

# SISTEM PENENTUAN LINTASAN TERPENDEK TRAVELING SALESMAN PROBLEM DENGAN ALGORITMA SIMPLE HILL CLIMBING

Abdul Mukthi Chifdhi<sup>1</sup>, Dwi Puspitasari<sup>2</sup>

Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang

<sup>1</sup>[amChifdhi@gmail.com](mailto:amChifdhi@gmail.com), <sup>2</sup>[dwi\\_sti@yahoo.com](mailto:dwi_sti@yahoo.com)

## Abstrak

Traveling Salesman Problem (TSP) merupakan permasalahan pencarian rute perjalanan terpendek oleh seorang kurier/salesman untuk melewati sejumlah alamat dengan jalur tertentu sehingga setiap alamat hanya terlewati satu kali dan perjalanan diakhiri dengan kembali ke kota semula. Banyak algoritma yang digunakan untuk menyelesaikan masalah TSP, tetapi ada beberapa yang algoritma yang kurang cocok. Salah satu algoritma yang dapat menyelesaikan masalah TSP adalah Simple Hill Climbing. Simple Hill Climbing merupakan algoritma yang didasarkan pada pemilihan *new state* secara langsung yang memiliki rute terbaik (terpendek) dari pada rute lainnya tanpa menghiraukan rute-rute yang lainnya. Pendekatan menggunakan algoritma Simple Hill Climbing ini diharapkan mampu memberikan solusi dalam perhitungan waktu yang lebih singkat dibandingkan dengan sejumlah algoritma lain. Tujuan yang ingin dicapai adalah mengaplikasikan metode heuristik Simple Hill Climbing untuk mensimulasikan dan menyelesaikan permasalahan Travelling Salesman Problem dengan baik untuk mendapatkan rute terpendek yang di aplikasikan langsung ke google map untuk mendapatkan jarak dan rute yang nyata (riil).

**Kata kunci :** *traveling salesman problem, sistem lintasan terpendek, hill climbing*

## 1. Pendahuluan

Ketatnya persaingan antara perusahaan mendorong setiap perusahaan untuk memiliki perencanaan dan strategi dalam menjalankan kegiatan usahanya. Perusahaan sangat memerlukan adanya karyawan yang dapat bekerja efektif, efisien, dan ekonomis. Karena selama ini sering kali terjadi keterlambatan datangnya paket menuju alamat yang ditujuh. Oleh karena itu dalam menjalankan fungsi manajemen, diperlukan adanya informasi pendukung yang relevan dan akurat untuk pengambilan keputusan. Selain itu, juga berguna untuk penyusunan perencanaan strategis bisnis perusahaan serta pengendalian kegiatan perusahaan secara efektif dan efisien dalam mencapai tujuannya.

Travelling Salesman Problem- TSP merupakan persoalan optimasi untuk mencari perjalanan terpendek bagi yang ingin berkunjung ke beberapa kota, dan kembali ke kota asal keberangkatan. Cara termudah untuk menyelesaikan TSP yaitu dengan mencoba semua kemungkinan rute dan mencari rute yang terpendek.

## 2. Travelling salesman problem (TSP)

Travelling salesman problem (TSP) adalah permasalahan yang sangat terkenal dalam Teori Graf. Permasalahan pada TSP adalah mencari sirkuit terpendek pada suatu graf tidak berarah yang berasal dari suatu simpul dengan melewati seluruh simpul

dan kembali ke simpul asal.

## 3. Simple Hill Climbing

*Simple Hill Climbing* adalah proses pengujian yang dilakukan dengan menggunakan fungsi heuristik. Pembangkitan keadaan berikutnya sangat tergantung pada feedback dari prosedur pengetesan. Tes yang berupa fungsi heuristik ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnya yang mungkin.

### 3.1 Algoritma

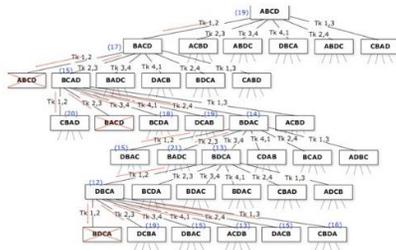
Berikut adalah algoritma dari *Simple Hill Climbing*:

1. Mulai dari keadaan awal, lakukan pengujian: jika merupakan tujuan, maka berhenti; dan jika tidak, lanjutkan dengan keadaan sekarang sebagai keadaan awal.
2. Kerjakan langkah-langkah berikut sampai solusinya ditemukan, atau sampai tidak ada operator baru yang akan diaplikasikan pada keadaan sekarang:
  - a) Cari operator yang belum pernah digunakan; gunakan operator ini untuk mendapatkan keadaan yang baru.
  - b) Evaluasi keadaan baru tersebut.
    - Jika keadaan baru merupakan tujuan, keluar.
    - Jika bukan tujuan, namun nilainya lebih baik

- daripada keadaan sekarang, maka jadikan keadaan baru tersebut menjadi keadaan sekarang
3. Jika keadaan baru tidak lebih baik daripada keadaan sekarang, maka lanjutkan iterasi.

Pada *simple hill climbing* ini, ada 3 masalah yang mungkin, yaitu:

- 1) Algoritma akan berhenti kalau mencapai nilai optimum lokal.
- 2) Urutan penggunaan operator akan sangat berpengaruh pada penemuan solusi.
- 3) Tidak diijinkan untuk melihat satupun langkah sebelumnya.0 mm. Persamaan-persamaan yang membutuhkan tempat lebih dari satu kolom, penulisannya dimungkinkan melintasi 2 kolom.



Gambar 1. Contoh metode *simple hill climbing* dengan 6 operator

Operator yang akan kita gunakan, adalah menukar urutan posisi 2 kota dalam suatu lintasan. Apabila ada n kota, dan kita ingin mencari kombinasi lintasan dengan menukar posisi urutan 2 kota, maka kita akan mendapatkan sebanyak :

$$\frac{n!}{2!(n-2)!} \quad (1)$$

Sehingga kalau ada 4 kota, kita bisa memperoleh :  $\frac{4!}{2!(4-2)!} = 6$  kombinasi. Keenam kombinasi ini akan kita pakai semuanya sebagai operator, yaitu:

- \* Tukar 1, 2 (menukar urutan posisi kota ke-1 dengan kota ke-2).
- \* Tukar 2, 3 (menukar urutan posisi kota ke-2 dengan kota ke-3).
- \* Tukar 3, 4 (menukar urutan posisi kota ke-3 dengan kota ke-4).
- \* Tukar 4, 1 (menukar urutan posisi kota ke-4 dengan kota ke-1).
- \* Tukar 2, 4 (menukar urutan posisi kota ke-2 dengan kota ke-4).
- \* Tukar 1, 3 (menukar urutan posisi kota ke-1 dengan kota ke-3).

Pada Gambar 1 terlihat bahwa, pada keadaan awal, lintasan terpilih adalah ABCD (=19). Pada level pertama, *hill climbing* akan mengunjungi BACD

(=17) yang ternyata memiliki nilai heuristik lebih kecil dibandingkan dengan ABCD (17<19), sehingga BACD menjadi pilihan selanjutnya dengan operator terpakai Tukar1,2. Pada level kedua, *hill climbing* akan mengunjungi ABCD. Karena operator Tukar 1, 2 sudah digunakan oleh BACD, maka viii dipilih *node* yang lain yaitu BCAD (=15). Karena nilai heuristik BCAD lebih kecil dibanding dengan BACD (15<17), maka *node* BCAD akan menjadi pilihan selanjutnya dengan operator Tukar2,3. Kemudian *hill climbing* akan mengunjungi CBAD (=20). Karena nilai heuristik CBAD lebih besar jika dibanding dengan BCAD (20>17), maka dipilih *node* lain. Pencarian menuju ke *node* BACD, karena operator Tukar2,3 sudah pernah digunakan oleh BCAD, maka dipilih *node* lain. Kunjungan berikutnya ke *node* BCDA (=18). Nilai inipun masih lebih besar dari nilai heuristik BCAD, sehingga dipilih *node* lain. *Node* yang dikunjungi berikutnya adalah DCAB (=19). Nilai heuristic DCAB ternyata juga lebih besar dibanding dengan BCAD, sehingga pencarian dilanjutkan di *node* lainnya lagi, yaitu BDAC (=14). Nilai heuristik ini sudah lebih kecil daripada nilai heuristik *node* BCAD (14<15), maka sekarang *node* ini yang akan diekplorasi. Pencarian pertama ditemukan *node* DBAC (=21), yang lebih besar daripada nilai BDAC. Nilai heuristik yang lebih kecil diperoleh pada *node* BDCA (=13). Sehingga *node* BDCA ini akan diekplorasi. Pencarian pertama sudah mendapatkan *node* dengan nilai heuristik yang lebih kecil, yaitu DBCA (=12). Sehingga *node* ini diekplorasi juga. Dari hasil ekplorasi dengan pemakaian semua operator, ternyata sudah tidak ada *node* yang memiliki nilai heuristik yang lebih kecil dibanding dengan nilai heuristik DBCA, sehingga sebenarnya *node* DBCA (=12) inilah lintasan terpendek yang kita cari (SOLUSI).

#### 4. Google Maps

*Google Maps* adalah layanan aplikasi peta online yang disediakan oleh *Google* secara gratis. Layanan peta *Google Maps* secara resmi dapat diakses melalui situs <http://maps.google.com>. Pada situs tersebut dapat dilihat informasi geografis pada hampir semua permukaan di bumi kecuali daerah kutub utara dan selatan. Layanan ini dibuat sangat interaktif, karena di dalamnya peta dapat digeser sesuai keinginan pengguna, mengubah level *zoom*, serta mengubah tampilan jenis peta. *Google Maps* mempunyai banyak fasilitas yang dapat dipergunakan misalnya pencarian lokasi dengan memasukkan kata kunci, kata kunci yang dimaksud seperti nama tempat, kota, atau jalan, fasilitas lainnya yaitu perhitungan rute perjalanan dari satu tempat ke tempat lainnya

## 5. Uji Coba

Berikut merupakan penjelasan dari setiap pengujian sistem.

### 5.1 Jumlah Alamat: 1.1.a

Data masukkan : Jumlah Alamat (Angka). Hasil yang diharapkan : Masuk textfield yang merupakan inputan dari nama-nama alamat.

#### Input Jumlah Alamat

Jumlah Kota :

### 5.2 Jumlah Alamat: 1.1.b

Data masukkan : Jumlah Alamat (Angka). Hasil yang diharapkan : Jumlah Alamat tidak dapat mengenerate jumlah alamat. Validasi : Jumlah Alamat kosong.

Langkah pengujian :

- Mengkosongkan *jumlah alamat* untuk masuk ke dalam halaman form nama-nama alamat.
  - Kemudian pilih *button Hitung*, maka akan muncul pesan seperti dibawah ini.
- Kesimpulan : Diterima  
Hasil Capture :

#### Input Jumlah Alamat

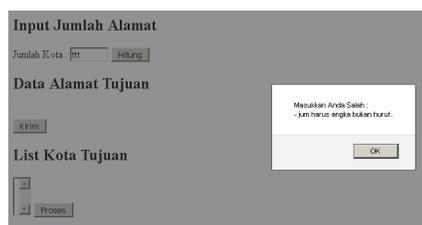
Jumlah Kota :

### 5.3 Jumlah Alamat : 1.1.c

Data masukkan : Jumlah Alamat (Angka). Hasil yang diharapkan : Jumlah Alamat tidak valid (bukan angka). Validasi: Jumlah Alamat tidak valid

Langkah pengujian :

- *Jumlah Alamat* yang dimasukkan bukan angka tapi huruf.
- Kesimpulan : Diterima  
Hasil Capture:



### 5.4 Masukkan Nama-Nama Alamat: 2.1.a

Data masukkan : Data Alamat Hasil yang diharapkan : Data tersimpan pada suatu array. Validasi : Data Lengkap.

Langkah pengujian :

- Mengisi semua data textfield yang ditampilkan dengan alamat tujuan

- Untuk penulisan alamat tidak menggunakan tanda spasi.  
Kesimpulan : Diterima  
Hasil Capture:

#### Input Jumlah Alamat

Jumlah Kota :

#### Data Alamat Tujuan

JALANSUKARNOHATTA  
JALANGAJAYANA  
JALANVETERAN  
JALANJOYOTAMBAKSARI

### 5.5 Masukkan Nama-Nama Alamat: 2.1.b

Data masukkan : Data Alamat Hasil yang diharapkan : Data tidak bisa tersimpan pada array. Validasi : Data Kosong.

Langkah pengujian :

- Tidak mengisi data alamat.
- Kesimpulan : Diterima  
Hasil Capture:

#### Input Jumlah Alamat

Jumlah Kota :

#### Data Alamat Tujuan

### 5.6 Memilih Nama-Nama Alamat:

Data masukkan : Data Alamat. Hasil yang diharapkan : Data tersimpan pada suatu array.. Validasi : Data dipilih semua.

Langkah pengujian :

- Memilih semua data alamat yang ada dalam *form list menu multiple*.
- Kesimpulan : Diterima  
Hasil Capture:

#### Input Jumlah Alamat

Jumlah Kota :

#### Data Alamat Tujuan

#### List Kota Tujuan

JALANSUKARNOHATTA  
JALANGAJAYANA  
JALANVETERAN  
JALANJOYOTAMBAKSARI

**5.7 Memilih Nama-Nama Alamat:**

Data masukkan : Data Alamat. Hasil yang diharapkan : Data tersimpan pada suatu array..  
Validasi : Data dipilih sebagian/ sesuai keinginan user.

- Langkah pengujian :
- Memilih data alamat yang ada dalam *form list menu multiple* menurut user.
- Kesimpulan : Diterima  
Hasil Capture:

**Input Jumlah Alamat**

Jumlah Kota :

**Data Alamat Tujuan**

**List Kota Tujuan**

JALANSUKARNOHATTA  
JALANGAJAYANA  
JALANVETERAN  
JALANDJOYOTAMBAKSARI

**5.8 Memilih Nama-Nama Alamat:**

Data masukkan : Data Alamat. Hasil yang diharapkan : Data tidak tersimpan pada suatu array..  
Validasi : Data tidak dipilih semua.

- Langkah pengujian :
- Tidak emilih data alamat yang ada dalam *form list menu multiple*.
- Kesimpulan : Diterima  
Hasil Capture:

**Input Jumlah Alamat**

Jumlah Kota :

**Data Alamat Tujuan**

**List Kota Tujuan**

JALANSUKARNOHATTA  
JALANGAJAYANA  
JALANVETERAN  
JALANDJOYOTAMBAKSARI

**5.9 Melihat Jarak:**

Data masukkan : Data Alamat. Hasil yang diharapkan : Data alamat hasil dari Metode Hill Climbing yang di sampingnya ada link jarak untuk mengetahui jarak.

- Langkah pengujian :
- Jika “jarak” diklik maka akan mengarahkan ke halaman rute dan jarak dari setiap alamat.
- Kesimpulan : Diterima  
Hasil Capture:

**Input Jumlah Alamat**

Jumlah Kota :

**Data Alamat Tujuan**

**List Kota Tujuan**

- 1. JALANGAJAYANA, MALANG JALANSUKARNOHATTA, MALANG JALANVETERAN, MALANG JALANDJOYOTAMBAKSARI, MALANG [jarak](#)
- 2. JALANVETERAN, MALANG JALANGAJAYANA, MALANG JALANSUKARNOHATTA, MALANG JALANDJOYOTAMBAKSARI, MALANG [jarak](#)
- 3. JALANDJOYOTAMBAKSARI, MALANG JALANVETERAN, MALANG JALANGAJAYANA, MALANG JALANSUKARNOHATTA, MALANG [jarak](#)
- 4. JALANGAJAYANA, MALANG JALANDJOYOTAMBAKSARI, MALANG JALANVETERAN, MALANG JALANSUKARNOHATTA, MALANG [jarak](#)
- 5. JALANSUKARNOHATTA, MALANG JALANGAJAYANA, MALANG JALANDJOYOTAMBAKSARI, MALANG JALANVETERAN, MALANG [jarak](#)
- 6. JALANVETERAN, MALANG JALANSUKARNOHATTA, MALANG JALANGAJAYANA, MALANG JALANDJOYOTAMBAKSARI, MALANG [jarak](#)

Hasil yang diharapkan : Data alamat dihitung jaraknya dan di tampilkan rutenya digoogle map. Validasi : Data dipilih semua.

- Langkah pengujian :
- Memilih semua data alamat yang ada pada *form list menu multiple*.
  - Kemudian pilih button Rute, maka akan muncul hasil seperti dibawah ini.
  - Setiap link jarak memiliki jarak yang berbeda-beda, jarak yang paling kecil maka itu adalah solusi yang digunakan untuk rute pengiriman di perusahaan jasa pengiriman barang.
- Kesimpulan : Diterima  
Hasil Capture:

**List Tujuan Pengiriman**

JALANSUKARNOHATTA, MALANG  
JALANVETERAN, MALANG  
JALANDJOYOTAMBAKSARI, MALANG

**Total Distance: 15.835 km**

**Jalan Gajayana, Kecamatan Lowokwaru, Kota Malang, Jawa Timur, Indonesia**

2.8 km - about 8 mins

1. Head south on Jl. Gajayana toward Jl. Kerto Raharjo 0.1 km
2. Turn left onto Jl. Kerto Raharjo 0.1 km
3. Turn right toward Jl. Kertosentono 14 m
4. Turn left onto Jl. Kertosentono 0.2 km
5. Turn left onto Jl. Watugong 0.3 km
6. Turn right onto Jl. Mt. Haryono 0.4 km
7. Turn left onto Jl. Soekarno Hatta 1.4 km
8. Make a U-turn 0.2 km

**Jalan Soekarno Hatta, Kecamatan Lowokwaru, Kota Malang, Jawa Timur, Indonesia**

3.3 km - about 8 mins

1. Head southwest on Jl. Soekarno Hatta 1.2 km
2. Turn left onto Jl. Mayjend D.I. Panjaitan 0.7 km
3. Continue onto Jl. Mayjend D.I. Panjaitan 0.5 km
4. Turn right onto Jl. Bogor Atas 0.3 km
5. Turn right onto Jl. Veteran Malang 0.6 km

**Jalan Veteran Malang, Kota Malang, Jawa Timur 65145, Indonesia**

5.	Continue onto <b>Jl. Mayjend D.I. Panjaitan</b>	0.7 km
6.	Continue onto <b>Jl. Mayjend D.I. Panjaitan</b>	1.1 km
7.	At the roundabout, take the 1st exit onto <b>Jl. Brigjen Slamet Riadi</b>	1.5 km
8.	Turn left onto <b>Jl. Cempol-Malang/Jl. Jaka Agung Suprpto</b>	1.2 km
9.	Turn right onto <b>Jl. Wr. Supratman</b>	0.4 km
10.	Continue onto <b>Jl. Raya Mahakam</b>	0.1 km
11.	At the roundabout, take the 2nd exit onto <b>Jl. Hamid Rusdi</b>	0.9 km

Jalan Hamid Rusdi, Blimbing, Kota Malang, Jawa Timur, Indonesia

Map data ©2015 Google

## 6. Kesimpulan dan Saran

Kesimpulan pada jurnal kali ini, agar bisa diberikannya sistemasi GPS pada system tersebut, sebagaimana kurir bisa langsung bersedia mengantarkan barang dengan mudah dan cepat.

### Daftar Pustaka:

- <http://code.google.com/apis/maps/documentation/v3/introduction.html>. (t.thn.). Dipetik Mei 10, 2015, dari <http://code.google.com>:
- <http://code.google.com/apis/maps/documentation/v3/introduction.html>
- <https://developers.google.com/maps/documentation/javascript/examples/directions-waypoints>. (t.thn.). Dipetik Juni 5, 2015, dari <https://developers.google.com/maps/documentation/javascript/examples/directions-waypoints>
- Kusumadewi, S. (2003). *Artificial Intelligence*. Yogyakarta: Graha Ilmu.
- Novitasari, D., Welsari, A., Yunita, W., Alfiyah, N., & P, C. (2014). Penyelesaian Travelling Salesman Problem Dengan Algoritma Simple Hill Climbing. 1-5.
- PENYELESAIAN TRAVELLING SALESMAN PROBLEM (TSP) DENGAN MENGGUNAKAN ALGORITMA HILL CLIMBING*. (t.thn.). Dipetik Mei 15, 2015, dari <http://karya-ilmiah.um.ac.id/index.php/matematika/article/view/8882>
- Shodiq, A. (2010). *Tutorial Dasar Pemrograman Google Maps API*.