

Penyelesaian Traveling Salesman Problem (TSP) dengan Dynamic Programming

Nama : Andrew Isra Saputra DB

NIM : 13523110

1. Pendahuluan

Traveling Salesman Problem (TSP) merupakan persoalan klasik dalam bidang optimasi kombinatorial dan algoritma. TSP bertujuan mencari rute terpendek yang mengunjungi setiap kota tepat satu kali dan kembali ke kota awal. Permasalahan ini bersifat NP-hard, sehingga solusi eksak dengan pendekatan brute-force menjadi sangat tidak efisien untuk jumlah kota yang besar.

2. Pemahaman Persoalan

Diberikan sebuah himpunan n kota dan jarak antar setiap pasangan kota, tugasnya adalah menentukan urutan kunjungan semua kota sehingga total jarak perjalanan minimum, dengan syarat:

1. Setiap kota harus dikunjungi satu kali saja.
2. Perjalanan harus berakhir kembali di kota awal.

Masalah ini memiliki ruang solusi sebesar $n!$ (faktorial dari jumlah kota), sehingga eksplorasi semua kemungkinan langsung tidak praktis.

3. Dynamic Programming untuk TSP

Dynamic Programming (DP) adalah teknik pemecahan masalah dengan memecahnya menjadi subproblem yang saling tumpang tindih dan menyimpan hasil subproblem agar tidak dihitung ulang (memoization). Untuk TSP, DP menggunakan pendekatan bitmasking untuk menyimpan status kota yang sudah dikunjungi.

4. Algoritma Penyelesaian

Implementasi algoritma TSP dengan DP sebagai berikut:

1. Mulai dari kota awal dengan mask yang hanya menandai kota awal sudah dikunjungi.
2. Pada setiap langkah, coba semua kota yang belum dikunjungi.
3. Hitung biaya total dengan rekursif menuju kota berikutnya, simpan hasil di memo agar tidak dihitung ulang.
4. Jika semua kota sudah dikunjungi, kembalikan biaya kembali ke kota awal.
5. Pilih jalur dengan biaya total minimum.

5. Uji Algoritma dengan Ruby

5.1. Test Case 1 dengan graf tidak berarah

```
===== Test case 1 (3 kota) =====  
Matriks jarak:  
  0   5   9  
  5   0   2  
  9   2   0  
Biaya perjalanan terpendek TSP adalah: 16  
Waktu eksekusi: 0.33 ms
```

5.2. Test Case 2 dengan graf berarah

```
===== Test case 2 (4 kota) =====  
Matriks jarak:  
  0   3   1   5  
  2   0   4   2  
  3   6   0   7  
  5   2   3   0  
Biaya perjalanan terpendek TSP adalah: 11  
Waktu eksekusi: 0.17 ms
```

5.3. Test Case 3 dengan 5 kota yang memiliki jarak yang cukup jauh

```
===== Test case 3 (5 kota, ada jarak jauh) =====  
Matriks jarak:  
  0  10  10  10 100  
 10   0  20  20 100  
 10  20   0  15 100  
 10  20  15   0 100  
100 100 100 100   0  
Biaya perjalanan terpendek TSP adalah: 235  
Waktu eksekusi: 0.21 ms
```

5.4. Test Case 4 dengan 1 kota

```
===== Test case 4 (1 kota) =====  
Matriks jarak:  
  0  
Biaya perjalanan terpendek TSP adalah: 0  
Waktu eksekusi: 0.11 ms
```

5.5. Test Case 5 dengan graf yang sangat besar (20 kota)

```
===== Test case 5 (20 kota, matriks besar acak) =====
Matriks jarak:
 0  84  21  90  66  79  36  14  66  89  99  4  13  87  34  91  8  85  59  65
28  0  59  74  3  71  94  38  73  55  57  51  76  2  37  32  49  58  7  22
7  71  0  4  71  50  28  31  57  15  27  66  53  36  16  14  73  47  15  72
4  83  12  0  15  17  9  28  57  80  47  9  58  14  45  75  35  56  31  58
89 28  55  43  0  80  27  86  82  59  46  84  8  24  62  16  72  17  75  98
94 5  34  23  10  0  80  23  9  15  87  35  34  66  98  72  9  79  12  59
4  98  27  4  74  96  0  58  94  28  47  33  52  55  50  27  57  27  36  79
87 47  2  35  17  16  59  0  53  81  22  42  80  29  4  25  5  69  77  80
41 18  96  46  22  68  1  15  0  56  81  72  12  98  33  41  92  93  73  31
96 66  66  93  10  16  23  4  57  0  40  46  81  57  32  71  74  78  19  25
23 20  95  88  57  46  37  90  29  97  0  23  47  25  39  75  92  19  44  66
100 71  38  11  57  48  66  27  81  15  4  0  1  14  15  87  81  23  71  3
59 17  10  53  72  35  3  85  40  69  36  47  0  89  100  67  57  34  91  50
52 92  66  6  35  51  58  41  16  24  66  33  64  0  100  38  82  18  56  80
10 9  10  95  17  18  53  62  30  60  56  51  77  65  0  100  26  95  39  53
11 44  94  57  67  84  60  91  10  56  8  82  84  50  98  0  71  35  65  2
46 48  14  11  59  68  98  51  35  47  54  93  33  46  14  63  0  70  57  34
91 76  34  1  47  83  32  67  20  16  40  67  87  98  83  2  67  0  49  53
23 54  58  89  18  15  57  41  4  23  74  25  57  67  95  94  59  69  0  96
73 58  50  3  29  72  27  34  45  7  44  27  98  50  22  97  52  59  14  0

Biaya perjalanan terpendek TSP adalah: 147
Waktu eksekusi: 93471.53 ms
```

Dapat dilihat waktu yang dihasilkan sangat eksponensial.

6. Kompleksitas

Algoritma DP ini memiliki kompleksitas waktu dan ruang sebesar $O(n^2 * 2^n)$, jauh lebih baik daripada brute-force $O(n!)$, tapi masih terbatas untuk jumlah kota kecil hingga sedang.

7. Kesimpulan

Pendekatan Dynamic Programming dengan bitmasking efektif untuk menyelesaikan TSP pada jumlah kota kecil hingga sedang dengan waktu yang relatif efisien dibandingkan brute-force. Namun, kompleksitas eksponensial tetap membatasi penggunaannya untuk graf dengan jumlah kota besar. Untuk graf besar, solusi heuristik atau aproksimasi lebih disarankan.