

**LAPORAN STRUKTUR DATA**

**UJIAN AKHIR SEMESTER**



**Disusun oleh :**

**Salsabilla Octavianingrum**

**21091397005**

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2022**

1. Membuat kodingan dan laporan sebuah algoritma yang membuat undirected graph menggunakan representasi adjacency list dengan input vertex dan edge. **Input :**

- 1) int jumlah vertex yang ada dalam graph
- 2) 2) (x,y,w) dengan x = vertex 1, y = vertex 2, w = weight

**Output :**

Satu per satu vertex, edge, dan weightnya

### Coding

```
Start here x 1. Graph.cpp x
1 //undirected graph menggunakan representasi adjacency list dengan input vertex dan edge
2 #include <bits/stdc++.h>
3 #include <iostream>
4 using namespace std;
5
6 //function 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 //function untuk merubah 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;
```

```
Start here x 1. Graph.cpp x
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     //cetak graph
37     printGraph(adj, V);
38     return 0;
39 }
40
```

## Output

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

Process returned 0 (0x0)   execution time : 0.072 s
Press any key to continue.
```

2. Terdapat seorang pedagang Rahmad, Rahmad setiap bulan berkeliling di Kerajaan Britan untuk berdagang. Tetapi suatu hari, pedagang ini mendapat berita bahwa ada seekor naga yang sedang menyerang salah satu kota. Jadi pedagang ini bergegas menuju ke istana untuk memberitahu raja bahwa ada kota yang sedang diserang sambil menghindari kota tersebut. Sehingga raja bisa mengirimkan pasukan untuk menyerang kota tersebut.

Buat kodingan dan laporan cara kerja kodingan tersebut. Jelaskan menggunakan algoritma apa kodingan anda berjalan (dijkstra, A\*, bellman ford, dll) dan jelaskan cara kerjanya. Peta kota adalah sebuah undirected, weighted graph. Boleh menggunakan adjacency list atau menggunakan adjacency matrix.

hint: hapus vertex naga dan semua edge yang berhubungan dengan nya

### Input :

- 1) int jumlah vertex yang ada dalam graph
- 2) (x,y,w) dipisahkan dengan spasi  
x = vertex 1, y = vertex 2, w = weight
- 3) Kota mana yang merupakan kota yang ditempati pedagang sekarang
- 4) Vertex mana yang merupakan kota yang diserang naga
- 5) Vertex mana yang merupakan kota tempat istana raja

### Output :

Satu per satu vertex, edge, dan weightnya

## Coding

```
1  #include <iostream>
2  #include <conio.h>
3  #include <string.h>
4  using namespace std;
5  int main()
6  {
7      char kotal[10], kota2[10], kota3[10], kota4[10], kota5[10];
8      int jumlah, panjang, hasil1, hasil2, hasil3, hasil4, hasil5, hasil6, hasil7;
9
10     //input jumlah kota
11     cout<<"Jumlah kota di Kerajaan Britan : ";
12     cin>>jumlah;
13     cout<<endl;
14
15     //input nama kota
16     cout<<"Kota Pertama : ";
17     cin>>kotal;
18     cout<<"Kota Kedua : ";
19     cin>>kota2;
20     cout<<"Kota Ketiga : ";
21     cin>>kota3;
22     cout<<"Kota Keempat : ";
23     cin>>kota4;
24     cout<<"Kota Kelima : ";
25     cin>>kota5;
26
27     cout<<endl;
28     cin>>kota5;
29
30     cout<<endl;
31
32     //deklarasi graph
33     //menampilkan graph yang terjadi
34     cout<<"Sisi-sisinya : ";
35     cout<<kotal<<kota2<<" ";
36     cout<<kotal<<kota4<<" ";
37     cout<<kotal<<kota5<<" ";
38     cout<<kota2<<kota3<<" ";
39     cout<<kota3<<kota5<<" ";
40     cout<<kota3<<kota4<<" ";
41     cout<<kota4<<kota5<<endl<<endl;
42
43     //deklarasi edge
44     //menampilkan panjang jalan yang menghubungkan vertex
45     cout<<"Panjang jalan antar kota"<<endl;
46     cout<<"Panjang " <<kotal<<" ke " <<kota2<<" : "; cin>> hasil1;
47     cout<<"Panjang " <<kotal<<" ke " <<kota4<<" : "; cin>> hasil2;
48     cout<<"Panjang " <<kotal<<" ke " <<kota5<<" : "; cin>> hasil3;
49     cout<<"Panjang " <<kota2<<" ke " <<kota3<<" : "; cin>> hasil4;
50     cout<<"Panjang " <<kota3<<" ke " <<kota5<<" : "; cin>> hasil5;
51     cout<<"Panjang " <<kota3<<" ke " <<kota4<<" : "; cin>> hasil6;
52     cout<<"Panjang " <<kota4<<" ke " <<kota5<<" : "; cin>> hasil7;
53
54     cout<<endl;
```

```

49     cout<<"Panjang "<<kota4<<" ke "<<kota5<<" : "; cin>> hasil7;
50
51     cout<<endl;
52
53     //deklarasi adjacent
54     //menampilkan jalan yang menghubungkan kedua simpul (x,y,z)
55     cout<<"Seluruh jalan yang ada di Kerajaan Britan dan panjang jalannya : ";
56     cout<<"("<<kota1<<","<<kota2<<","<<hasil1<<") ";
57     cout<<"("<<kota1<<","<<kota4<<","<<hasil2<<") ";
58     cout<<"("<<kota1<<","<<kota5<<","<<hasil3<<") ";
59     cout<<"("<<kota2<<","<<kota3<<","<<hasil4<<") ";
60     cout<<"("<<kota3<<","<<kota5<<","<<hasil5<<") ";
61     cout<<"("<<kota3<<","<<kota4<<","<<hasil6<<") ";
62     cout<<"("<<kota4<<","<<kota5<<","<<hasil7<<") ";
63
64     cout<<endl;
65
66     //hasil
67     //menampilkan tempat pedagang berada
68     cout<<"Kota tempat pedagang sekarang : ";
69     cout<<kota1;
70
71     cout<<endl<<endl;
72
73     //menampilkan kota yang diserang naga
74     cout<<"Kota yang diserang naga : ";
75     cout<<kota3;
76
77     cout<<endl<<endl;
78
79     //menampilkan kota yang terdapat kastil
80     cout<<"Kota yang memiliki kastil : ";
81     cout<<kota5;
82
83     cout<<endl<<endl;
84
85     //menampilkan vertex tercepat untuk selamat
86     cout<<"Jalur yang paling cepat ditempuh : ";
87     cout<<kota1<<"- "<<kota4<<"- "<<kota5;
88
89     cout<<endl<<endl;
90
91     //total edge yang harus ditempuh
92     cout<<"Dengan jarak yang ditempuh : ";
93     cout<<hasil2+hasil7<<endl<<endl;
94
95
96     getch();
97     return 0;
98 }

```

## Output

```

Kota Pertama : e
Kota Kedua : d
Kota Ketiga : c
Kota Keempat : b
Kota Kelima : a

Sisi-sisinya : ed,eb,ea,dc,ca,cb,ba

Panjang jalan antar kota
Panjang e ke d: 5
Panjang e ke b: 4
Panjang e ke a: 3
Panjang d ke c: 2
Panjang c ke a: 1
Panjang c ke b: 3
Panjang b ke a: 4

Seluruh jalan yang ada di Kerajaan Britan dan panjang jalannya : (e,d,5) (e,b,4) (e,a,3) (d,c,2) (c,a,1) (c,b,3) (b,a,4)

Kota tempat pedagang sekarang : e

Kota yang diserang naga : c

Kota yang memiliki kastil : a

Jalur yang paling cepat ditempuh : e-b-a

Dengan jarak yang ditempuh : 8

```