

PENYELESAIAN TRAVELING SALESMAN PROBLEM DENGAN DYNAMIC PROGRAMMING

Laporan Tugas Kecil Tambahan

Disusun untuk memenuhi mata kuliah IF2211 Strategi Algoritma



oleh:

Nayla Zahira 13523079

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2025**

DAFTAR ISI

DAFTAR ISI	1
A. DESKRIPSI TUGAS	2
B. IMPLEMENTASI PROGRAM	2
1. Struktur Kode Program	2
2. Penjelasan Kode Program	2
C. HASIL OUTPUT PROGRAM	3

A. DESKRIPSI TUGAS

Program ini merupakan implementasi solusi Traveling Salesman Problem (TSP) menggunakan pendekatan Dynamic Programming dengan teknik bitmasking. TSP adalah masalah optimisasi kombinatorial yang mencari rute terpendek untuk mengunjungi semua kota tepat sekali dan kembali ke kota asal. Program ini ditulis dalam bahasa Ruby dan mampu menyelesaikan TSP dengan solusi optimal

B. IMPLEMENTASI PROGRAM

1. Struktur Kode Program

Folder program disusun sebagai berikut:

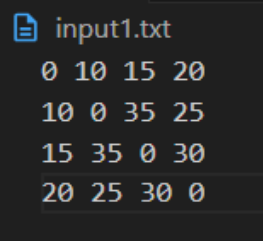
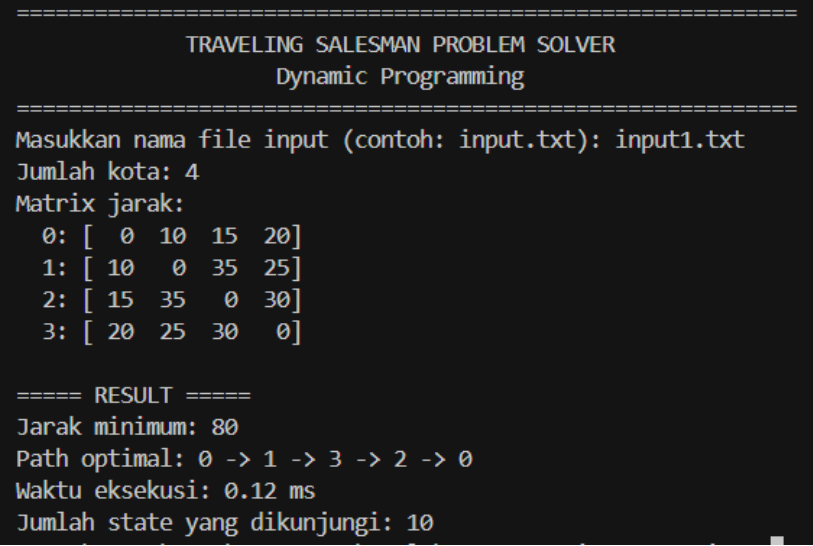
```
IF2211_TSP_DP_13523079
├── doc/
│   └── laporan.pdf      # Output Program dalam format PDF
├── src/                 # Folder source code utama
│   └── main.rb          # Main program
├── test/                # Folder untuk file input
│   ├── input1.txt
│   ├── input2.txt
│   └── input3.txt
├── run.bat              # Batch untuk run program
└── README.md            # Dokumentasi program
```

2. Penjelasan Kode Program

Kelas TSPSolver	
Metode	Penjelasan
initialize(distance_matrix)	Menginisialisasi solver dengan matrix jarak, menghitung jumlah kota, membuat hash untuk memoization, dan memvalidasi input
solve	Menampilkan informasi matrix, menjalankan algoritma TSP, dan mencetak hasil beserta statistik waktu eksekusi
calc	Menghitung jarak minimum dan konstruksi path optimal dengan memanggil fungsi rekursif dan path reconstruction
validate_input	Memvalidasi bahwa matrix input adalah array 2D persegi dengan diagonal bernilai 0

tsp(mask, pos)	Fungsi rekursif utama yang menggunakan dynamic programming dengan bitmask untuk menghitung jarak minimum TSP
construct_path	Merekonstruksi path optimal berdasarkan hasil memoization dari fungsi tsp
display_matrix	Menampilkan matrix jarak dalam format yang terstruktur
read_matrix_from_file(filename)	Membaca file input, memvalidasi format, dan mengkonversi menjadi matrix integer 2D
main	Fungsi utama yang menangani input user, membaca file, dan menjalankan solver TSP

C. HASIL OUTPUT PROGRAM

Test Case	Output Program
 <pre> input1.txt 0 10 15 20 10 0 35 25 15 35 0 30 20 25 30 0 </pre>	 <pre> ===== TRAVELING SALESMAN PROBLEM SOLVER Dynamic Programming ===== Masukkan nama file input (contoh: input.txt): input1.txt Jumlah kota: 4 Matrix jarak: 0: [0 10 15 20] 1: [10 0 35 25] 2: [15 35 0 30] 3: [20 25 30 0] ===== RESULT ===== Jarak minimum: 80 Path optimal: 0 -> 1 -> 3 -> 2 -> 0 Waktu eksekusi: 0.12 ms Jumlah state yang dikunjungi: 10 </pre>

<pre> 0 10 15 20 5 0 9 10 6 13 0 12 8 8 9 0 </pre>	<pre> ===== TRAVELING SALESMAN PROBLEM SOLVER Dynamic Programming ===== Masukkan nama file input (contoh: input.txt): input2.txt Jumlah kota: 4 Matrix jarak: 0: [0 10 15 20] 1: [5 0 9 10] 2: [6 13 0 12] 3: [8 8 9 0] ===== RESULT ===== Jarak minimum: 35 Path optimal: 0 -> 1 -> 3 -> 2 -> 0 Waktu eksekusi: 0.16 ms Jumlah state yang dikunjungi: 10 </pre>
<pre> 0 29 20 21 29 0 15 17 20 15 0 28 21 17 28 0 </pre>	<pre> ===== TRAVELING SALESMAN PROBLEM SOLVER Dynamic Programming ===== Masukkan nama file input (contoh: input.txt): input3.txt Jumlah kota: 4 Matrix jarak: 0: [0 29 20 21] 1: [29 0 15 17] 2: [20 15 0 28] 3: [21 17 28 0] ===== RESULT ===== Jarak minimum: 73 Path optimal: 0 -> 2 -> 1 -> 3 -> 0 Waktu eksekusi: 0.07 ms Jumlah state yang dikunjungi: 10 </pre>

Tautan repository Github

https://github.com/naylzhra/IF2211_TSP_DP_13523079