumsikan dengan waktu, sedangkan wadah maksimum diasumsikan dengan total waktu yang dimiliki oleh wisatawan. Algoritma *Greedy* digunakan untuk mendukung dalam optimasi dan perhitungan waktu yang paling singkat dari waktu perjalanan dan waktu kunjung. Dari permasalahan diatas maka penulis mengangkat judul Aplikasi Perencanaan Wisata di Malang Raya dengan Algoritma *Greedy*.

2. LANDASAN KEPUSTAKAAN

2.1. Wisata Malang Raya

Malang Raya adalah wilayah metropolitan yang merupakan gabungan dari tiga wilayah seperti Kota Batu, Kota Malang, dan Kabupaten Malang Provinsi Jawa Timur. Kawasan Malang Raya merupakan metropolitan terbesar kedua di

Jawa Timur. Wilayah Malang Raya dikenal sebagai salah satu tujuan wisata utama di Indonesia (Malang, 2015).

Dengan potensi alam yang dimiliki oleh Kota Malang, yaitu pemandangan alam yang elok serta hawa yang sejuk, teduh dan asri serta bangunan-bangunan kuno peninggalan Belanda, Kota Malang menjadi tujuan wisata bagi wisatawan dalam maupun luar negeri. Berbagai pilihan tempat perbelanjaan, baik yang bersifat tradisional maupun modern yang tersebar di berbagai penjuru kota sangat menunjang Kota Malang sebagai Kota Pariwisata. Banyak sekali lokasi wisata yang terdapat di kawasan Malang Raya. Berikut adalah sebagian beberapa wisata menarik yang populer di kawasan Malang Raya ini, baik wisata alam maupun buatan diantaranya adalah:

- A. Jatim Park 1
- B. Batu Sreet Zoo
- C. Ecogreenpark
- D. Alun-Alun Batu
- E. Paralayang
- F. Coban Rondo
- G. Selecta
- H. Sengkaling
- I. Hawaii Water Park
- J. Alun-Alun Malang

2.2. Optimasi

Optimasi bisa berupa maksimasi atau minimasi. Jika berkaitan dengan masalah keuntungan, maka keadaan optimum adalah keadaan yang memberikan keuntungan maksimum (maksimasi). Jika berkaitan dengan masalah pengeluaran/pengorbanan, maka keadaan optimum adalah keadaan yang memberikan pengeluaran/pengorbanan minimum (minimasi) (Kholisoh, 2008).

Dalam penelitian ini, optimasi digunakan untuk mendapatkan rekomendasi rute wisata yang dikunjungi. Untuk mendapatkan hasil yang optimal, maka pertimbangan yang digunakan adalah sebagai berikut :

1. Mengutamakan jumlah objek wisata yang dapat dikunjungi dengan waktu yang ditentukan.
2. Waktu, banyak dihabiskan pada lokasi wisata dari pada waktu perjalanan.

Dengan pertimbangan diatas, maka tujuan utama dalam penelitian menggunakan algoritma *greedy* adalah optimasi jumlah wisata yang dapat dikunjungi dengan waktu yang dimiliki oleh pengguna. Kemudian, mencari rasio optimal dari waktu dengan cara membandingkan

total waktu perjalanan dan total waktu kunjung pada lokasi.

2.3. Algoritma Greedy

Algoritma *Greedy* merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Hanya ada dua macam persoalan optimasi, yaitu maksimasi dan minimasi. Algoritma *greedy* membentuk solusi langkah per langkah. Pada setiap langkah, terdapat banyak pilihan yang perlu dieksplorasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Pada setiap langkahnya merupakan pilihan, untuk membuat langkah optimum local dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global. Sesuai arti harafiah, *Greedy* berarti tamak. Prinsip utama dari algoritma ini adalah mengambil sebanyak mungkin apa yang dapat diperoleh sekarang (Munir, 2005)

2.4 Greedy dalam Knapsack Problem

Salah satu penggunaan metode *greedy* adalah untuk menyelesaikan permasalahan knapsack (*knapsack problem*). Pada penyelesaian *Knapsack Problem* terdapat 3 jenis algoritma *Greedy* yang dapat digunakan yaitu (Rachmawati, 2013):

A. Greedy By Weight

Pada setiap langkah pilih objek yang mempunyai berat teringan. Mencoba memaksimalkan keuntungan dengan memasukkan sebanyak mungkin objek ke dalam knapsack.

Pertama kali yang dilakukan adalah program mengurutkan secara menaik objek-objek berdasarkan weightnya. Kemudian baru diambil satu-persatu objek yang dapat ditampung oleh knapsack sampai knapsack penuh atau sudah tidak ada objek lagi yang bisa dimasukkan.

B. Greedy by profit

Pada setiap langkah, pilih objek yang mempunyai keuntungan terbesar. Mencoba memaksimalkan keuntungan dengan memilih objek yang paling menguntungkan terlebih dahulu.

Pertama kali yang dilakukan adalah program mengurutkan secara menurun objek-objek berdasarkan profitnya. Kemudian baru diambil satu-persatu objek yang dapat ditampung oleh knapsack sampai knapsack penuh atau sudah tidak ada objek lagi yang bisa dimasukkan.

C. Greedy By Density

Pada setiap langkah knapsack di isi dengan objek yang mempunyai pi/wi terbesar, dimana p adalah keuntungan dan w adalah berat barang. Mencoba memaksimalkan keuntungan dengan memilih objek yang mempunyai density per unit berat terbesar.

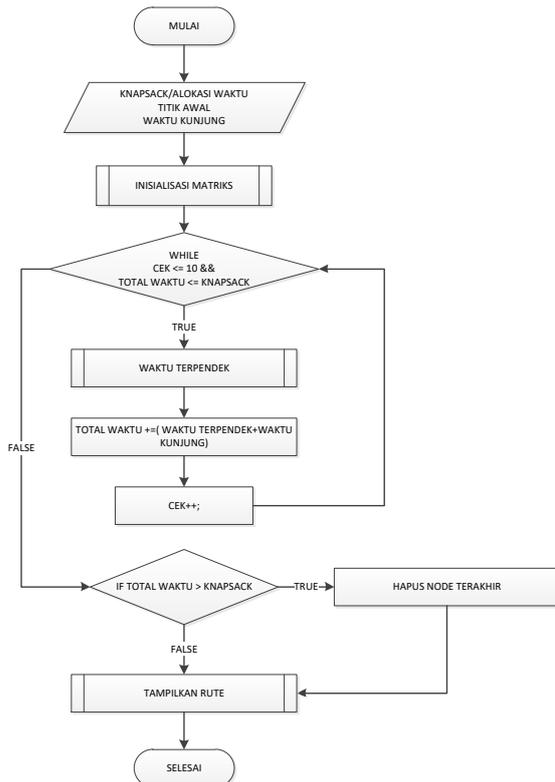
Pertama kali yang dilakukan adalah program mencari nilai profit per berat tiap – tiap unit (density) dari tiap- tiap objek. Kemudian objek-objek tersebut diurutkan berdasarkan density-nya. Kemudian baru diambil satu-persatu objek yang dapat ditampung oleh knapsack sampai knapsack penuh atau sudah tidak ada objek lagi yang bisa dimasukkan (Rachmawati, 2013).

3. METODOLOGI DAN PERANCANGAN

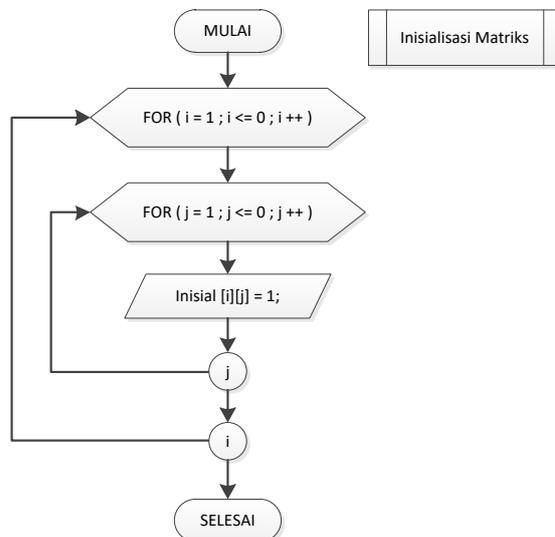
3.1. Perancangan Sistem

Dalam Tahap alur aplikasi perencanaan wisata di Malang Raya dengan algoritma *greedy* langkah-langkahnya adalah sebagai berikut :

Gambar 1 akan menjelaskan bagaimana alur aplikasi menampilkan rute yang paling optimal. Pada flowchart tersebut total alokasi waktu yang digunakan untuk berwisata diasumsikan sebagai knapsack kemudian perhitungan dimulai dari titik awal yang ditentukan oleh pengguna. Inisialisasi matriks digunakan untuk mengidentifikasi apakah nilai yang dimiliki dari titik tersebut dihitung atau tidak. Setelah itu program berjalan dengan perulangan *while* total waktu < knapsack. Kemudian program dilanjutkan dengan menginisialisasi matrik yang ditunjukkan pada Gambar 2 dan perhitungan waktu terpendek yang akan ditunjukkan pada Gambar 3. Perhitungan berjalan sampai habis, jika total waktu lebih dari knapsack maka program akan menghapus rute terakhir dan jika tidak lebih dari knapsack maka program akan menampilkan rute optimal.



Gambar 1 Flowchart Aplikasi Rute Optimal

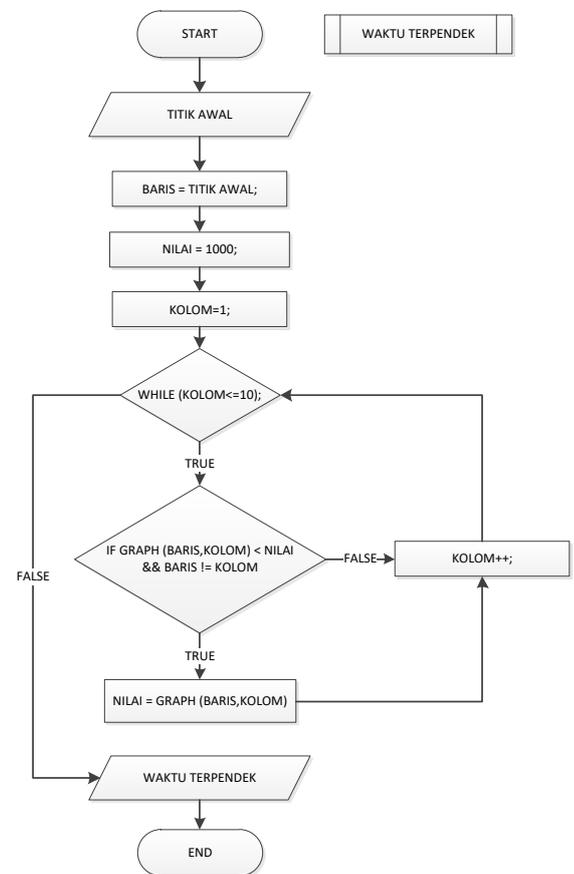


Gambar 2 Flowchart Inisialisasi Matriks

Dari proses Gambar 2 membentuk nilai matriks yang ditunjukkan pada Tabel 1. Nilai matriks ini digunakan untuk menginisialisasi apakah nilai yang ada pada matriks tersebut dihitung dalam mencari optimasi rute atau tidak. Pada awal perhitungan, semua nilai akan diinisialisasikan dengan nilai 1. Apabila nilai tersebut sudah dihitung maka program akan merubahnya menjadi 0 dan tidak akan dihitung lagi dalam perhitungan selanjutnya.

Tabel 1 Nilai Inisialisasi Matriks

Matriks (i, j)	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1



Gambar 3 Flowchart Perhitungan Waktu Terpendek

Dari Gambar 3 maka dapat dilihat bagaimana alur proses perhitungan waktu terpendek. Data yang telah dimasukkan oleh pengguna yaitu titik awal digunakan untuk memulai baris pada matrik yang akan dihitung terlebih dahulu. Nilai awal diasumsikan 1000 agar pada perhitungan awal bisa berjalan. Kemudian dicari nilai terkecil dari kolom pada baris matrik dengan perulangan tersebut. Hasil dari perhitungan tersebut akan menjadi rute selanjutnya. Kemudian data akan dihitung pada

sistem sampai alokasi waktu (knapsack) tidak mencukupi. sekarang

3.2. Perhitungan Manual

Dari data waktu tempuh yang didapatkan dari *google maps* maka berikut adalah tabel data yang didapatkan dan contoh perhitungan manual dari perencanaan wisata di Malang Raya dengan algoritma *greedy*.

Tabel 2 Data waktu tempuh antar lokasi di Malang Raya

X	A	B	C	D	E	F	G	H	I	J
A	X	3	4	6	23	30	18	17	36	40
B	3	X	1	7	25	32	20	18	36	41
C	4	1	X	8	26	33	21	11	37	42
D	6	6	7	X	19	28	13	18	36	40
E	18	20	21	16	X	17	25	33	48	56
F	29	31	32	27	17	X	35	43	58	66
G	19	19	20	13	25	36	X	30	42	52
H	19	18	19	18	38	45	32	X	23	25
I	39	39	40	38	57	64	45	26	X	18
J	41	46	47	45	64	70	58	30	27	X

Sumber : *maps.google.com*

Keterangan :

A : JATIM PARK 1

B : BATU SCREET ZOO

C : ECOGREENPARK

D : ALUN-ALUN BATU

E : PARALAYANG

F : COBAN RONDO

G : SELECTA

H : SENGKALING

I : HAWAI WATER PARK

J : ALUN-ALUN MALANG

Dari data tersebut, dimisalkan total alokasi yang dimiliki adalah 6 jam (360 menit) dan perjalanan dimulai dari titik Jatim Park 1 (A). Lokasi dan alokasi waktu kunjung masing-masing tempat wisata yang akan dikunjungi adalah sebagai berikut :

a. Batu Sreet Zoo (B) : 200 menit

b. Alun-alun Batu (D) : 60 menit

c. Coban Rondo (F) : 120 menit

d. Selecta (G) : 120 menit

3.4.1 Model Perhitungan Pertama

Model perhitungan pertama ini menjumlah waktu tempuh dan waktu kunjung wisata terlebih dahulu, kemudian dicari nilai yang terkecil. Apabila wisata tersebut tidak dikunjungi maka tidak termasuk dalam perhitungan. Titik yang terpilih sebagai rute selanjutnya menjadi titik awal untuk perhitungan berikutnya. Maka perhitungan dari kasus diatas adalah sebagai berikut.

- $A - B = 3 + 200 = 203$ menit

- $A - D = 6 + 60 = 66$ menit

- $A - F = 30 + 120 = 150$ menit

- $A - G = 18 + 120 = 128$ menit

Dari perhitungan tersebut maka $A - D$ adalah nilai terkecil dan D titik yang dipilih. Maka sisa waktu yang dimiliki menjadi $360 - 66 = 294$ menit. Kemudian perhitungan dilanjutkan dari titik D.

- $D - B = 6 + 200 = 206$ menit

- $D - F = 28 + 120 = 148$ menit

- $D - G = 13 + 120 = 133$ menit

Dari perhitungan tersebut maka $D - G$ adalah nilai terkecil dan G titik yang dipilih. Maka sisa waktu yang dimiliki menjadi $294 - 133 = 161$ menit. Kemudian perhitungan dilanjutkan dari titik D.

- $G - B = 19 + 200 = 219$ menit

- $G - F = 36 + 120 = 156$ menit

Dari perhitungan tersebut maka $G - F$ adalah nilai terkecil dan F titik yang dipilih. Maka sisa waktu yang dimiliki menjadi $161 - 156 = 5$ menit. Kemudian perhitungan dilanjutkan dari titik F.

- $F - B = 31 + 200 = 231$ menit

Hanya tersisa titik B yang belum dikunjungi tetapi waktu total waktu tidak mencukupi sehingga perhitungan selesai. Jadi titik B tidak masuk rute perjalanan. Maka rute perjalanan optimal yang memenuhi dengan perhitungan *knapsack greedy* adalah :

1. D. Alun-alun Batu = 60 menit

2. G. Selecta = 120 menit

3. F. Coban Rondo = 120 menit.

Dengan total waktu kunjung dan perjalanan adalah 355 menit / 5 jam 55 menit.

3.4.2 Model Perhitungan Kedua

Model perhitungan kedua ini mencari titik terdekat dari titik awal. Titik yang terpilih sebagai rute selanjutnya menjadi titik awal untuk perhitungan berikutnya. Apabila titik terpilih memiliki waktu kunjung, maka waktu tersebut ditambahkan dalam perhitungan. Maka perhitungan dari kasus diatas adalah sebagai berikut.

- $A - B = 3$ menit

- $A - C = 4$ menit

- $A - D = 6$ menit

- $A - E = 18$ menit

- $A - F = 29$ menit

- $A - G = 19$ menit

- $A - H = 19$ menit

- $A - I = 39$ menit

- $A - J = 41$ menit

Dari perhitungan tersebut maka $A - B$ adalah nilai terkecil dan B titik yang dipilih.

Karena B memiliki waktu kunjung, maka waktu tempuh + waktu kunjung = 3 + 200 = 203 menit. Maka, sisa waktu yang dimiliki menjadi 360 – 203 = 157 menit. Kemudian perhitungan dilanjutkan dari titik B.

- B – C = 1 menit
- B – D = 7 menit
- B – E = 25 menit
- B – F = 32 menit
- B – G = 20 menit
- B – H = 18 menit
- B – I = 36 menit
- B – J = 41 menit

Dari perhitungan tersebut maka B – C adalah nilai terkecil dan C titik yang dipilih. Karena C tidak memiliki waktu kunjung, Maka, sisa waktu yang dimiliki menjadi 157 – 1 = 156 menit. Kemudian perhitungan dilanjutkan dari titik C.

- C – D = 8 menit
- C – E = 26 menit
- C – F = 33 menit
- C – G = 21 menit
- C – H = 11 menit
- C – I = 37 menit
- C – J = 42 menit

Dari perhitungan tersebut maka C – D adalah nilai terkecil dan D titik yang dipilih. Karena D memiliki waktu kunjung, maka waktu tempuh + waktu kunjung = 8 + 60 = 68 menit. Maka, sisa waktu yang dimiliki pengguna menjadi 156 – 68 = 88 menit. Kemudian perhitungan dilanjutkan dari titik D.

- D – E = 19 menit
- D – F = 28 menit
- D – G = 13 menit
- D – H = 18 menit
- D – I = 36 menit
- D – J = 40 menit

Dari perhitungan tersebut maka D – G adalah nilai terkecil dan G titik yang dipilih. Karena D memiliki waktu kunjung, maka waktu tempuh + waktu kunjung = 13 + 120 = 133 menit. Tetapi waktu yang tersisa hanya 88 menit jadi tidak mencukupi, maka perhitungan berhenti di titik D. Jadi rute perjalanan optimal yang memenuhi dengan perhitungan kedua adalah :

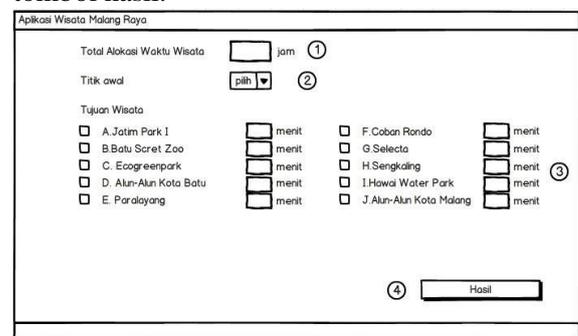
1. B. Batu Secret Zoo = 200 menit
2. C. Ecogreenpark = 0 menit
3. D. Alun-Alun Batu = 60 menit.

Dengan total waktu kunjung dan perjalanan adalah 272 menit / 4 jam 32 menit.

3.3. Perancangan Antarmuka

Dalam Desain antarmuka (interface) untuk tampilan awal aplikasi wisata Malang Raya akan ditunjukkan pada Gambar 3.4. Diawal aplikasi user akan disuguhkan dengan interface dimana user dapat memasukkan data untuk keperluan perencanaan rute wisata yaitu total alokasi waktu wisata, titik awal dan tujuan wisata dimana setiap tujuan wisata dialokasikan waktu dalam satuan menit.

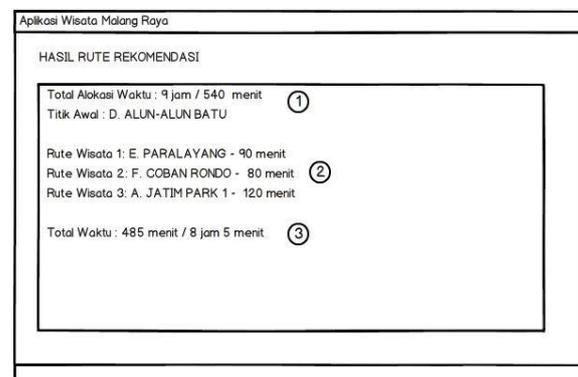
Kemudian setelah selesai memasukkan data yang diperlukan, user menekan tombol hasil pada aplikasi yang terdapat pada Gambar 3.4, kemudian aplikasi tersebut akan mengolah data yang telah dimasukkan oleh user. Gambar 3.5 adalah desain tampilan apabila user menekan tombol hasil.



Gambar 4 Desain Awal Tampilan Aplikasi

Keterangan dari rancangan antarmuka pada Gambar 4 adalah :

1. Ruang untuk pengguna mengisi total alokasi waktu
2. Tab untuk memilih lokasi titik awal
3. Pilihan alternatif tujuan wisata dengan memberikan waktu kunjungnya.
4. Tab hasil untuk memproses perhitungan



Gambar 5 Hasil Rute Optimal Perjalanan

Keterangan dari rancangan antarmuka pada Gambar 5 adalah :

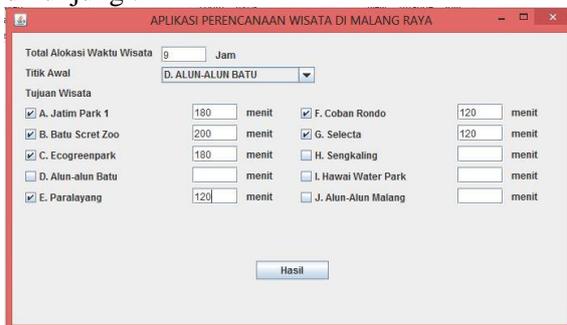
1. Keterangan total waktu alokasi wisata dan titik awal yang di masukkan.

2. Keterangan rute wisata terpilih dari hasil perhitungan aplikasi.
3. Total waktu perjalanan dan waktu kunjung wisata dari hasil perhitungan aplikasi.

4. HASIL DAN PEMBAHASAN

4.1. Contoh Antarmuka

Ketika menjalankan program perencanaan wisata ini, pengguna akan diberikan tampilan seperti Gambar 6 dibawah ini. Pengguna diperlukan untuk mengisi total alokasi waktu wisata, titik awal berangkat dan mengisi waktu masing-masing tempat wisata yang ingin dikunjungi.



Gambar 6 Antarmuka Halaman Awal

Setelah menekan tombol hasil pada halaman utama, program akan menampilkan rute rekomendasi seperti yang ditunjukkan pada Gambar 7.



Gambar 7 Implementasi Halaman Hasil

4.2. Hasil Pengujian

Pada pengujian ini akan dilakukan percobaan sebanyak 10 kali. Pengujian ini dilakukan untuk mencari tingkat akurasi pada jumlah wisata yang dapat dikunjungi dan tingkat akurasi wisata yang dikunjungi. Pengujian dilakukan dengan memberi contoh kasus yang akan dihitung dan mencari semua kemungkinan yang terjadi dengan kombinasi pada setiap titik.

Setelah mendapatkan hasil kombinasi dari perhitungan tersebut kemudian dicari nilai yang paling optimal sesuai dengan kriteria optimal yang sudah dijelaskan pada sub bab Optimasi.

Pada 10 contoh kasus ini menggunakan 4 kasus dengan 3 kombinasi, 3 kasus dengan 4 kombinasi dan 3 kasus dengan 5 kombinasi.

Hasil pengujian dari 10 contoh kasus yang telah diuji ditunjukkan pada Tabel 2.

Hasil akurasi bernilai 1 artinya keluaran dari perhitungan sistem sama dengan perhitungan kombinasi. Sebaliknya, hasil akurasi bernilai 0 artinya hasil perhitungan sistem tidak sama dengan hasil perhitungan kombinasi.

Tabel 2 Tabel Hasil Pengujian Perhitungan Kedua Aplikasi dengan Hasil Perhitungan Kombinasi.

No.	Contoh Kasus	H1	H2	HK	AJ1	AJ2	AH1	AH2
1	Titik awal : C Waktu Kunjung : 6 Jam Alternatif tujuan : E, G, I	E, G	G, E	G, E	1	1	0	1
2	Titik awal : E Waktu Kunjung : 6 Jam Alternatif tujuan : C, G, I	G, C	C, I	C, G	1	1	0	0
3	Titik awal : G Waktu Kunjung : 6 Jam Alternatif tujuan : C, E, I	E, C	C, I	C, I	1	1	0	1
4	Titik awal : I Waktu Kunjung : 6 Jam Alternatif tujuan : C, G, E	E, G	C, G	C, G	1	1	0	1
5	Titik awal : A Waktu Kunjung : 7 Jam Alternatif tujuan : C, E, G, I	E, G	C, G, E	C, G, E	0	1	0	1
6	Titik awal : C Waktu Kunjung : 7 Jam Alternatif tujuan : A, E, G, I	E, G	A, G	A, G	1	1	0	1
7	Titik awal : E Waktu Kunjung : 7 Jam Alternatif tujuan : A, C, G, I	G, C	A, C	A, C	1	1	0	1
8	Titik awal : B Waktu Kunjung : 8 Jam Alternatif tujuan : A, C, E, G, I	E, G, C	C, A, G	C, A, G	1	1	0	1
9	Titik awal : D Waktu Kunjung : 8 Jam Alternatif tujuan : A, C, E, G, I	E, G, C	A, C	C, A, G	1	0	0	0
10	Titik awal : F Waktu Kunjung : 8 Jam Alternatif tujuan : A, C, E, G, I	E, G, C	E, A, C	E, A, C	1	1	0	1

Keterangan :

- H1 : Hasil aplikasi dengan perhitungan model pertama
- H2 : Hasil aplikasi dengan perhitungan model kedua
- HK : Hasil dari perhitungan kombinasi
- AJ1 : Akurasi jumlah wisata yang dapat dikunjungi dengan perhitungan model pertama
- AJ2 : Akurasi jumlah wisata yang dapat dikunjungi dengan perhitungan model kedua
- AH1 : Akurasi tempat wisata yang dapat dikunjungi dengan perhitungan model pertama
- AH2 : Akurasi tempat wisata yang dikunjungi dengan perhitungan model kedua

Berdasarkan Tabel 2 telah dilakukan pengujian akurasi dengan 10 contoh kasus dan menghasilkan nilai akurasi dengan Persamaan (1) :

$$\text{Nilai akurasi} = \frac{\text{Jumlah data akurat}}{\text{jumlah seluruh data}} \times 100\% \quad (1)$$

$$\text{Nilai akurasi AJ1} = \frac{9}{10} \times 100\% = 90\%$$

$$\text{Nilai akurasi AJ2} = \frac{9}{10} \times 100\% = 90\%$$

$$\text{Nilai akurasi AH1} = \frac{0}{10} \times 100\% = 0\%$$

$$\text{Nilai akurasi AH1} = \frac{8}{10} \times 100\% = 80\%$$

Dari perhitungan akurasi jumlah wisata yang dapat dikunjungi, kedua model perhitungan memiliki tingkat akurasi yang sama yaitu 90%. Sedangkan pada tingkat akurasi tempat wisata yang dapat dikunjungi, kedua perhitungan memiliki hasil akurasi yang sangat jauh berbeda yaitu 0% pada model perhitungan pertama dan 80% pada model perhitungan kedua.

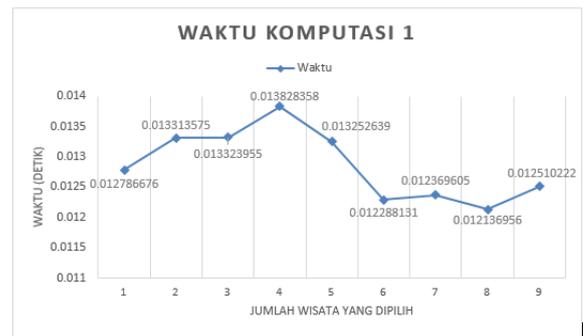
Jadi dapat disimpulkan bahwa, kedua model perhitungan memiliki tingkat akurasi yang cukup baik dalam akurasi jumlah wisata yang dapat dikunjungi. Sedangkan untuk tingkat akurasi tempat wisata yang dapat dikunjungi, model perhitungan pertama tidak memiliki nilai akurasi yang tepat dan hanya model perhitungan kedua yang memiliki tingkat akurasi yang cukup baik.

4.3. Pengujian Performa

Pengujian dilakukan untuk mengetahui waktu komputasi berdasarkan jumlah banyaknya lokasi wisata yang dipilih oleh pengguna. Tujuannya adalah mengetahui apakah dengan semakin banyaknya alternatif lokasi yang dipilih akan mempengaruhi waktu komputasi aplikasi tersebut. Pada pengujian performa ini, percobaan menggunakan alternatif lokasi wisata mulai dari 1-9. Masing-masing percobaan dengan jumlah alternatif lokasi yang sama dilakukan sebanyak 5 kali. Kemudian, dihitung rata-rata dari hasil perhitungan waktu komputasi tersebut.

4.3.1 Model Perhitungan Pertama

Pada Gambar 8 menunjukkan hasil waktu komputasi perhitungan pertama yang didapatkan dengan mencari rata-rata dari 5 kali percobaan.



Gambar 8 Grafik Pengujian Perhitungan Pertama

Melihat dari grafik yang ditunjukkan pada Gambar 8 menunjukkan hasil yang tidak sesuai dengan apa yang diharapkan, dimana dengan jumlah wisata yang dipilih semakin banyak akan mempengaruhi lama waktu komputasi perhitungan. Grafik menunjukkan pola peningkatan dari 1 jumlah wisata sampai dengan 4 jumlah wisata, tetapi mengalami penurunan pada titik selanjutnya dan kemudian menunjukkan hasil yang tidak stabil. Maka dapat disimpulkan bahwa jumlah wisata yang digunakan untuk aplikasi model perhitungan pertama tidak mempengaruhi waktu komputasi sistem.

4.3.2 Model Perhitungan Kedua

Pada Gambar 9 menunjukkan hasil waktu komputasi perhitungan kedua yang didapatkan dengan mencari rata-rata dari 5 kali percobaan.



Gambar 9 Grafik Pengujian Perhitungan Kedua

Sama dengan grafik yang ditunjukkan sebelumnya pada waktu komputasi aplikasi model perhitungan pertama Gambar 8 bahwa waktu komputasi pada aplikasi model perhitungan kedua ini, grafik pada Gambar 9 menunjukkan hasil yang tidak stabil. Sehingga dapat disimpulkan juga bahwa banyaknya lokasi wisata yang digunakan untuk aplikasi model perhitungan kedua ini juga tidak

5. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian untuk perencanaan wisata di Malang Raya dengan metode algoritma *greedy*, maka dapat diambil beberapa kesimpulan yaitu:

1. Dengan menggunakan metode algoritma *greedy* dapat dihasilkan solusi optimum dalam pencarian rute terpendek yang dapat membantu merumuskan hasil rekomendasi rute wisata di Malang Raya. Metode algoritma *greedy* dapat diimplementasikan dengan baik pada perencanaan wisata di Malang Raya. Lokasi wisata yang digunakan hanya 10 berdasarkan data yang didapatkan dari google maps dengan waktu maksimal 9 jam.

2. Hasil pengujian aplikasi dengan 10 contoh kasus permasalahan mendapatkan hasil akurasi sebesar 90% pada kedua model perhitungan dalam mencari jumlah lokasi yang dapat dikunjungi dengan alokasi waktu yang dimiliki. Sedangkan hasil akurasi wisata optimal yang dikunjungi yaitu 0% pada model perhitungan pertama dan 80% pada model perhitungan kedua. Dengan demikian, model perhitungan kedua lebih direkomendasikan untuk digunakan.

3. Pada pengujian performa dengan menghitung waktu komputasi pada aplikasi ini mendapatkan hasil yang tidak tentu pada setiap pengujian dengan jumlah lokasi wisata yang berbeda. Sehingga dapat disimpulkan bahwa jumlah lokasi wisata yang dipilih tidak mempengaruhi performa aplikasi tersebut. Hal yang mungkin dapat mempengaruhi performa aplikasi tersebut disebabkan oleh performa komputer pada saat itu.

6. DAFTAR PUSTAKA

- Defindal, IP., Ariesanda, B. dan Christoforus., 2005. Algoritma Greedy untuk Menentukan Lintasan Terpendek. Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika, Institut Teknologi Bandung
- Hayati, E.N. dan Yohanes, A. PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA GREEDY. Seminar Nasional IENACO-2014. ISSN: 2337-4349. Universitas Stikubank Semarang.
- Kartina, 2010. Penyelesaian Knapsack Problem

Menggunakan Algoritma Genetika. UPN “Veteran” Yogyakarta.

- Kholisoh, D. 2008. Optimasi Numerik. UPN “Veteran” Yogyakarta.
- Kurniasari, Y. 2006. Penerapan Algoritma Greedy. PT Gramedia Pustaka Utama, Jakarta.
- Munir, R. 2005. Strategi Algoritmik, Sekolah Teknik Informatika dan Elektro, Institut Teknologi Bandung.
- Priatmoko, S. 2014. Algoritma Dijkstra Untuk Pencarian Jalur Terdekat Dan Rekomendasi Objek Pariwisata Di Pulau Bali. Program Studi Teknik Informatika, Universitas Dian Nuswantoro Jl. Nakula No.1 Semarang.
- Rachmawati, D dan Chandra, A. 2013. IMPLEMENTASI ALGORITMA GREEDY UNTUK MENYELESAIKAN MASALAH KNAPSACK PROBLEM. Jurnal SAINTIKOM Vol. 12, No. 3, September 2013