

LAPORAN STUDI KASUS
MATEMATIKA DISKRIT
RUTE PENGIRIMAN PAKET



Oleh: DEDY APRIYANTO

Nama : Dedy Apriyanto
NIM : 2400018103
Kelas : B
Slot : Selasa 10.00

LABORATORIUM KOMPUTER INFORMATIKA
PROGRAM STUDI INFORMATIKA FAKULTAS
TEKNOLOGI INDUSTRI UNIVERSITAS
AHMAD DAHLAN YOGYAKARTA
2025

A. DESKRIPSI STUDI KASUS

Masalah :

Seorang kurir harus mengirimkan paket dari Gudang paket ke beberapa tempat dengan jarak terpendek masalah ini bisa di modelkan seperti graf

Dimana:

- Simpul : kota – kota
- Sisi : jalan yang menghubungkan 2 kota
- Bobot Sisi : jarak antara 2 kota

Tujuan :

Program ini untuk menentukan rute terpendek dari kota asal ke kota tujuan dengan menggunakan materi matematika diskrit dan program ini di rancang tanpa menggunakan struktur data yang terlalu kompleks

B. KESULITAN MASALAH

Jaringan yang terlalu kompleks :

- Terlalu banyak jalan terpendek yang harus di pilih untuk menentukan jalur terpendek
- Terlalu banyak jumlah kota dan jalan yang menyebabkan perhitungan menjadi tidak efisien

Optimasi waktu dan biaya :

- Terjadi kesalahan memilih rute yang menyebabkan pengantaran paket menjadi lama dan menghabiskan bahan bakar

Faktor jalan :

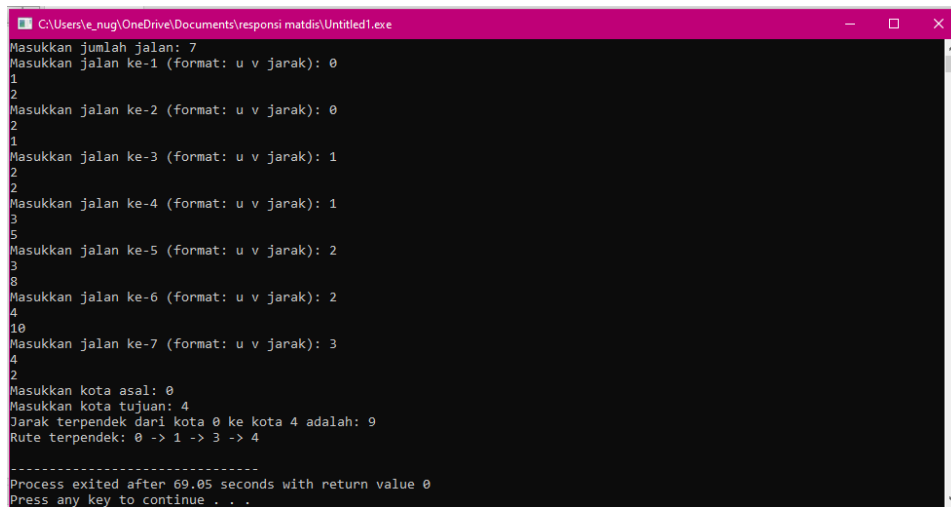
- Dibalik jarak nya dari program ini juga harus bisa menentukan jalan dari gudang sampai ke tujuan paket

C. MANFAAT

- Lebih menghemat biaya perjalanan seperti bensin dan waktu pengerjaan
- Pelayanan pengiriman menjadi lebih cepat dan bisa memberikan pelayanan yang bagus ke pelanggan
- Program ini juga bisa membantu orang untuk mempelajari konsep dari graf untuk menentukan jarak tempuh dari kota ke kota

D. HASIL DAN PEMBAHASAN

A. Hasil program :



```
C:\Users\ve_nug\OneDrive\Documents\responsi matdis\Untitled1.exe
Masukkan jumlah jalan: 7
Masukkan jalan ke-1 (format: u v jarak): 0
1
2
Masukkan jalan ke-2 (format: u v jarak): 0
2
1
Masukkan jalan ke-3 (format: u v jarak): 1
2
2
Masukkan jalan ke-4 (format: u v jarak): 1
3
5
Masukkan jalan ke-5 (format: u v jarak): 2
3
8
Masukkan jalan ke-6 (format: u v jarak): 2
4
10
Masukkan jalan ke-7 (format: u v jarak): 3
4
2
Masukkan kota asal: 0
Masukkan kota tujuan: 4
Jarak terpendek dari kota 0 ke kota 4 adalah: 9
Rute terpendek: 0 -> 1 -> 3 -> 4
-----
Process exited after 69.05 seconds with return value 0
Press any key to continue . . .
```

Ini adalah hasil input dan output yang menentukan jarak antara kota ke kota dan bisa memberikan jarak dari kota a ke kota b serta bisa memberikan jalur terpendek untuk menempuh jalur tersebut

B. Kelebihan program :

- Program yang lebih sederhana karena tidak menggunakan struktur yang kompleks
- Program ini juga mudah untuk dipahami untuk kita yang belajar tentang algoritma graf dari dasar maka dari itu program ini cocok untuk belajar karena mudah dipahami

C. minDistance fungsi ini untuk menentukan simpul dari jarak kota ke kota supaya bisa menentukan jarak terkecil yang belum dikunjungi contoh program :

```
7 // Fungsi untuk mencari simpul dengan jarak terkecil
8 int minDistance(int dist[], bool visited[], int n) {
9     int min = INT_MAX, index = -1;
10    for (int i = 0; i < n; i++) {
11        if (!visited[i] && dist[i] <= min) {
12            min = dist[i];
13            index = i;
14        }
15    }
16    return index;
17 }
```

- D. Program ini untuk menentukan jalur antar kota ke kota dan menunjukkan tentang fungsi program yang berfungsi menunjukkan percabangan 2 jalur yang berbeda

Contoh program :

```
78 // Input jalur antar kota
79 for (int i = 0; i < m; i++) {
80     int u, v, w;
81     cout << "Masukkan jalan ke-" << i+1 << " (format: u v jarak): ";
82     cin >> u >> v >> w;
83     graph[u][v] = w;
84     graph[v][u] = w; // Karena jalan dua arah
85 }
86
87
88 int source, dest;
89 cout << "Masukkan kota asal: ";
90 cin >> source;
91 cout << "Masukkan kota tujuan: ";
92 cin >> dest;
93
94 dijkstra(graph, n, source, dest);
95
96 return 0;
97 }
```

- E. Dari gambar ini menunjukan program yang digunakan untuk menentukan tampilan hasil, rute jalur dan program yang menentukan tampilan jalur dari awal ke akhir

```
// Tampilkan hasil
cout << "Jarak terpendek dari kota " << source << " ke kota " << dest << " adalah: " << dist[dest] << endl;

// Rekonstruksi jalur
cout << "Rute terpendek: ";
int path[MAX_CITIES], pathIndex = 0;
int current = dest;
while (current != -1) {
    path[pathIndex++] = current;
    current = prev[current];
}

// Tampilkan jalur dari akhir ke awal
for (int i = pathIndex - 1; i >= 0; i--) {
    cout << path[i];
    if (i != 0) cout << " -> ";
}
cout << endl;
```