

Penyelesaian TSP Simetris dengan Algoritma Greedy Dua Arah

Dibuat oleh : Samuel Cahyawijaya / 13509082
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509082@students.if.itb.ac.id

Abstraksi—Permasalahan Travelling Salesperson Problem (TSP) adalah sebuah permasalahan yang sampai saat ini belum ditemukan solusi optimalnya. Berbagai cara telah diujikan untuk menyelesaikan persoalan TSP ini, namun hasil yang ada belum memuaskan karena memiliki permasalahan dalam hal kompleksitas waktu atau keoptimalan solusi yang dihasilkan. Dalam karya tulis ini akan diujikan penyelesaian masalah TSP dengan algoritma Greedy dua arah. Karya tulis ini terdiri dari 5 bagian, yaitu : Pendahuluan, Landasan teori, Pembahasan Algoritma Greedy dua Arah, Perbandingan optimalitas solusi antara Algoritma Greedy Dua Arah dan Program Dinamis, Kesimpulan dan Saran. Pendahuluan berisi latar belakang mengapa dilakukan pengujian TSP dengan algoritma greedy yang dikembangkan. Landasan teori berisi teori-teori yang perlu dipahami untuk dapat mengerti hal yang akan dibahas pada bagian berikutnya. Bagian Pembahasan Algoritma Greedy dua arah berisi teori dasar dan cara kerja dari algoritma greedy dua arah. Bagian Perbandingan Optimalitas Solusi antara Algoritma Greedy Dua Arah dan Program dinamis menampilkan bagan dan analisis dari hasil yang ada. Bagian kesimpulan dan saran berisi kesimpulan dan saran mengenai algoritma greedy dua arah dan permasalahan TSP

Kata Kunci—Greedy, TSP, NP-hard, NP-complete, optimalitas. Greedy dua arah

I. PENDAHULUAN

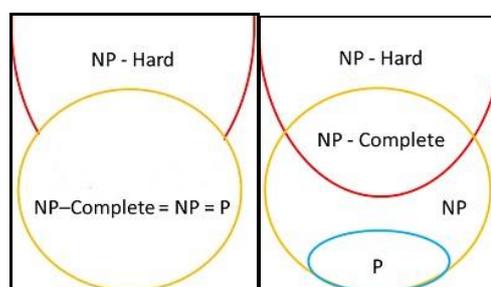
TSP (Travelling Salesperson Problem) adalah sebuah permasalahan dalam ilmu komputasi yang diumpamakan dengan seorang sales yang hendak pergi ke beberapa kota, namun hanya mengunjungi setiap kota satu kali dan kembali ke kota awal setelah mengunjungi seluruh kota. TSP didefinisikan pertama kali pada tahun 1800 oleh seorang matematikawan Irish bernama W.R. Hamilton dan seorang matematikawan Inggris bernama Thomas Kirkman^[1].

Permasalahan TSP telah dicoba untuk diselesaikan dengan berbagai macam cara, namun belum dapat ditemukan hasil yang optimal. Salah satu penyelesaian permasalahan TSP adalah dengan menggunakan algoritma Greedy atau dikenal juga dengan *Nearest Neighbour Algorithm*. Berbagai macam pengembangan algoritma

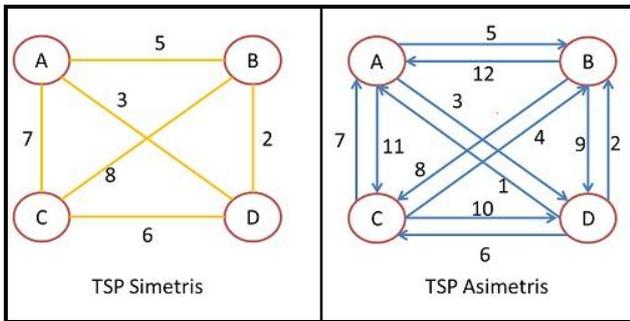
Greedy telah diujikan untuk menyelesaikan masalah TSP. Dalam karya tulis ini, akan diujikan penyelesaian permasalahan TSP dengan menggunakan Greedy yang dilakukan melalui dua arah dan perbandingan ketepatan solusi antara algoritma greedy dua arah dan program dinamis.

II. LANDASAN TEORI

TSP (Travelling Salesperson problem) dalam permasalahan mencari jalur tempuh minimum merupakan sebuah permasalahan yang termasuk kedalam kategori permasalahan NP-hard. Permasalahan NP-hard adalah permasalahan yang memiliki solusi non polinomial tidak ada solusi polinomialnya. Sedangkan dalam persoalan keputusan (contoh : apakah jarak minimum untuk suatu persoalan TSP lebih kecil dari X satuan), TSP termasuk kedalam kategori NP-Complete, yaitu permasalahan yang masih mungkin ditemukan solusi polinomialnya. Terdapat 2 jenis TSP, yaitu TSP simetris dan TSP asimetris. Pada TSP simetris, jarak tempuh dari suatu kota A ke kota lain ,kota B, sama dengan jarak tempuh dari kota B ke kota A. Sedangkan pada TSP asimetris, jarak untuk pergi dan kembali antar kota A dan kota B tidaklah sama.



Gambar 1. Diagram euler untuk permasalahan P, NP, NP-complete, dan NP-hard

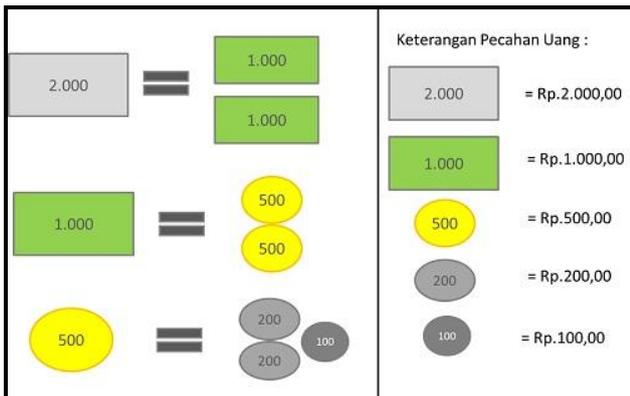


Gambar 2. TSP Simetris dan Asimetris

Greedy adalah suatu algoritma yang dilakukan dengan mengambil solusi-solusi optimal lokal untuk mencapai solusi optimal global. Algoritma Greedy memiliki 5 komponen utama, yaitu :

1. Himpunan kandidat
2. Himpunan solusi
3. Fungsi objektif
4. Fungsi seleksi
5. Fungsi kelayakan

Algoritma Greedy memiliki waktu yang cepat (kompleksitas waktu rendah), namun hanya menghasilkan solusi yang optimal dalam permasalahan – permasalahan tertentu. Contoh permasalahan yang dapat menemukan hasil optimal untuk algoritma greedy adalah permasalahan penukaran uang, dimana untuk setiap penukaran uang menjadi pecahan yang lebih kecil, selalu diambil pecahan dengan nilai uang terbesar yang paling mendekati nilai nominal uang yang akan ditukar.



Gambar 3. Penyelesaian permasalahan penukaran uang dengan Algoritma Greedy

III. PEMBAHASAN ALGORITMA GREEDY DUA ARAH

Algoritma greedy dua arah dibuat berdasarkan definisi algoritma greedy pada umumnya. Algoritma greedy dua arah ini dilakukan dengan mengasumsikan titik pertama pada persoalan TSP sebagai titik awal perjalanan. Komponen utama yang dimiliki oleh greedy dua arah adalah :

1. Titik ujung kiri
Satu dari dua titik ujung dari jalur yang telah dilalui dari titik awal.
2. Titik ujung kanan
Satu dari dua titik ujung dari jalur yang telah dilalui dari titik awal.
3. Himpunan kandidat
Himpunan semua jalur yang dapat dilalui dari titik ujung kiri dan titik ujung kanan
4. Himpunan jalur kiri
Bagian dari himpunan solusi yang terbentuk dari penelusuran terhadap titik ujung kiri
5. Himpunan jalur kanan
Bagian dari himpunan solusi yang terbentuk dari penelusuran terhadap titik ujung kanan
6. Himpunan solusi
Himpunan yang terbentuk dari penggabungan antara himpunan jalur kiri dan himpunan jalur kanan.
7. Fungsi objektif
Fungsi yang menentukan tujuan dari suatu algoritma greedy. Dalam greedy 2 arah ini, fungsi objektif adalah menciptakan jalur yang minimum hingga jumlah jalur sama dengan jumlah titik
8. Fungsi seleksi
Fungsi untuk menentukan pilihan yang terbaik dari antara pilihan –pilihan jalur lain yang telah lolos fungsi kelayakan. Dalam greedy 2 arah ini, fungsi seleksi adalah nilai minimal dari penjumlahan jalur terpendek yang akan dilalui dengan jalur minimal selanjutnya yang akan dilewati
9. Fungsi kelayakan
Fungsi untuk menentukan apakah sebuah pilihan layak untuk dipilih atau tidak. Dalam greedy 2 arah ini ,fungsi kelayakan adalah apakah jalur yang akan dilalui sudah pernah dilalui atau belum

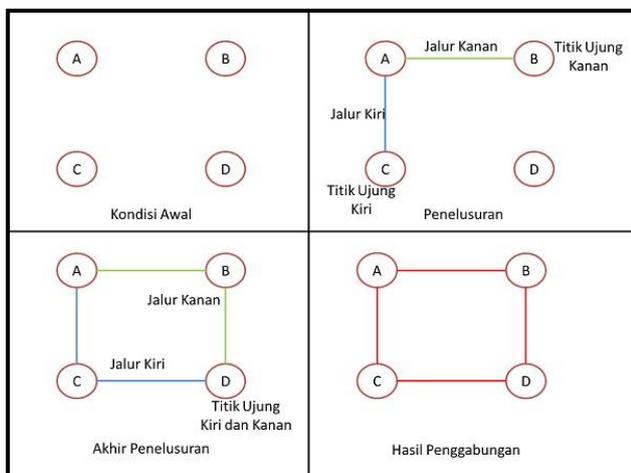
Dalam menyelesaikan permasalahan TSP, algoritma greedy dua arah dilakukan kedalam beberapa tahap bagian langkah kerja, yaitu :

1. Persiapan
Tahap persiapan dilakukan sebelum dilakukan penelusuran jalur, hal ini dilakukan untuk mempermudah dilakukannya proses penelusuran. Hal-hal yang dilakukan dalam tahap persiapan adalah :
 - a) Pembentukan matriks TSP
 - b) Pembentukan himpunan terurut berisi jarak dan titik tujuan dari setiap titik pada matriks
 - c) Menginisialisasi titik ujung kiri ,titik ujung kanan, himpunan jalur kiri, dan himpunan jalur kanan
2. Penelusuran

Tahap penelusuran dilakukan setelah seluruh proses persiapan selesai dilakukan. Pada tahap ini dilakukan penelusuran dari 2 buah titik, yaitu titik ujung kiri dan titik ujung kanan dengan mencari titik yang belum pernah dilalui kedua ujung titik dan memiliki nilai penjumlahan jarak dan jalur minimum selanjutnya yang paling minimum. Jika titik tujuan antara titik ujung kiri dan titik ujung kanan sama, maka akan pilih titik mana yang memiliki nilai hasil penjumlahan lebih kecil, dan titik yang lainnya akan mencari titik minimum yang lain. Jika titik tujuan kedua titik sama dan nilai hasil penjumlahannya pun sama, maka kedua titik ujung akan melakukan pencarian terhadap titik minimum lainnya, dan bila hasil perhitungan antar minimum kedua titik kanan dan minimum kedua titik kiri lebih kecil atau sama dengan, maka titik ujung kiri akan mengambil jalur pertama, dan titik ujung kanan akan mengambil jalur kedua. Sedangkan bila nilai perhitungan minimum kedua titik ujung kanan lebih besar dari minimum kedua titik ujung kiri, maka titik ujung kanan akan mengambil jalur pertama, dan titik ujung kiri akan mengambil jalur kedua. Setelah jalur terpilih, titik tujuan dari jalur yang terpilih akan dimasukkan kedalam himpunan jalur kiri dan himpunan jalur kanan.

3. Penggabungan

Proses penelusuran akan dilakukan sampai jumlah jalur yang terbentuk oleh titik ujung kiri dan titik ujung kanan berjumlah sama dengan jumlah titik. Setelah proses penelusuran selesai, maka akan dilakukan penggabungan antara himpunan jalur kiri dan himpunan jalur kanan, sehingga hingga membentuk sebuah jalur sirkuit pada himpunan hasil, untuk kemudian ditampilkan total jarak dan keseluruhan jalur yang ditempuh pada layar.



ALGORITMIK DALAM BAHASA C

```

/** Tipe bentukkan */
typedef struct Node
{
    int jarak;
    int nomor;
}Node;

/** Persiapan */
int** matrix = createMatrix;
titikKanan = 1;
titikKiri = 1;
JalurKiri[0] = 1;
JalurKanan[0] = 1;
JalurKiri[1] = cariMinimum(0);
JalurKanan[1] = cariMinimum(0);

/** Penelusuran */
while (i < matrix_size/2)
{
    if (cariMinimum(Kiri) ==
cariMinimum(Kanan))
    {
        if (cariMinimum(titikKiri).jarak <
cariMinimum(titikKanan).jarak)
        {
            // Mencari jalur kanan lain
        }
        else if
(cariMinimum(titikKiri).jarak >
cariMinimum(titikKanan).jarak)
        {
            // Mencari jalur kiri lain
        }
        else // tempJalurKiri.jarak ==
tempJalurKanan.jarak
        {
            // Mencari jalur kanan lain
            // Mencari jalur kiri lain
            // Bandingkan hasil
            if (cariNextMinimum(Kiri).jarak
<= cariNextMinimum(Kanan).jarak)
            {
                // Kanan -> jalur kedua,
Kiri -> jalur pertama
            }
            else //
cariNextMinimum(Kiri).jarak >
cariNextMinimum(Kanan).jarak
            {
                // Kanan -> jalur pertama,
Kiri -> Jalur kedua
            }
        }
    }

    JalurKiri[i] = tempJalurKiri;
    JalurKanan[i] = tempJalurKanan;
    titikKiri = JalurKiri[i];
    titikKanan = JalurKanan[i];
    i++;
}

/** Penggabungan */
int* Hasil = concat(JalurKiri,
Inverse(JalurKanan));
int totalJarak = totalJarak(JalurKiri) +
totalJarak(JalurKanan);

```

/* Tipe bentukkan */

IV. PERBANDINGAN OPTIMALITAS SOLUSI ANTARA ALGORITMA GREEDY DUA ARAH DAN PROGRAM DINAMIS

Berikut adalah hasil uji coba penyelesaian permasalahan TSP dengan algoritma greedy dua arah dan program dinamis:

Soal 1

8
99 1 1 1 1 1 1 1
1 99 2 2 2 2 2 2
1 2 99 3 3 3 3 3
1 2 3 99 4 4 4 4
1 2 3 4 99 5 5 5
1 2 3 4 5 99 6 6
1 2 3 4 5 6 99 7
1 2 3 4 5 6 7 99

Solusi dengan program dinamis :

Simpul penghasil jalur terpendek :
 1 ---> 5 ---> 2 ---> 4 ---> 6 ---> 3 ---> 7 ---> 8 ---> 1
 Bobot tur terpendek = 16

Solusi dengan greedy dua arah :

Jarak : 22
 Result : 1 4 2 8 5 7 3 6 1

Soal 2

5
99 1 2 3 4
1 99 6 7 8
2 6 99 11 12
3 7 11 99 16
4 8 12 16 99

Solusi dengan program dinamis :

Simpul penghasil jalur terpendek :
 1 ---> 5 ---> 2 ---> 3 ---> 4 ---> 1
 Bobot tur terpendek = 32

Solusi dengan greedy dua arah :

Jarak : 38
 Result : 1 2 4 5 3 1

Soal 3

12
99 1 2 3 4 5 6 7 8 9 1 2
1 99 2 3 4 5 6 7 8 9 1 2
2 2 99 3 4 5 6 7 8 9 1 2
3 3 3 99 4 5 6 7 8 9 1 2
4 4 4 4 99 5 6 7 8 9 1 2
5 5 5 5 5 99 6 7 8 9 1 2
6 6 6 6 6 6 99 7 8 9 1 2
7 7 7 7 7 7 7 99 8 9 1 2
8 8 8 8 8 8 8 8 99 9 1 2
9 9 9 9 9 9 9 9 9 99 1 2
1 1 1 1 1 1 1 1 1 1 99 2
2 2 2 2 2 2 2 2 2 2 2 99

Solusi dengan program dinamis

(Tidak berhasil ditemukan karena terlalu lama)

Solusi dengan greedy dua arah

Jarak : 52
 Result : 1 2 12 8 7 10 9 6 5 3 4 11 1

Soal 4

7
99 1 3 9 8 2 7
1 99 5 7 7 6 3
3 5 99 9 2 5 4
9 7 9 99 4 6 3
8 7 2 4 99 8 1
2 6 5 6 8 99 2
7 3 4 3 1 2 99

Solusi dengan program dinamis

Simpul penghasil jalur terpendek :
 1 ---> 5 ---> 7 ---> 4 ---> 2 ---> 3 ---> 6 ---> 1
 Bobot tur terpendek = 19

Solusi dengan greedy dua arah

Jarak : 24
 Result : 1 2 4 3 5 7 6 1

Soal	Jumlah titik	Jarak greedy	Jarak Program Dinamis
Soal 1	8	22	16
Soal 2	5	38	32
Soal 3	12	52	-
Soal 4	6	24	19

Tabel 1. Tabel perbandingan hasil uji coba penyelesaian masalah TSP dengan greedy dua arah dan program dinamis

Perbandingan waktu tidak dapat dilakukan karena program untuk program dinamis dan greedy dua arah dibuat dalam lingkungan yang berbeda

Berdasarkan hasil percobaan, diketahui bahwa

program dinamis menghasilkan hasil yang optimal sedangkan greedy dua arah belum menghasilkan hasil yang optimal. Hasil yang tidak optimal ini diakibatkan oleh penentuan hasil berdasarkan lokalitas per penelusuran pada greedy dua arah. Besar persentase penyimpangan hasil dengan menggunakan greedy dua arah diperkirakan sebesar 20-40%.

Walaupun perbandingan kompleksitas waktunya tidak diketahui, namun dari soal 3 dapat diketahui bahwa greedy dua arah memiliki kompleksitas waktu lebih kecil dibandingkan dengan algoritma program dinamis

V. KESIMPULAN DAN SARAN

1. Kesimpulan

Walaupun memiliki kompleksitas waktu yang lebih baik daripada program dinamis, greedy dua arah belum dapat menghasilkan solusi yang optimal untuk permasalahan TSP. Greedy dua arah memiliki besar persentase penyimpangan hasil sekitar 20-40% dari hasil optimal

2. Saran

Untuk melakukan perbandingan waktu antara dua algoritma, harus dilakukan pada lingkungan yang sama, spesifikasi komputer yang sama, dan keadaan komputer yang sama.

Untuk menyelesaikan suatu permasalahan algoritmik, diperlukan pemahaman teori algoritma yang baik dan juga penelusuran yang mendalam mengenai solusi-solusi permasalahan algoritmik.

REFERENSI

- [1] http://en.wikipedia.org/wiki/Travelling_salesman_problem, tanggal akses 8 Desember 2011
- [2] http://en.wikipedia.org/wiki/P_versus_NP_problem, tanggal akses 8 Desember 2011
- [3] http://en.wikipedia.org/wiki/Greedy_algorithm, tanggal akses 8 Desember 2011
- [4] <http://www.tsp.gatech.edu/>, tanggal akses 6 Desember 2011
- [5] Munir, Rinaldi. 2009. *Diktat Kuliah IF3051 Strategi Algoritma*. Bandung : Laboratorium Ilmu dan Rekayasa Komputasi Program Studi Informatika Sekolah Teknik Elektro dan Informatika ITB.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2010



(Samuel Cahyawijaya)
NIM : 13509082