

# IMPLEMENTASI ALGORITMA GREEDY PADA PEMODELAN KNAPSACK SEBAGAI PERENCANAAN BIAYA WISATA KOTA SEMARANG

Setiawan Megananta Yuniyanto<sup>1</sup>, Noor Ageng Setiyanto<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

Jl. Nakula I No. 5-11 Semarang & Jl. Imam Bonjol No. 207 Semarang, 50131, (024) 3517261

E-mail : nanta.blora@gmail.com<sup>1</sup>, nasetiyanto@gmail.com<sup>2</sup>

---

## Abstrak

Indonesia memiliki Sumber Daya Alam (SDA) yang sangat eksotis, salah satu pemanfaatannya yaitu pada bidang pariwisata, data yang telah dihimpun oleh Badan Pusat Statistik (BPS) Jawa Tengah yang bekerjasama dengan Dinas Pariwisata Kota Semarang jumlah kunjungan wisatawan mancanegara (wisman) ke Indonesia pada tahun 2013 mencapai 3.442 kunjungan dan sebanyak 1.457.576 kunjungan oleh wisatawan nusantara sama seperti tahun 2012. Untuk itulah agar dapat terus meningkatkan kunjungan wisata khususnya di kota Semarang maka penulis bermaksud membuat sebuah aplikasi yang dapat membantu para wisatawan untuk melakukan perjalanan wisata di kota Semarang. Pemodelan knapsack digunakan untuk memecahkan permasalahan biaya menginap yang akan digunakan wisatawan untuk melakukan perjalanan. Dengan menggunakan Algoritma Greedy yang akan di implementasikan pada pemodelan knapsack diharapkan dapat memberikan solusi perjalanan wisata dengan biaya yang murah dan dapat dilakukan secara mandiri sesuai keinginan wisatawan. Ada beberapa penyelesaian yang bisa digunakan yaitu greedy by weight, greedy by profit, greedy by density.

**Kata Kunci:** Pemodelan Knapsack, Algoritma Greedy, Biaya, Wisata

## Abstract

Indonesia has Natural Resources is very exotic, one of use was that in the field of tourism, the data that has been compiled by the Central Statistics Agency in Central Java in cooperation with the Tourism Office of Semarang number of tourist arrivals to Indonesia on in 2013 reached 3,442 visits and as many as 1,457,576 visits by tourists same as in 2012. For that to be able to continue to increase tourist traffic, especially in the city of Semarang, the author intends to make an application that can help the tourists to travel in the city of Semarang. knapsack modeling is used to solve the problem of the cost of stay will be used by tourists to travel. By using Greedy algorithms that will be implemented in the modeling knapsack expected to provide solutions and travel with low cost and can be done independently as desired tourists. There are some solutions that can be used are greedy by weight, greedy by profit, greedy by density.

**Keywords:** Knapsack modeling, Greedy algorithm, Cost, Travel

## 1. Latar Belakang

Indonesia memiliki Sumber Daya Alam (SDA) yang sangat eksotis, salah satu pemanfaatannya yaitu pada bidang pariwisata, data yang telah dihimpun oleh Badan Pusat Statistik (BPS) Jawa Tengah yang bekerjasama dengan Dinas Pariwisata Kota Semarang jumlah kunjungan wisatawan mancanegara (wisman) ke Indonesia pada tahun 2013 mencapai 3.442 kunjungan dan

sebanyak 1.457.576 kunjungan oleh wisatawan nusantara sama seperti tahun 2012 [1]. Hal ini sungguh disayangkan karena tidak ada peningkatan yang cukup berarti. Setelah berkeliling di sekitar kota Semarang ternyata informasi tentang daerah wisata di kota ini sangatlah terbatas. Kurangnya petunjuk arah yang akurat membuat sebagian orang enggan untuk mengunjungi tempat tempat wisata, selain itu bagi

pengunjung luar daerah Kota Semarang yang baru pertama kali ke Semarang pasti belum mengetahui tempat tempat yang menarik di kota Semarang.

Dan pada era saat ini telah berkembang komunitas *backpeker* atau biasa disebut wisatawan yang melakukan perjalanan wisatawan secara mandiri mulai dari pengaturan jadwal hingga akomodasinya dimana mereka menerapkan prinsip mendapatkan kesenangan sebanyak banyaknya dan mengeluarkan dana sedikit mungkin agar mencapai tempat wisata yang diinginkan [2]. Menanggapi prinsip yang dimiliki *backpeker* penulis berharap dapat memberikan alternatif perjalanan sesuai destinasi yang ingin dituju yang disesuaikan kemampuan dana yang dimiliki oleh wisatawan namun dengan tempat tempat yang berkelas. Dalam hal ini penulis memberikan salah satu alternatif mengenai biaya penginapan dan makan yang akan digunakan selama berada di sekitar kota Semarang, sehingga wisatawan dapat memprediksi biaya yang harus dibawa untuk berwisata tanpa harus takut kekurangan dana .

Dengan pemodelan pemecahan masalah *knapsack* dengan algoritma *greedy* sebagai solusi diharapkan dapat memeberikan hasil yang optimal. Dimana masalah dalam hal ini yaitu mengenai dana yang disediakan. Sehingga dengan pembatas berupa dana diharapkan dapat memberikan beberapa solusi yang dapat digunakan untuk berwisata ke suatu tempat tersebut mulai dari tempat menginap hingga biaya makan dan biaya masuk ketempat wisata jika diperlukan biaya masuk ke tempat lokasi, dengan dana sebagai pembatasnya. Selain itu juga dengan memanfaatkan *Google Maps API* yang menyediakan peta *satelite* seluruh dunia, kita dapat mengembangkan sebuah informasi geografis untuk memberikan informasi letak tempat pariwisata dan tempat menginap yang telah dipilih.

Dalam hal ini penulis menggunakan *latitude* dan *longitude* sehingga dapat membuat tempat tempat yang ingin dituju lebih akurat.

Dari uraian di atas penulis ingin memberikan sebuah solusi perencanaan biaya dengan menggunakan algoritma *greedy* pada pemodelan *knapsack* yang dapat digunakan oleh wisatawan. Selain itu wisatawan dapat mengetahui beberapa tempat-tempat yang dapat dikunjungi di kota Semarang mulai dari tempat wisata, penginapan dan tempat makan yang ada di kota Semarang secara akurat

Adapun atasan dari penelitian ini adalah sebagai berikut :

- 1 Hanya menerapkan pemodelan *knapsack* dengan algoritama *greedy* dalam memberikan solusi optimal untuk perencanaan biaya menginap dan makan diluar biaya belanja dan transportasi.
- 2 Visualisasi peta letak wisata yang dibuat hanya daerah Kota Semarang.
- 3 Aplikasi dibangun dengan berbasis *web*.

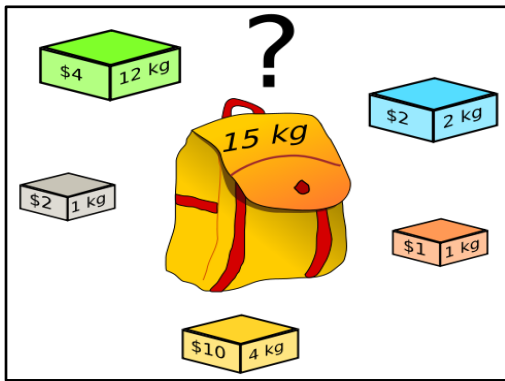
Objek wisata yang ditampilkan berupa wisata bangunan beserjarah yang ada di Kota Semarang.

## 2. Tinjauan Pustaka

Bagian kedua dari isi jurnal adalah metode penelitian, dimana bagian ini berisi tentang langkah-langkah penelitian, objek dan instrumen penelitian, teknik pengumpulan data, variabel penelitian dan teknik analisis.

### 2.1 Pemodelan *Knapsack*

Dilihat dari bahasanya *knapsack* mempunyai arti ransel/tas jika diterjemahkan ke dalam bahasa Indonesia. Sesuai kegunaanya secara harfiah digunkan untuk menampung / membawa sebuah barang dengan ketentuan tertentu. Berikut ilustrasi pemodelan *knapsack* [3] :



**Gambar 1** : Ilustrasi Permasalahan Knapsack

Dapat dilihat dari gambar tersebut terdapat sebuah ransel yang mempunyai kapasitas sebesar 15 Kg, dan terdapat 5 buah barang yang masing-masing mempunyai berat dalam Kg dan *profit* (nilai guna) dalam dollar (\$) yang berbeda-beda dan berat dengan satuan kilogram (kg). Dan persoalannya bagaimana menggantung barang - barang tersebut agar mendapatkan sebuah *profit* yang terbaik namun tidak melebihi dari kapasitas tas tersebut

## 2.2 Algoritma Greedy

Algoritma Greedy merupakan algoritma yang lazim untuk memecahkan persoalan optimasi meskipun hasilnya tidak selalu merupakan solusi yang optimum [3]. Solusi optimum (terbaik) adalah solusi yang bernilai minimum atau maksimum dari sekumpulan alternatif solusi yang mungkin [4]. Prinsip greedy adalah: “take what you can get now!”. Pendekatan yang digunakan di dalam algoritma greedy adalah membuat pilihan yang “tampaknya” memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal (local optimum) pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi optimum global (global optimum) [4].

Ada beberapa elemen yang digunakan pada algoritma *greedy*, elemen-elemennya sebagai berikut [4]:

- Himpunan kandidat.  
Berisi elemen - elemen pembentuk solusi.

- Himpunan solusi  
Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
- Fungsi seleksi (selection function)  
Memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
- Fungsi kelayakan (feasible)  
Memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (constraints) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.
- Fungsi obyektif  
yaitu fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

Pro dan Kontra Algoritma *Greedy* adalah sebagai berikut [5]:

Pro algoritma *greedy*

1. Biasanya (sangat) mudah untuk merancang algoritma serakah
2. Mudah untuk diterapkan dan dapat berjalan dengan cepat karena algoritma *greedy* yang sederhana
3. Beberapa kasus rumit dapat memberikan hasil yang efektif / optimal
4. Mengarah ke heuristik sederhana ketika masalah tidak dipahami dengan baik

Kontra algoritma *greedy*

1. Sangat sering algoritma *greedy* tidak berjalan dengan baik. Mudah membuat kita percaya bahwa algoritma *greedy* bekerja dengan baik.
2. Algoritma *greedy* banyak digunakan untuk memecahkan sebuah masalah namun ada kemungkinan untuk mengatasi sebuah masalah tersebut tidak ada cara yang terstruktur untuk menemukan solusi yang efektif

### 2.3 Penyelesaian *Knapsack* Dengan *Greedy*

Pada permasalahan *knapsack* telah dijelaskan tentang barang apa yang akan dimasukkan kedalam ransel (*knapsack*) yang memiliki kapasitas tertentu. Algoritma *greedy* dapat membantu dan memberikan sebuah solusi yang dapat terlihat optimum secara cepat, terdapat beberapa pendekatan yang digunakan untuk persoalan *knapsack* dengan algoritma *greedy* sebagai berikut:

#### 1. *Greedy by Weight*

Pada setiap langkah dipilih objek yang mempunyai berat teringan. Mencoba memaksimalkan keuntungan dengan memasukkan sebanyak mungkin objek ke dalam *knapsack* [3]. Dalam pendekatan ini diutamakan dengan berat teringan yang akan selalu dipilih pertama hingga kapasitas terpenuhi, bisa kurang dari kapasitas namun tidak boleh melebihi.

#### 2. *Greedy by Profit*

Pada setiap langkah dipilih objek yang mempunyai keuntungan terbesar. Mencoba memaksimalkan keuntungan dengan memilih objek yang paling menguntungkan terlebih dahulu[3]. Dalam pendekatan ini diutamakan memilih keuntungan terbesar namun tetap menggunakan fungsi kelayakan berdasarkan berat(*weight*), jika lolos dalam fungsi kelayakan maka akan dimasukkan ke dalam ransel(*knapsack*) hingga kapasitas terpenuhi, bisa kurang

dari kapasitas namun tidak boleh melebihi.

#### 3. *Greedy by Density*

Pada setiap langkah *knapsack* di isi dengan objek yang mempunyai  $p_i/w_i$  terbesar, dimana  $p$  adalah keuntungan (*profit*) dan  $w$  adalah berat barang (*weight*). Mencoba memaksimalkan keuntungan dengan memilih objek yang mempunyai *density* per unit berat terbesar[3]. Dalam pendekatan ini diutamakan memilih *density* terbesar namun tetap menggunakan fungsi kelayakan berdasarkan berat (*weight*), jika lolos dalam fungsi kelayakan maka akan dimasukkan ke dalam ransel (*knapsack*) hingga kapasitas terpenuhi, bisa kurang dari kapasitas namun tidak boleh melebihi.

Berikut Contoh permasalahan *knapsack* dengan algoritma *greedy*:

Permasalahan *knapsack* dengan 6 objek:

$$w_1 = 100; p_1 = 40$$

$$w_2 = 50; p_2 = 35$$

$$w_3 = 45; p_3 = 18$$

$$w_4 = 20; p_4 = 4$$

$$w_5 = 10; p_5 = 10$$

$$w_6 = 5; p_6 = 2$$

Kapasitas *knapsack*  $W = 100$

**Tabel 1** : Perhitungan *Knapsack* Dengan Algoritma *Greedy*

I	Properti objek			Greedy by			Solusi Optimal
	$w_i$	$p_i$	$p_i/w_i$	Profit	Weight	density	
1	100	40	0,4	1	0	0	0
2	50	35	0,7	0	0	1	1
3	45	18	0,4	0	1	0	1
4	20	4	0,2	0	1	1	0
5	10	10	1,0	0	1	1	0
6	5	2	0,4	0	1	1	0
Total bobot				100	80	85	100
Total keuntungan				40	34	51	55

Pada contoh tersebut dijelaskan bahwa *greedy* tidak mendapatkan solusi optimal yaitu profit 55, namun *greedy* dapat memberikan solusi yang seakan-akan optimal dengan nilai profit 51. Telah dijelaskan sebelumnya bahwa *greedy* terkadang tak memberikan hasil yang tepat optimal namun *greedy* dapat

berjalan dengan cepat dan mudah di implementasikan.

Berikut *pseudo-code* dari algoritma greedy [4]:

```

procedure greedy(input C: himpunan_kandidat;
                output S: himpunan_solusi)
{menentukan solusi optimum dari persoalan optimasi dengan algoritma greedy
Masukan: himpunan_kandidat C
Keluaran: himpunan_solusi S
}
Deklarasi
x : kandidat;

Algoritma:
S ← {}           {inisialisasi S dengan kosong}
while (belum SOLUSI(S) and (C ≠ {})) do
x ← SELEKSI(C); {pilih sebuah kandidat dari C}
C ← C - {x}     {elemen himpunan kandidat berkurang satu}
if LAYAK(S ∪ {x}) then
S ← S ∪ {x}
endif
endwhile
{SOLUSI(S) sudah diperoleh or C = {} }

```

**Gambar 2 :** *pseudo-code* dari algoritma greedy

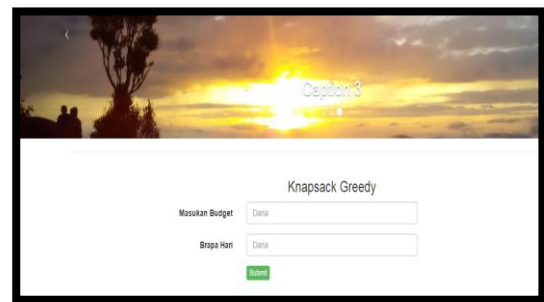
## 2.4 Biaya

Biaya adalah suatu pengorbanan sumber ekonomi yang ditukar dalam satuan uang, untuk mendapatkan barang atau jasa yang diharapkan akan memberikan keuntungan/manfaat pada saat ini atau masa yang akan datang [6]. Dalam penelitian ini biaya digunakan salah satu variabel yang digunakan pada pemodelan masalah *knapsack*, sesuai dengan teori yang telah menjelaskan di atas bahwa masalah *knapsack* dapat dipecahkan dengan algoritma *greedy by weight* dimana biaya (*cost*) dari wisatawan sebagai *wieght*-nya. Berikut definisi – definisi biaya [7]:

1. Hansen dan Mowen (2006), biaya adalah kas atau nilai ekuivalen kas yang dikorbankan untuk mendapatkan barang atau jasa yang diharapkan memberi manfaat saat ini dimasa datang bagi organisasi.
2. Mulyadi (1990), Biaya adalah pengorbanan sumber ekonomi, yang diukur dalam satuan uang, yang telah terjadi atau yang kemungkinan akan terjadi untuk tujuan tertentu.
3. *Internasional Accounting*

*Standards Committee* (IASC) biaya dalam Standar Akuntansi Keuangan (2002), Beban adalah penurunan manfaat ekonomi selama suatu periode akuntansi dalam bentuk arus keluar atau berkurangnya aktiva atau terjadinya kewajiban yang mengakibatkan penurunan ekuitas yang tidak menyangkut pembagian kepada penanam modal.

## 3. Hasil Dan Pembahasan



**Gambar 3 :** Halaman Awal User

### 3.1 Penerapan Algoritma Greedy Pada Knapsack

Berikut hasil dari penerapan *pseudo-code* algoritma *greedy* yang di implmentasikan dengan bahasa pemrograman *PHP* :

```

$dnb=$dana;
$mkn=getMinWg1();
$tx=$lg; $hbv=0;
while($ht=mysql_fetch_array($mkn)){
$ij=$ht['harga'];
$tt=seleksi($ij,$dnb);
if($tt==1 &&
(layak($hbv,$ij,$dnb))==1){
    $temp[$tx]=$ht['kd_mkn'];
    $hbv=$hbv+$ht['harga'];
    $tx++;
}
}

```

Berikut penjelasannya:

\$dnb merupakan sebuah variabel yang menampung nilai dana yang digunakan, dana ini digunakan sebagai pembatas atau bisa disebut sebagai kapasitas *knapsack* (W). Lalu \$mkn merupakan variabel yang menampung himpunan kandidat. Fungsi seleksi (\$ij,\$dnb) di-

gunakan sebagai seleksi, terdapat 2 paramter yaitu variabel \$j<sub>j</sub> merupakan harga yang di dapat dari himpunan kandidat lalu diseleksi apakah harga tersebut kurang dari dana atau tidak dalam hal ini dana di dapat dari variabel \$d<sub>nb</sub>, apabila \$j<sub>j</sub> < \$d<sub>nb</sub> maka akan lanjut ke proses selanjutnya, namun apabila tidak maka akan menyeleksi kandidat selanjutnya. Fungsi layak (\$h<sub>bv</sub>, \$i, \$d<sub>nb</sub>) yaitu fungsi yang megecek apakah bilangan kandidat ini layak atau tidak, yang dimaksud layak yaitu apabila harga pada bilangan kandidat ditambahkan dengan jumlah harga masih dibawah dari dana (pembatas W), atau secara singkatnya \$j<sub>j</sub> + \$h<sub>bv</sub> < \$d<sub>nb</sub> jika kondisi ini terpenuhi maka akan dimasukan pada himpunan solusi pada variabel \$temp.

### 3.2 Tampilan Interface

Berikut hasil tampilan berbasis web : Halaman ini merupakan halaman awal yang dapat diakses oleh user, pada halaman ini user dapat memasukan sebuah inputan berupa dana dan hari. Dimana dana digunakan sebagai kapasitas/pembatas dalam pemodelan *knapsack*, sedangkan hari akan dikalikan dengan himpunan kandidat khusus-nya untuk penginapan yang dihitung berdasarkan hari apabila kita menggunakannya. *Field* dana hanya bisa diisi angka sebanyak 8 digit, sedangkan *field* hari juga hanya bisa diisi angka sebanyak 2 digit.

Hasil Perhitungan Dengan Dana 400000 dan Lama 1 Hari

Pilih Kater: [Mengganti Biosk] • Pilih Destinasi: [Ganti "Cikini"] • Pilih Hari: [2018]

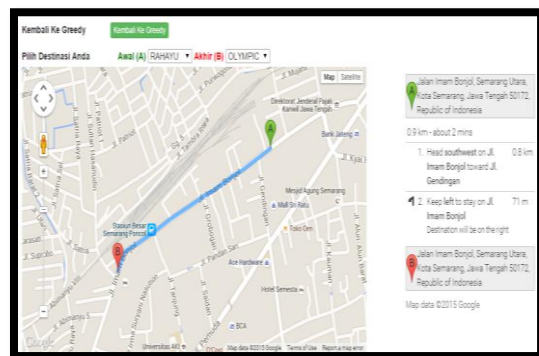
No	Nama Kater	Harga	Nama Hotel	Kapasitas	Amanat	Keterangan
1	STANDARD D1	10000	RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang	Fasilitas: Mezb Dalam, Kapasitas 2 Orang
2	ELITE	10000	OLYMPIC	2	Jalan Imam Bonjol 107 Jawa Tengah	1 Dinding Besi, Kapasitas 1 Orang, Dikawatir

No	Nama Makanan	Harga	Nama Restoran	Kapasitas	Amanat	Keterangan	Total
1	TEH HANGAT	2000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		2000
2	KOPI PAKAI	5000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		5000
3	JERUK HANGAT	5000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		5000
4	ES TEH	5000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		11000
5	KOPI CREASER PAKAI	4000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		15000
6	ES KOP	4000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		19000
7	ES JERUK	4000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		23000
8	ES KOP CREASER	4000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		27000
9	SOFT DRINK	5000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang	gpr, tema, meja kayu	32000
10	SODA BEMBINA	5000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		37000
11	MIE SOPRENO POLICE	5000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		42000
12	MIE SOPRENO POLICE	5000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		47000
13	MIE SOPRENO POLICE	1000	DEPOT RAKATU	1	Jalan Imam Bonjol 25-27 Kota Semarang		58000

Gambar 4 : Halaman Perhitungan User

Pada halaman perhitungan user telah ditampilkan hasil perhitungan menggunakan pemodelan *knapsack* dengan algoritma *greedy*, disitu terlihat ada 3 macam penerapan algoritma *greedy* yaitu *greedy by weight* yang dialiaskan dengan harga termurah, *greedy by profit* dialiaskan dengan kelas teratas, *greedy by density* dialiaskan dengan keuntungan tertinggi. Halaman ini menampilkan harga penginapan dan kamar, daftar menu dan restoran ditampilkan secara detail dari nama hingga harga. Disesuaikan dengan dana dan hari yang telah diinputkan sebelumnya kemudian ditampilkan dengan perhitungan algoritma *greedy*. Apabila ingin melihat lokasi tempat – tempat tersebut terdapat tombol lihat peta.



Gambar 5 : Halman Peta

Pada halaman melihat peta kita dapat melihat lokasi-lokasi yang telah tercatat dalam database, mulai dari tempat wisata, penginapan, hingga restoran. Terdapat pilihan destinasi yang ingin dituju dengan memilih *combobox* yang telah tersedia yang menampilkan tempat – tempat dari hasil perhitungan yang di dapat sebelumnya. Dibagian kanan ditampilkan pula jalan mana saja yang harus dilalui. Sehingga user dapat mengetahui arah menuju lokasi yang diinginkan, bila ingin melihat kembali perhitungan terdapat tombol kembali ke perhitungan.

Kode Wisata	Nama	Alamat	Latitude	Longitude	Masuk Lokal	Masuk Manca	Aksi
WIS-0001	GEREJA BLENDUK KOTA LAWA	Jalan Ledgend Supripto No 32 Kota Lama Semarang	-6.968781	110.426561	0	0	<a href="#">Edit</a> <a href="#">Delete</a>
WIS-0002	VISWA BLEDHAGAYA	Jl Perintis Kemerdekaan Putatopang Banyuwangi, Semarang	-7.06875	110.409196	0	0	<a href="#">Edit</a> <a href="#">Delete</a>
WIS-0003	LAWANG BESUK	Jalan Perintis, Komplek Tugu Wisata, Semarang, Jawa Tengah	-6.963273	110.410009	10000	10000	<a href="#">Edit</a> <a href="#">Delete</a>
WIS-0004	MUSEUM HANICHA	Jalan HSR Bojoparoto No.2 Semarang	-6.96407	110.410165	0	0	<a href="#">Edit</a> <a href="#">Delete</a>

**Gambar 6 :** Halaman Utama

Halaman utama (*dashboard*) akan menampilkan semua data yang tersimpan di *database* yang ditampilkan menggunakan tabel dilengkapi dengan pencarian (*search*) dan jumlah data yang akan ditampilkan (*records per page*). Akan ditampilkan data tempat wisata, penginapan, kamar, tempat makan, makanan/minuman. Pada setiap baris data terdapat tombol *edit* yang digunakan untuk merubah data apabila ada kesalahan dan tombol *delete* yang digunakan untuk menghapus data apabila data tersebut telah tidak digunakan. Tombol edit berwarna biru dan tombol delete berwarna merah.

Form Input Tempat Wisata

Kode Tempat Wisata: WIS-0001

Nama Tempat Wisata: Nama

Alamat: Alamat

Biaya Masuk Lokal:
 

- Biaya Turis Lokal
- Manca
- Biaya Turis Mancanegara

Latitude: \_\_\_\_\_

Longitude: \_\_\_\_\_

**Gambar 7 :** Halaman Input Tempat Wisata

Halaman input tempat wisata digunakan untuk menyimpan data tempat wisata yang telah di dapat ke dalam *database*. Adapun field yang tersedia yaitu kode tempat wisata, nama tempat wisata, alamat, biaya masuk lokal dan mancanegara, latitude dan longitude. Untuk kode tempat wisata otomatis akan muncul secara *unix* dan tidak sama

antara satu dengan yang lainnya. Untuk *longitude* dan *latitude* hanya bisa diinput dengan angka desimal dengan tanda titik sebagai pemisah, serta tanda negatif (-) yang digunakan untuk membentuk koordinat agar lebih akurat saat digunakan pada peta. Selain itu juga untuk inputan biaya baik lokal ataupun mancanegara hanya bisa diinput dengan angka. Untuk *field* lainya dapat diisi sesuai dengan kebutuhan dari nama dari yang ada pada form tersebut.

Form Input Tempat Makan/Minuman

Kode Resto: RST-0005

Nama Resto: Nama

Alamat: Alamat

Bintang/Kelas: Non Bintang

Dalam Hotel: Tidak

Latitude: \_\_\_\_\_

Longitude: \_\_\_\_\_

**Gambar 8 :** Halaman Input Tempat Makan

Halaman input tempat makan digunakan untuk menyimpan data tempat makan yang telah di dapat ke dalam *database*. Adapun field yang tersedia yaitu kode tempat makan, nama tempat makan, alamat, bintang/kelas, pada hotel atau tidak, latitude dan longitude. Untuk kode tempat makan otomatis akan muncul secara *unix* dan tidak sama antara satu dengan yang lainnya. Untuk *longitude* dan *latitude* hanya bisa diinput dengan angka desimal dengan tanda titik sebagai pemisah, serta tanda negatif (-) yang digunakan untuk membentuk koordinat agar lebih akurat saat digunakan pada peta. Untuk *field* dalam hotel ditunjukan untuk restoran yang ada di dalam penginapan yang telah diinput terlebih dahulu. Nama penginapan otomatis diambil dari *database* dan nantinya dimungkinkan akan ada relasi antara tabel tempat makan dan penginapan. Untuk *field* bintang/kelas tersedia terdapat 6 pilihan mulai dari non-bintang hingga bintang 5 dengan default

non-bintang.

**Gambar 9** : Halman Input Penginapan

Halaman input penginapan digunakan untuk menyimpan penginapan yang telah di dapat ke dalam *database*. Adapun field yang tersedia yaitu kode penginapan, nama penginapan, alamat, bintang/kelas, latitude dan lonngitude. Untuk kode penginapan otomatis akan muncul secara *unix* dan tidak sama antara satu dengan yang lainnya. Untuk *longitude* dan *latitude* hanya bisa diinput dengan angka desimal dengan tanda titik sebagai pemisah, serta tanda negatif (-) yang digunakan untuk membentuk koordinat agar lebih akurat saat digunakan pada peta. Untuk *field* bintang/kelas tersedia terdapat 6 pilihan mulai dari non-bintang hingga bintang 5 dengan default non-bintang.

**Gambar 10** : Halaman Input Makanan/Minuman

Halaman input makanan/minuman digunakan untuk menyimpan data makanan dan minuman dari setiap restoran yang telah di dapat kemudian disimpan ke dalam *database*. Adapun field yang tersedia yaitu kode

makanan/minuman, nama restoran, jenis menu, harga, keterangan. Untuk kode makanan/minuman otomatis akan muncul secara *unix* dan tidak sama antara satu dengan yang lainnya. Untuk *field* nama restoran otomatis akan keluar dan data diambil dari *database*. nantinya akan ada relasi antara tabel tempat makan dan makanan. Untuk *field* jenis makanan hanya ada 2 pilihan yaitu makanan dan minuman. Sedangkan untuk field harga hanya bisa diinput dengan angka.

**Gambar 11**: Halaman Input Kamar

Halaman input kamar wisata digunakan untuk menyimpan kamar dari setiap penginapan yang telah di dapat kemudian disimpan ke dalam *database*. Adapun field yang tersedia yaitu kode kamar, nama penginapan, nama kamar, harga, keterangan. Untuk *field* nama hotel otomatis akan keluar dan data diambil dari *database*. nantinya akan ada relasi antara tabel penginapan dan kamar. Untuk *field* harga hanya bisa diisi dengan angka.

## 4. Kesimpulan dan Saran

### 4.1 Kesimpulan

Penulis memiliki beberapa kesimpulan, berikut kesimpulannya :

- a. Dengan menggunakan pemodelan *knapsack* dapat memberikan beberapa solusi tentang



perencanaan biaya wisata mulai dari tempat menginap hingga menu makanan dan minuman dengan rincian biaya yang dapat digunakan untuk *user* mempersiapkan biaya untuk berwisata di kota Semarang.

- b. Algoritma *greedy* dapat diterapkan dalam pemodelan *knapsack* sehingga dapat memberikan sebuah solusi perencanaan biaya wisata di kota Semarang. Sesuai dengan prinsip *greedy* yaitu selalu memilih “yang paling” dalam hal ini memilih yang paling murah, memilih kelas tertinggi, dan juga keuntungan tertinggi

#### 4.2 Saran

Aplikasi ini dapat dikembangkan kembali dengan penambahan beberapa penyempurnaan, berikut saran yang diberikan oleh penulis :

- a. Penambahan biaya akomodasi selama melakukan perjalanan wisata juga dapat ditambahkan untuk menyempurnakan aplikasi ini.
- b. Untuk kelas/bintang pada susunan menu makanan dan minuman bisa lebih ditingkatkan pada makanan dan minuman secara langsung bukan mengikuti bintang/kelas pada restoran, sehingga solusi yang di dapat lebih akurat.

#### 5. DAFTAR PUSTAKA

- [1] Bappeda Kota Semarang dan Badan Pusat Statistik Kota Semarang, “Semarang Dalam Angka 2013,” Bappeda, Semarang, 2014.
- [2] A. Primasari dan T. Siswojo, “Promosi Kuliner Lokal Sebagai Daya Jual Pariwisata Indonesia Untuk Backpacker Asing,” Program Studi Sarjana Desain Komunikasi Visual Fakultas Seni Rupa dan Desain ITB, Bandung.
- [3] D. Rachmawati dan A. Candra, “IMPLEMENTASI ALGORITMA GREEDY UNTUK MENYELESAIKAN MASALAH KNAPSACK PROBLEM,” *Jurnal Ilmiah Sains dan Komputer*, vol. III, pp. 185-192, 2013.
- [4] R. Munir, *Algoritma Greedy*, Bandung: Departemen Teknik Informatika Institut Teknologi Bandung, 2004.
- [5] Jeff Erickson, Sariel Har-Peled, Chandra Chekuri and the slides of Mahesh Viswanathan and the books by Kleinberg and Tardos, and by Dasgupta, Papadimitriou and Vazirani, “Engineering IT Shared Services,” 1 March 2011. [Online]. Available: [https://courses.engr.illinois.edu/cs473/sp2011/lectures/11\\_notes.pdf](https://courses.engr.illinois.edu/cs473/sp2011/lectures/11_notes.pdf). [Diakses 26 November 2014].
- [6] D. Akuntansi Biaya, Semarang: UPT Perpustakaan Universitas Diponegoro, 2009.
- [7] M. Akbar, “ANALISIS PENERAPAN METODE ACTIVITY BASED COSTING SYSTEM DALAM PENENTUAN HARGA POKOK KAMAR HOTEL PADA HOTEL COKLAT MAKASSAR,” JURUSAN AKUNTANSI FAKULTAS EKONOMI UNIVERSITAS HASANUDIN, Makasar, 2011.