

# Praktikum Kecerdasan Buatan



Nama : Andyka Salom

Nim : 434221054

TI -B4

D4 TEKNIK INFORMATIKA

Fakultas VOKASI

Universitas Airlangga

### 1.Import Library dan Modules:

```
Python_repo > 434221054_prakt3.py
1 import itertools
2
```

Module `itertools` digunakan untuk menghasilkan semua kemungkinan permutasi dari sekumpulan data.

### 2. Fungsi calculate\_path\_length :

```
3 def calculate_path_length(path, graph):
4     length = 0
5     for i in range(len(path) - 1):
6         length += graph[path[i]][path[i+1]]
7     length += graph[path[-1]][path[0]]
8     return length
```

Fungsi ini menghitung panjang jalur dari satu titik ke titik lainnya dalam jalur yang diberikan, berdasarkan bobot yang diberikan dalam bentuk graf.

### 3. Fungsi `traveling\_salesman` :

```
9 def traveling_salesman(num_cities, graph):
10     cities = [i for i in range(num_cities)]
11     shortest_path = None
12     shortest_length = float('inf')
13
14     for perm in itertools.permutations(cities):
15         length = calculate_path_length(perm, graph)
16         if length < shortest_length:
17             shortest_length = length
18             shortest_path = perm
19
20     return shortest_path, shortest_length
```

Fungsi ini menggunakan metode brute force untuk menemukan jalur terpendek yang mengunjungi setiap kota tepat satu kali dan kembali ke kota awal.

### 4. Fungsi main :

```
def main():
    num_cities = int(input("Masukkan jumlah kota: "))
    graph = [[] for _ in range(num_cities)]

    # Input bobot/jarak antar kota
    for i in range(num_cities):
        for j in range(num_cities):
            if i == j:
                graph[i].append(0)
            else:
                weight = float(input(f"Masukkan bobot antara kota {i} dan {j}: "))
                graph[i].append(weight)

    start_city = int(input("Masukkan kota asal (dalam range 0 sampai {num_cities - 1}): "))
    end_city = int(input("Masukkan kota tujuan (dalam range 0 sampai {num_cities - 1}): "))

    shortest_path, shortest_length = traveling_salesman(num_cities, graph)
    print(f"Jalur terpendek: {shortest_path}")
    print(f"Panjang jalur terpendek: {shortest_length}")

if __name__ == "__main__":
    main()
```

Fungsi `main` berisi logika utama program. Program meminta pengguna untuk memasukkan jumlah kota, bobot antara kota, serta kota asal dan tujuan. Kemudian, fungsi `traveling\_salesman` dipanggil untuk mencari jalur terpendek. Hasilnya kemudian dicetak ke layar.

##### 5. Pemanggilan Fungsi `main` :

```
if __name__ == "__main__":
    main()
```

Ini adalah bagian dari kode yang akan menjalankan fungsi `main` ketika program dieksekusi. Itu memungkinkan program dapat diimpor sebagai modul ke dalam skrip lain tanpa menjalankan `main()` secara otomatis.