

Penyelesaian *Travelling Salesman Problem* (TSP) Menggunakan Algoritma *Hill Climbing* dan MATLAB

Muhammad Irfan

Fakultas Keguruan dan Ilmu Pendidikan, Universitas Sarjanawiyata Tamansiswa

muhammad.irfan@ustjogja.ac.id

Abstrak. *Travelling Salesman Problem* (TSP) adalah permasalahan dimana seorang salesman harus mengunjungi semua kota yang mana tiap kota hanya dikunjungi sekali, dan harus kembali ke kota asal. Masalah utama yang dihadapi sebuah TSP adalah bagaimana mencari rute terpendek dari perjalanan seorang salesman dengan biaya minimum. Penelitian ini bertujuan untuk mengetahui penyelesaian masalah TSP dengan menggunakan algoritma *Hill Climbing* baik *Simple Hill Climbing* (SHC) maupun *Steepest-Ascent Hill Climbing* (SAHC). Berdasarkan hasil kajian teori yang dilakukan, dapat disimpulkan bahwa algoritma *Hill Climbing* dapat digunakan untuk menyelesaikan TSP. Algoritma *Hill Climbing* dalam menyelesaikan masalah TSP yaitu: menentukan *initial state*, melakukan pengujian panjang lintasan, melakukan kombinasi penukaran dua kota, dan kemudian melakukan pengujian terhadap nilai heuristiknya. Penyelesaian masalah TSP dengan menggunakan algoritma *Hill Climbing* masing-masing mempunyai karakteristik yang berbeda-beda. Dalam menyelesaikan masalah TSP, SHC memilih keadaan yang lebih baik tanpa melakukan pengujian dikombinasi pertukaran kota pada iterasi yang sama. Sedangkan SAHC membandingkan keadaan yang lebih baik sebelum memilih keadaan tersebut sebagai *new state*. Selanjutnya untuk membantu proses komputasi pada saat menyelesaikan TSP, dibuat sebuah program dengan model algoritma *Hill Climbing* menggunakan MATLAB. Program yang terbentuk memberikan solusi berupa rute perjalanan yang disajikan dalam bentuk diagram.

Kata kunci: Algoritma, *Hill Climbing*, *Travelling Salesman Problem*

Abstract. (*Completion of Traveling Salesman Problem (TSP) Using the Hill Climbing and MATLAB Algorithms*) Traveling Salesman Problem (TSP) is a problem where a salesman must visit all cities where each city is visited only once, and he/she must return to the hometown. The main problem facing a TSP is how to find the shortest route of a sales trip with minimum cost. This study aims to determine the solution of TSP problems using Hill Climbing algorithm both Simple Hill Climbing (SHC) and Steepest-Ascent Hill Climbing (SAHC). Based on the result of the theoretical study, it can be concluded that Hill Climbing algorithm can be used to solve TSP. Hill Climbing algorithm in solving TSP problem is: determining initial state, conducting track length test, performing combination of two city exchanges, and then testing the heuristic value. TSP problem solving using Hill Climbing algorithm each have different characteristics. In solving the TSP problem, SHC chooses a better state, without performing tests in combination of city exchanges on the same iteration. While SAHC compares the situation better before choosing the state as a new state. Furthermore, to assist the computing process when completing the TSP then made a program model Hill Climbing algorithm using MATLAB language. The program is formed to provide solutions in the form of travel routes presented in the form of diagrams.

Keywords: Algorithm, *Hill Climbing*, *Travelling Salesman Problem*

1. Pendahuluan

Setiap hari manusia dihadapkan pada banyak permasalahan hidup yang semakin kompleks dan terkadang sering menjadi terbiasa dengan permasalahan tersebut, karena permasalahan tersebut selalu muncul di dalam setiap aktivitas. Salah satu dari permasalahan tersebut adalah pencarian rute terpendek. Sebagai contoh sebuah perusahaan akan membangun instalasi air bersih menggunakan pipa-pipa untuk mendistribusikan air bersih dari kota satu ke kota yang lain maka perusahaan tersebut akan mencari jarak terpendek untuk meminimumkan biaya. Contoh lain misalnya sebuah perusahaan tekstil yang mendapat pesanan dalam skala besar dan kondisi perusahaan tersebut mempunyai beberapa mesin penghasil kain dengan kemampuan tertentu dalam menghasilkan jenis kain yang berbeda maka diperlukan suatu strategi jitu untuk memenuhi pesanan sehingga menghasilkan keuntungan maksimal tetapi dengan biaya produksi minimum.

Untuk menyelesaikan masalah tersebut, berbagai disiplin ilmu mengembangkan metode-metode untuk menyelesaikan kasus pencarian rute terpendek, salah satunya yaitu *Artificial Intelligence* (AI). *Artificial Intelligence* sebagai cabang ilmu yang berusaha memahami kecerdasan manusia menyajikan bermacam-macam pokok bahasan di antaranya adalah teknik pencarian atau *searching*. Metode pencarian dibedakan menjadi dua jenis yaitu metode pencarian buta/tanpa informasi atau *blind/un-informed search* dan metode pencarian heuristik/dengan informasi atau *heuristic/informed search*. Alasan digunakannya kata *blind* atau buta pada jenis pertama metode pencarian karena tidak adanya informasi awal yang digunakan untuk awal proses pencarian sedangkan kata heuristik pada jenis kedua mengandung arti bahwa metode pencarian jenis ini mempunyai informasi awal sebagai acuan untuk melakukan proses selanjutnya dan metode ini juga mempunyai aturan tertentu dalam proses pencariannya untuk mencapai tujuan. Dalam metode pencarian heuristik terdapat bermacam-macam teknik pencarian dan salah satunya adalah Hill Climbing (Pendakian Bukit). Teknik pencarian Hill Climbing (HC) dibagi menjadi dua jenis yaitu *Simple HC* (HC sederhana) dan *Steepest-Ascent HC* (HC dengan memilih kemiringan yang paling tajam/curam) (Suyanto, 2007 : 23).

Dengan menggunakan metode pencarian HC diharapkan dapat memberikan solusi pada masalah optimasi. Untuk membantu perhitungan dalam penyelesaian masalah optimasi maka dibuatlah suatu algoritma HC yang diimplementasikan dengan MATLAB untuk mempermudah perhitungan dan akan diaplikasikan untuk mencari solusi beberapa masalah optimasi.

Algoritma HC menjadi salah satu cara yang dapat menjadi alternatif pilihan untuk menyelesaikan masalah TSP, karena algoritma ini dapat dengan cepat menemukan rute terdekat dari suatu permasalahan yang ada. Akan tetapi, algoritma ini juga mempunyai beberapa kelemahan – kelemahan. Oleh karena itu, algoritma ini menjadi sesuatu hal yang menarik untuk dapat dibahas dan di kaji kelebihan dan kelemahan dari dua metode, yaitu *simple hill climbing* (SHC) dan *steepest ascent hill climbing* (SAHC).

1.1 Travelling Salesman Problem

Permasalahan TSP (*Traveling Salesman Problem*) adalah permasalahan di mana seorang *salesman* harus mengunjungi semua kota di mana tiap kota hanya dikunjungi sekali, dan dia harus mulai dari dan kembali ke kota asal. Tujuannya adalah menentukan rute dengan jarak total atau biaya yang paling minimum.

Selain masalah sarana transportasi, efisiensi pengiriman surat atau barang ditentukan pula oleh lintasan yang diambil untuk mengirimkan surat atau barang tersebut. Oleh karena itu solusi optimal dari permasalahan TSP ini, akan sangat membantu perusahaan pengiriman surat atau barang untuk mengefisienkan proses pengiriman barang, baik dari segi waktu maupun dana.

1.2 Algoritma Hill Climbing

Hill Climbing merupakan salah satu metode yang masuk ke dalam metode pencarian heuristik.

Metode pencarian *heuristic/informed* pencarian solusinya dilakukan menggunakan keadaan awal (*initial state*) dan aturan produksi berupa fungsi heuristik (suatu fungsi untuk menghitung nilai atau biaya perkiraan dari suatu solusi permasalahan yang dicari) sebagai modal untuk melakukan iterasi menuju goal state.

Hill Climbing (HC) dibagi menjadi dua bagian, yaitu: *Simple Hill Climbing* (SHC) dan *Steepest-Ascent Hill Climbing* (SAHC). Secara garis besar, langkah – langkah dari HC adalah sebagai berikut:

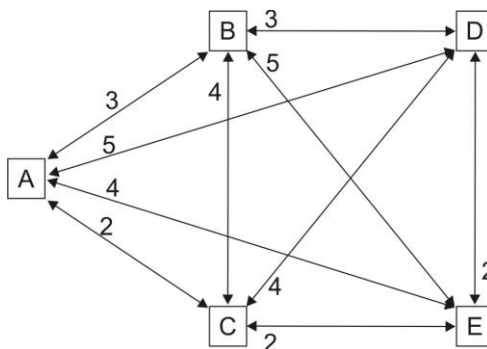
- Dibentuk lintasan awal sebagai keadaan awal (*initial state*), kemudian dilakukan pengujian dari keadaan awal tersebut. Pengujian dilakukan dengan menghitung total jarak yang ditempuh pada *initial state*. Pengujian tersebut dilakukan untuk mendapatkan nilai heuristik sebagai pembandingan terhadap nilai – nilai heuristik yang ada setelah dilakukan kombinasi pertukaran dua kota. Hasil dari pengujian tersebut adalah jarak minimal yang ditempuh pada permasalahan TSP. Hasil tersebut akan digunakan sebagai pembandingan hasil dari langkah berikutnya.
- Setelah dibentuk lintasan awal sebagai keadaan awal, langkah selanjutnya yaitu melakukan kombinasi penukaran dua kota kemudian dilakukan pengujian seperti pada poin *a*, hingga mencapai kondisi goal, jika tidak maka pilihlah elemen yang lain hingga mencapai goal. Kondisi goal diartikan sebagai keadaan yang merupakan solusi optimum dari permasalahan TSP dengan kondisi: *panjang jalur baru < panjang jalur lama*.
- Jika kondisi goal telah tercapai maka kondisi tersebut adalah solusinya, jika tidak, maka bukan solusinya.

2. Hasil dan Pembahasan

2.1 Algoritma Simple Hill Climbing dalam Menyelesaikan TSP

Algoritma SHC untuk menyelesaikan masalah TSP adalah:

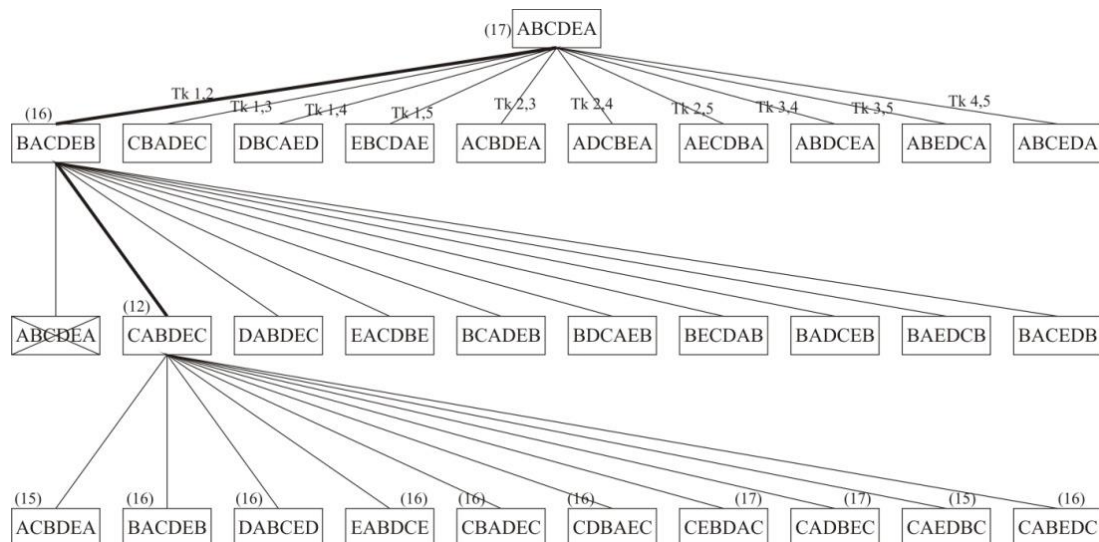
- Evaluasi *initial state*. Jika *initial state* adalah *goal state* maka jadikan *state* ini sebagai solusi dan keluar dari program. Jika bukan *goal state*, lanjutkan proses dengan *initial state* sebagai *current state*.
- Ulangi sampai solusi ditemukan atau sampai tidak ada operator baru yang dapat diaplikasikan terhadap *current state*:
 - Pilih operator yang belum diaplikasikan terhadap *current state* dan aplikasikan operator tersebut sehingga menghasilkan *new state*.
 - Evaluasi *new state* :
 - Jika *state* ini merupakan *goal state* maka jadikan *state* ini sebagai solusi dan keluar dari program.
 - Jika *state* ini bukan *goal state* tetapi lebih baik dari *current state* maka jadikan *state* ini sebagai *current state* baru.
 - Jika *state* ini tidak lebih baik dari *current state* maka kembali ke langkah 2.a.



Gambar 1. Chinese Postman Problem

2.2 Persoalan Tukang Pos Cina (Chinese Postman Problem)

Seorang tukang pos akan mengantarkan surat ke alamat-alamat sepanjang jalan di suatu daerah. Bagaimana ia merencanakan rute perjalanannya supaya ia melewati setiap jalan tepat sekali dan kembali lagi ke tempat awal keberangkatan.



Gambar 2. Diagram SHC

Karena sudah tidak ada node yang memiliki nilai heuristik yang lebih kecil dibanding nilai heuristik CABDEC=12, maka node CABDEC =12 adalah lintasan terpendek (SOLUSI). Tukang pos tersebut dalam mengantarkan surat ke alamat-alamat yang dituju agar menempuh rute terpendek harus menempuh rute dari C – A – B – D – E - C dengan panjang lintasan yaitu 12 km.

Berikut adalah penggalan script dengan menggunakan MATLAB untuk menyelesaikan TSP dengan algoritma SHC.

```
%Iterasi
Initial_State = 0;
Initial_State_Awal = 0;
PjgJalur = PjgJalur_Awal;
%Matrik kombinasi operator tukar dua kota
a = combntns(1:N,2);
Banyak_Operator = size(a,1);
disp(['Banyaknya operator penukaran kota = ' num2str(Banyak_Operator)]);
disp('=====');
A = zeros(Banyak_Operator,N); %Matrik penukaran rute kota pada setiap level
B = zeros(Banyak_Operator,1); %Matrik pjg_rute pada setiap level
Iterasi = 1;
t = 1;
while Iterasi == 1,
    tic
    Cur_Rute = Rute;
    Cur_PjgJalur = PjgJalur;
    disp(['Level ke ' num2str(t)]);
    it_level = 1;
    i = 1;
    while ( it_level == 1 ) & ( i <= Banyak_Operator),
        Rute_1 = Rute;
        b = a(i,1);
        c = a(i,2);
        d = Rute(1,c);
        Rute(1,c) = Rute(1,b);
        Rute(1,b) = d;
        Initial_State = Initial_State_Awal;
```

```

        for k = 1:1:(N-1),
            Initial_State = Initial_State + D(Rute(1,k),Rute(1,k+1));
        end
        Initial_State = Initial_State + D(Rute(1,N),Rute(1,1));
        disp(['      Rute >> ' num2str(Rute) '      Panjang Rute = '
num2str(Initial_State)]);
        if Initial_State < PjgJalur,
            it_level = 0;
            PjgJalur = Initial_State;
            Rute = Rute;
        else
            it_level = 1;
            PjgJalur = PjgJalur;
            Rute = Rute_1;
        end
        i = i + 1;
    end
    if Cur_PjgJalur == PjgJalur,
        Iterasi = 0;
    else
        Iterasi = 1;
    end
    disp([' **Rute terpendek sementara >> ' num2str(Rute) '      Panjang = '
num2str(PjgJalur)]);
    disp('=====');
    time(t) = toc;
    t = t + 1;
end
Rute_Terpendek = Rute;
Panjang_Rute_Terpendek = PjgJalur;
hold on
P = X(Rute_Terpendek,:);
P = [P;P(1,:)];
plot(P(:,1),P(:,2),'bo-','linewidth',1.5);
hold off
disp(['Lama proses ( dalam detik ) = ' num2str(sum(time(:,:)))]);
disp(['Optimum pada level ke ' num2str(t-2)]);

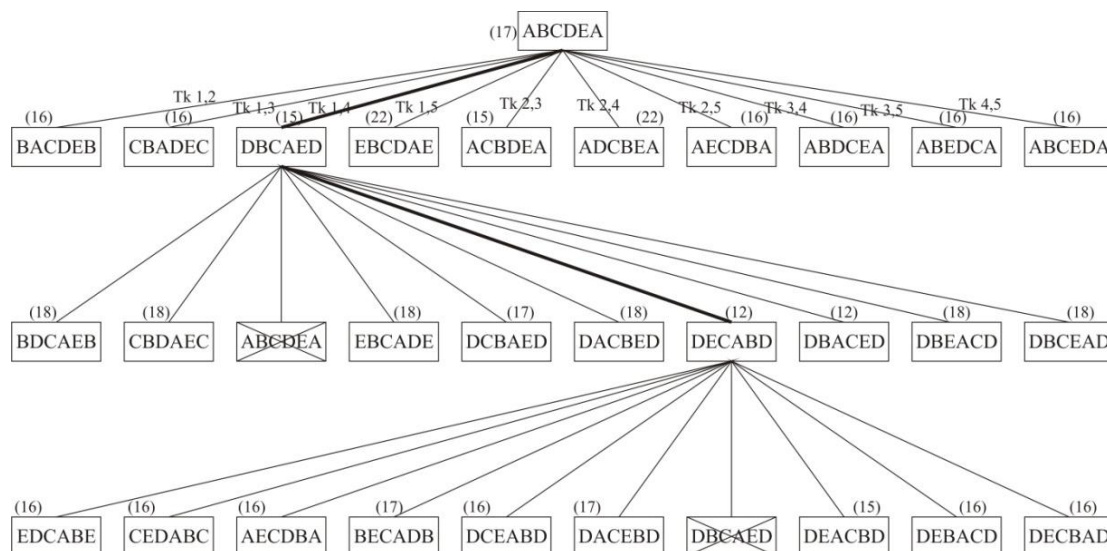
```

1.3 Algoritma Steepest-Ascent Hill Climbing dalam Menyelesaikan TSP

Berikut ini adalah algoritma dari *Steepest – Ascent Hill Climbing* (SAHC):

1. Evaluasi *initial state*. Jika *initial state* adalah *goal state* maka jadikan *state* ini sebagai solusi dan keluar dari program. Jika bukan *goal state*, lanjutkan proses dengan *initial state* sebagai *current state*.
2. Ulangi sampai solusi ditemukan atau sampai tidak ada perubahan terhadap *current state* :
3.
 - a. Misalkan SUK adalah suatu *state* yang menjadi suksesor dari *current state*.
 - b. Untuk setiap operator yang bisa dilakukan terhadap *current state*, kerjakan :
 - i. Aplikasikan semua operator yang ada, untuk menemukan *new state*.
 - ii. Evaluasi *new state*. Jika merupakan *goal state*, jadikan ini sebagai solusi dan keluar dari program. Jika bukan *goal state*, bandingkan dengan *new state* dengan SUK. Jika *new state* lebih baik dari SUK maka ganti SUK dengan *new state*. Jika *new state* tidak lebih baik dari SUK, tidak perlu diganti.
4. Jika SUK lebih baik dari *current state* maka ganti *current state* dengan SUK.

Pada permasalahan Tukang Pos Cina (*Chinese Postman Problem*) yang sebelumnya, akan dicari solusinya dengan menggunakan SAHC. Berikut adalah diagram dari iterasi – iterasi di atas:



Gambar 3. Diagram SAHC

Karena sudah tidak ada node yang memiliki nilai heuristik yang lebih kecil dibanding nilai heuristik DECABD=12, maka node DECABD=12 adalah lintasan terpendek (SOLUSI). Sehingga tukang pos tersebut dalam mengantar surat ke alamat-alamat yang dituju agar menempuh rute terpendek harus menempuh rute dari D – E – C – A – B - D dengan panjang lintasan yaitu 12 km.

Berikut adalah penggalan script dengan menggunakan MATLAB untuk menyelesaikan TSP dengan algoritma SAHC.

```
%Iterasi
Initial_State = 0;
Initial_State_Awal = 0;
PjgJalur = PjgJalur_Awal;
%Matrik kombinasi operator tukar dua kota
a = combntns(1:N,2);
Banyak_Operator = size(a,1);
disp(['Banyaknya operator penukaran kota = ' num2str(Banyak_Operator)]);
disp('=====');
A = zeros(Banyak_Operator,N);%Matrik penukaran kota pada setiap level
B = zeros(Banyak_Operator,1);%Matrik pjg_rute pada setiap level
Iterasi = 1;
t = 1;
while Iterasi == 1,
    tic
    disp(['Level ke ' num2str(t)]);
    Rute_1 = Rute;
    for i = 1:1:Banyak_Operator,
        b = a(i,1);
        c = a(i,2);
        d = Rute(1,c);
        Rute(1,c) = Rute(1,b);
        Rute(1,b) = d;
        A(i,:) = Rute;
        Initial_State = Initial_State_Awal;
        for k = 1:1:(N-1),
            Initial_State = Initial_State + D(A(i,k),A(i,k+1));
        end
        Initial_State = Initial_State + D(A(i,N),A(i,1));
        B(i,:) = Initial_State;
        disp(['Rute >> ' num2str(Rute) ' Panjang Rute = '
num2str(Initial_State)]);
        Initial_State = Initial_State_Awal;
        Rute = Rute_1;
    end
    C = [A B];
```

```

e = C(1,size(C,2));
Rute = C(1,1:N);
for l = 1:1:Banyak_Operator,
    if C(l,size(C,2)) < e,
        e = C(l,size(C,2));
        Rute = C(l,1:N);
    else
        e = e;
        Rute = Rute;
    end
end
if e < PjgJalur,
    Iterasi = 1;
    PjgJalur = e;
    Rute = Rute;
else
    Iterasi = 0;
    PjgJalur = PjgJalur;
    Rute = Rute_1;
end
disp([' **Rute terpendek sementara >> ' num2str(Rute) '          Panjang = '
num2str(PjgJalur)]);
disp('=====');
t = t + 1;
time(t) = toc;
end
Rute_Terpendek = Rute;
Panjang_Rute_Terpendek = PjgJalur;
hold on
P = X(Rute_Terpendek,:);
P = [P;P(1,:)];
plot(P(:,1),P(:,2),'bo-','linewidth',1.5);
hold off
disp(['Lama proses ( dalam detik) = ' num2str(sum(time(:,:)))]);
disp(['Optimum pada level ke ' num2str(t-2)]);

```

Dari hasil analisis rute terpendek dengan menggunakan MATLAB tersebut, maka hasil tersebut dapat disajikan dalam bentuk tabel sebagai berikut:

Tabel 1. Analisis Rute Terpendek

No.	Banyak kota	SHC			SAHC		
		Panjang rute	Lama proses (s)	Optimum Level ke-	Panjang rute	Lama proses (s)	Optimum Level ke-
1	5	10,2333	0,076508	0	10,2333	0,076508	0
2	10	29,5368	1,3792	10	29,3997	1,7662	5
3	15	48,9862	7,3775	21	45,0327	6,1237	8
4	21	76,0776	19,676	25	73,0083	17,8481	12
5	31	145,4273	43,9535	53	129,0912	50,0819	15
6	60	727,6823	427,1647	97	669,0115	473,6006	37

Dari tabel 1, terlihat perbandingan hasil dari SHC dengan SAHC, baik panjang rute, lama proses perhitungan, dan level optimumnya. Pencarian solusi TSP dengan menggunakan metode SAHC membutuhkan waktu lebih lama, akan tetapi panjang rute yang diperoleh serta pencapaian optimumnya lebih baik dari pada SHC.

5. Kesimpulan

Berdasarkan hasil pembahasan mengenai penyelesaian TSP menggunakan algoritma *Hill Climbing*, maka dapat disimpulkan bahwa: (1) Algoritma *Hill Climbing* dapat digunakan untuk menyelesaikan TSP; (2) Baik SHC maupun SAHC mempunyai dasar yang sama, yaitu kombinasi penukaran dua kota; (3) Dalam melakukan pengujian panjang rute SHC dikerjakan dari kiri ke kanan. Ketika ada panjang rute yang lebih kecil dari pada yang sebelumnya, maka nilai dan lintasan tersebut menjadi *new state* dan iterasi berhenti; (4) Sedangkan SAHC harus membandingkan nilai heuristik dari seluruh lintasan dan dipilih nilai heuristik yang terkecil untuk dijadikan *new state*. Dari kedua metode tersebut, SAHC lebih efektif dalam menyelesaikan masalah TSP dibandingkan dengan SHC. Hanya saja SAHC membutuhkan waktu yang lebih lama dari pada SHC dalam proses pencarian solusi.

Referensi

- [1] Aulia Rahma Amin, Mukhammad Ikhsan, Lastiko Wibisono. *Travelling Salesman Problem*. Penelitian. ITB. 2004.
- [2] Brian R Hunt, Ronald L Lipsman, Jonathan M Rosenberg. *A Guide to MATLAB for Beginners and Experienced Users*. Cambridge: University Press. 1995.
- [3] Edhy, Susanta. *Teori dan Praktek Pemrograman Turbo Pascal*. Yogyakarta: Graha Ilmu. 2005.
- [4] Ferdian, Filman. “*Penyelesaian Travelling salesman Problem dengan Algoritma Heuristik*”. Penelitian. Institut Teknologi Bandung. Bandung. 1971.
- [5] Hamdy Taha. *Riset Operasi*. Jilid satu. Jakarta: Binarupa Aksara. 1996.
- [6] Irfan, M. *Penyelesaian Travelling Salesman Problem (TSP) Menggunakan Algoritma Hill Climbing*. Doctoral dissertation, UNY. 2011.
- [7] Kallrath, Julia. *Online Storage Systems and Transportation Problems with Applications*. Boston: Springer Science , Business Media, Inc. 2005.
- [8] Karris, Steven T. *Circuit Analysis with MATLAB Applications*. California: Orchard Publications. 2003
- [9] Ruhul Amin Sarker, Hussein Aly Abbas, Charles Newton. *Heuristics and Optimization for Knowledge Discovery*. United States: Idea Group Publishing. 2002.
- [10] Russel, Stuart., Norvig, Peter. *Artificial Intelligence: A Modern Approach*. New York: Prentice Hall, Inc. 1995.
- [11] Sri Kusumadewi, Hari Purnomo. *Penyelesaian Masalah Optimasi Menggunakan Teknik-teknik Heuristik*. Yogyakarta: Graha Ilmu. 2005.
- [12] Suyanto, *Artificial Intelligence Searching, Reasoning, Planning, and Learnig*. Bandung: Informatika. 2007.
- [13] Wahid, F. *Dasar–dasar Algoritma dan Pemrograman*. Yogyakarta: Penerbit Andi. 2004.
- [14] I. Fadhillah, Y. Permanasari, E. Harahap. *Representasi Matriks untuk Proses Crossover Pada Algoritma Genetika untuk Optimasi Travelling Salesman Problem*. Jurnal Matematika, Universitas Islam Bandung. Vol. 16, No. 1, Mei 2017.
- [15] Respitawulan. *Properti Eigen Untuk Graf k-Regular Tak Terhubung*. Jurnal Matematika, Universitas Islam Bandung. Vol. 16, No. 2, Desember 2017.