

# **LAPORAN UAS STRUKTUR DATA**

Disusun untuk memenuhi UAS mata kuliah

Struktur Data



**Disusun oleh :**

Kania Meliana Fityanti (21091397028)

**PROGRAM VOKASI**

**PROGRAM STUDI D IV MANAJEMEN INFORMATIKA**

**UNIVERSITAS NEGERI SURABAYA**

**TAHUN 2022**

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

Input :

0. int jumlah vertex yang ada dalam graph

1. (x,y,) dengan x = vertex 1, y = vertex 2, w = weight

Output :

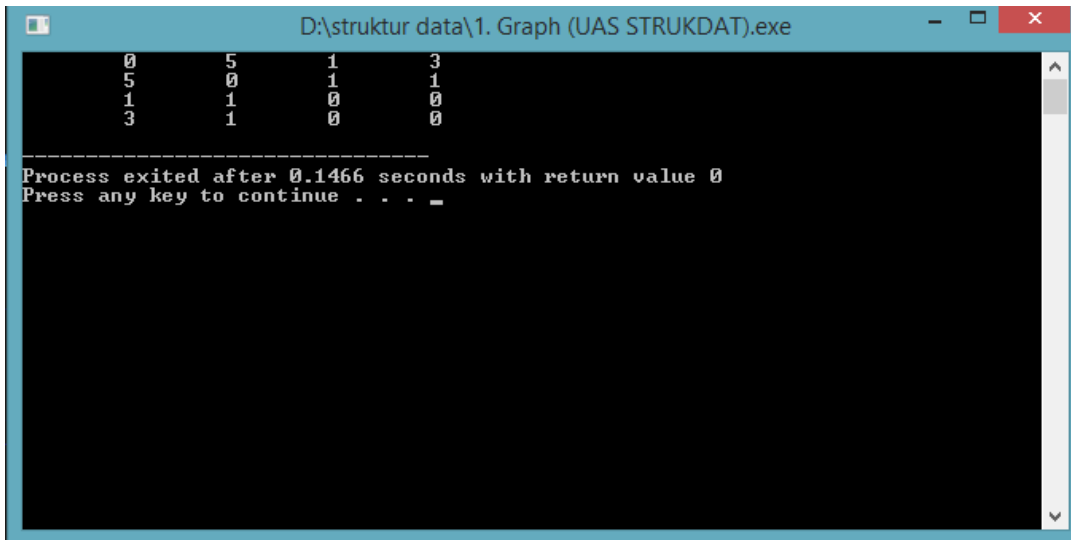
Satu per satu vertex, edge, dan weightnya

## CODINGAN

```
[*] 1. Graph (UAS STRUKDAT).cpp ×
1  #include<iostream>
2  #define Max 100
3
4  using namespace std;
5
6  int adjMat[Max][Max];
7
8  // instalasi matriks ke nol
9  void initializeMat(int v)
10 {
11     for(int i = 0; i < v; i++)
12     {
13         for(int j = 0; j < v; j++)
14         {
15             adjMat[i][j] = 0;
16         }
17     }
18 }
19
20 // function untuk menambahkan edges
21 void addEdge(int u, int v, int w)
22 {
23     adjMat[u][v] = w;
24     adjMat[v][u] = w;
25 }
26
27 // mencetak matriks
28 void displayMat(int v)
29 {
30     for (int i = 1; i <= v; i++)
31     {
32         cout << "\t";
33         for (int j = 1; j <= v; j++)
34         {
35             adjMat[i][j] << "\t";
36         }
37     }
38 }
```

```
[*] 1. Graph (UAS STRUKDAT).cpp ×
28 void displayMat(int v)
29 {
30     for (int i = 1; i <= v; i++)
31     {
32         cout << "\t";
33         for (int j = 1; j <= v; j++)
34         {
35             cout << adjMat[i][j] << "\t";
36         }
37         cout << endl;
38     }
39 }
40
41 int main()
42 {
43     int vertice = 4;
44     initializeMat(vertice);
45
46     //menambahkan nilai edge
47     addEdge(1,2,5);
48     addEdge(2,3,1);
49     addEdge(4,1,3);
50     addEdge(2,4,1);
51     addEdge(3,1,1);
52
53     displayMat(vertice);
54
55     return 0;
56 }
```

## OUTPUT



```
0      5      1      3
5      0      1      1
1      1      0      0
3      1      0      0

-----
Process exited after 0.1466 seconds with return value 0
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 yang diserang 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.

Input :

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

Output :

1. Jalur yang paling cepat ditempuh oleh pedagang untuk ke kastil tanpa melewati kota yang diserang naga
2. Jarak yang ditempuh

# CODINGAN

[\*] 2. Naga (UAS STRUKDAT).cpp ✕

```
1  #include <iostream>
2  #include <conio.h>
3  #include <string.h>
4  using namespace std;
5  int main()
6  {
7      char kota1,kota2,kota3,kota4,kota5;
8      int jumlah,panjang, hasil1,hasil2,hasil3,hasil4,hasil5,hasil6,hasil7;
9
10     cout<<"          _Kania Meliana Fityanti_"<<endl;
11     cout<<"          _21091397028_"<<endl;
12
13     cout<<endl;
14
15     // input jumlah kota
16     cout<<"* Jumlah kota yang berada di Kerajaan Britan : "<< endl;
17     cin>>jumlah;
18
19     // input nama kota
20     // deklarasi vertex
21     // menampilkan masing-masing vertex
22     cout<<"Kota Pertama : ";
23     cin>>kota1;
24     cout<<"Kota Kedua   : ";
25     cin>>kota2;
26     cout<<"Kota Ketiga  : ";
27     cin>>kota3;
28     cout<<"Kota Keempat  : ";
29     cin>>kota4;
30     cout<<"Kota kelima   : ";
31     cin>>kota5;
32
33     cout<<endl;
34
35     // deklarasi edge
36     // menampilkan setiap edge yang terjadi
```

[\*] 2. Naga (UAS STRUKDAT).cpp ✕

```
34
35     // deklarasi edge
36     // menampilkan setiap edge yang terjadi
37     cout<<"* Sisi-sisinya adalah : "<<endl<<endl;
38     cout<<kota1<<kota2<<" ";
39     cout<<kota1<<kota4<<" ";
40     cout<<kota1<<kota5<<" ";
41     cout<<kota2<<kota3<<" ";
42     cout<<kota3<<kota5<<" ";
43     cout<<kota3<<kota4<<" ";
44     cout<<kota4<<kota5<<endl<<endl;
45
46     // deklarasi weight
47     // menampilkan panjang jalan yang menghubungkan vertex
48     cout<<"* Panjang jalan antar kota : "<<endl;
49     cout<<"panjang "<<kota1<<" ke "<<kota2<<" : "; cin>> hasil1;
50     cout<<"panjang "<<kota1<<" ke "<<kota4<<" : "; cin>> hasil2;
51     cout<<"panjang "<<kota1<<" ke "<<kota5<<" : "; cin>> hasil3;
52     cout<<"panjang "<<kota2<<" ke "<<kota3<<" : "; cin>> hasil4;
53     cout<<"panjang "<<kota3<<" ke "<<kota5<<" : "; cin>> hasil5;
54     cout<<"panjang "<<kota3<<" ke "<<kota4<<" : "; cin>> hasil6;
55     cout<<"panjang "<<kota4<<" ke "<<kota5<<" : "; cin>> hasil7;
56
57     cout<<endl;
58
59     //deklarasi adjacent
60     //menampilkan jalan yang menghubungkan kedua simpul (x,y,z)
61     cout<<"* seluruh jalan yang ada dalam Kerajaan Britan dan panjang jalannya : "<< endl;
62     cout<<"("<<kota1<<","<<kota2<<","<<hasil1<<") ";
63     cout<<"("<<kota1<<","<<kota4<<","<<hasil2<<") ";
64     cout<<"("<<kota1<<","<<kota5<<","<<hasil3<<") ";
65     cout<<"("<<kota2<<","<<kota3<<","<<hasil4<<") ";
66     cout<<"("<<kota3<<","<<kota5<<","<<hasil5<<") ";
67     cout<<"("<<kota3<<","<<kota4<<","<<hasil6<<") ";
68     cout<<"("<<kota4<<","<<kota5<<","<<hasil7<<") ";
69
```

[\*] 2. Naga (UAS STRUKDAT).cpp ✕

```
69
70     cout<<endl<<endl;
71
72     //hasil
73     //menampilkan tempat pedagang berada
74     cout<<"* kota tempat pedagang sekarang berada : "<<endl<<endl;
75     cout<<kota1;
76
77     cout<<endl<<endl;
78
79     //menampilkan kota yang diserang naga
80     cout<<"* kota yang diserang naga : "<<endl<<endl;
81     cout<<kota3;
82
83     cout<<endl<<endl;
84
85     //menampilkan kota yang terdapat kastil
86     cout<<"* kota yang memiliki kastil : "<<endl<<endl;
87     cout<<kota5;
88
89     cout<<endl<<endl;
90
91     //menampilkan vertex tercepat untuk selamat
92     cout<<"* jalur yang paling cepat ditempuh : "<<endl<<endl;
93     cout<<kota1<<"-<<kota4<<"-<<kota5<<endl;
94
95     cout<<endl<<endl;
96
97     //total edge yang harus ditempuh
98     cout<<"* dengan jarak yang ditempuh : "<<endl<<endl;
99     cout<<hasil2+hasil7<<endl<<endl;
100
101
102     getch();
103     return 0;
104 }
```

## OUTPUT

```
D:\struktur data\2. Naga (UAS STRUKDAT).exe
_Kania Meliana Fityanti_
_21091397028_

* Jumlah kota yang berada di Kerajaan Britan :
5
Kota Pertama : 1
Kota Kedua : 2
Kota Ketiga : 3
Kota Keempat : 4
Kota Kelima : 5

* Sisi-sisinya adalah :
12,14,15,23,35,34,45

* Panjang jalan antar kota :
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 :
1-4-5

* dengan jarak yang ditempuh :
21
```