

Laporan Struktur Data

UAS



Disusun Oleh :

Nama : Alvin Febrianto

NIM : 21091397031

Kelas : A

Fakultas Vokasi

D4 Manajemen Informatika

Universitas Negeri Surabaya

2022

- Nomor 1

```

1  #include <iostream>
2  #include <conio.h>
3  #include <bits/stdc++.h>
4  using namespace std;
5
6  // Fungsi void addEdge untuk menambahkan edge
7  void addEdge(vector < pair < int, int > > adj[], int u, int v, int wt) {
8      adj[u].push_back(make_pair(v, wt));
9      adj[v].push_back(make_pair(u, wt));
10 }
11
12 // Fungsi void printGraph untuk mengubah graph menjadi adjacency list
13 void printGraph(vector < pair < int, int > > adj[], int V) {
14     int v, w;
15     for (int u = 1; u < V; u++) {
16         cout << u;
17         for (auto it = adj[u].begin(); it!=adj[u].end(); it++) {
18             v = it -> first;
19             w = it -> second;
20             cout << " -> " << "[" << v << ", " << w << "]";
21         }
22         cout << "\n";
23     }
24 }
25
26 int main() {
27     int V = 5;
28     vector < pair < int, int > > adj[V];
29     // Menambahkan nilai edge
30     addEdge(adj, 1, 2, 5);
31     addEdge(adj, 2, 3, 1);
32     addEdge(adj, 4, 1, 3);
33     addEdge(adj, 2, 4, 1);
34     addEdge(adj, 3, 1, 1);
35
36     // Mencetak graph
37     printGraph(adj, V);
38     getch();
39 }

```

```

1 -> [2,5] -> [4,3] -> [3,1]
2 -> [1,5] -> [3,1] -> [4,1]
3 -> [2,1] -> [1,1]
4 -> [1,3] -> [2,1]

```

- Nomor 2

```

1  #include <iostream>
2  #include <conio.h>
3  #include <string.h>
4  using namespace std;
5
6  int main() {
7      char kota_1[10], kota_2[10], kota_3[10], kota_4[10], kota_5[10];
8      int jumlah, panjang, hasil_1, hasil_2, hasil_3, hasil_4, hasil_5, hasil_6, hasil_7;
9
10     // Input jumlah kota
11     cout << "Jumlah kota dalam Kerajaan Britan: " << endl;
12     cin >> jumlah; cout << endl;
13
14     // Input nama kota
15     cout << "- Kota Pertama : "; cin >> kota_1;
16     cout << "- Kota Kedua : "; cin >> kota_2;
17     cout << "- Kota Ketiga : "; cin >> kota_3;
18     cout << "- Kota Keempat : "; cin >> kota_4;
19     cout << "- Kota Kelima : "; cin >> kota_5;
20     cout << endl;
21
22     // Menampilkan graph yang terjadi
23     cout << "Sisi-sisinya adalah: " << endl;
24     cout << "(" << kota_1 << ", " << kota_2 << "), ";
25     cout << "(" << kota_1 << ", " << kota_4 << "), ";
26     cout << "(" << kota_1 << ", " << kota_5 << "), ";
27     cout << "(" << kota_2 << ", " << kota_3 << "), ";
28     cout << "(" << kota_3 << ", " << kota_5 << "), ";
29     cout << "(" << kota_3 << ", " << kota_4 << "), ";
30     cout << "(" << kota_4 << ", " << kota_5 << ")\n\n";
31
32     // Menampilkan panjang jalan yang menghubungkan vertex
33     cout << "Panjang jalan antarkota:" << endl;
34     cout << "- Panjang " << kota_1 << " ke " << kota_2 << " : "; cin >> hasil_1;
35     cout << "- Panjang " << kota_1 << " ke " << kota_4 << " : "; cin >> hasil_2;
36     cout << "- Panjang " << kota_1 << " ke " << kota_5 << " : "; cin >> hasil_3;
37     cout << "- Panjang " << kota_2 << " ke " << kota_3 << " : "; cin >> hasil_4;
38     cout << "- Panjang " << kota_3 << " ke " << kota_5 << " : "; cin >> hasil_5;
39     cout << "- Panjang " << kota_3 << " ke " << kota_4 << " : "; cin >> hasil_6;
40     cout << "- Panjang " << kota_4 << " ke " << kota_5 << " : "; cin >> hasil_7;
41     cout << endl;
42
43     // Menampilkan jalan yang menghubungkan kedua simpul (x,y,z)
44     cout << "Seluruh jalan yang ada dalam Kerajaan Britan dan panjang jalannya: " << endl;
45     cout << "(" << kota_1 << ", " << kota_2 << ", " << hasil_1 << ") ";
46     cout << "(" << kota_1 << ", " << kota_4 << ", " << hasil_2 << ") ";
47     cout << "(" << kota_1 << ", " << kota_5 << ", " << hasil_3 << ") ";
48     cout << "(" << kota_2 << ", " << kota_3 << ", " << hasil_4 << ") ";
49     cout << "(" << kota_3 << ", " << kota_5 << ", " << hasil_5 << ") ";
50     cout << "(" << kota_3 << ", " << kota_4 << ", " << hasil_6 << ") ";
51     cout << "(" << kota_4 << ", " << kota_5 << ", " << hasil_7 << ") ";
52     cout << "\n\n";
53
54     // Menampilkan tempat pedagang berada
55     cout << "Kota tempat pedagang sekarang berada: \n";
56     cout << kota_1 << "\n\n";
57
58     // Menampilkan kota yang diserang naga
59     cout << "Kota yang diserang naga: \n";
60     cout << kota_3 << "\n\n";
61
62     // Menampilkan kota yang memiliki kastil
63     cout << "Kota yang memiliki kastil: \n";
64     cout << kota_5 << "\n\n";
65
66     // Menampilkan vertex tercepat untuk sampai ke tujuan
67     cout << "Jalur yang paling cepat ditempuh tanpa melewati kota yang diserang naga: \n";
68     cout << kota_1 << "- " << kota_4 << "- " << kota_5 << "\n\n";
69
70     // Total edge yang harus ditempuh
71     cout << "Dengan jarak: \n";
72     cout << hasil_2 + hasil_7 << "\n\n";
73
74     getch();
75 }

```

Jumlah kota dalam Kerajaan Britan:

5

- Kota Pertama : 1
- Kota Kedua : 2
- Kota Ketiga : 3
- Kota Keempat : 4
- Kota Kelima : 5

Sisi-sisinya adalah:

(1, 2), (1, 4), (1, 5), (2, 3), (3, 5), (3, 4), (4, 5)

Panjang jalan antarkota:

- Panjang 1 ke 2 : 12
- Panjang 1 ke 4 : 11
- Panjang 1 ke 5 : 30
- Panjang 2 ke 3 : 14
- Panjang 3 ke 5 : 5
- Panjang 3 ke 4 : 15
- Panjang 4 ke 5 : 10

Seluruh jalan yang ada dalam Kerajaan Britan dan panjang jalannya:

(1, 2, 12) (1, 4, 11) (1, 5, 30) (2, 3, 14) (3, 5, 5) (3, 4, 15) (4, 5, 10)

Kota tempat pedagang sekarang berada:

1

Kota yang diserang naga:

3

Kota yang memiliki kastil:

5

Jalur yang paling cepat ditempuh tanpa melewati kota yang diserang naga:

1-4-5

Dengan jarak:

21