

LAPORAN UAS STRUKTUR DATA



Disusun oleh:
Widya Sari Wibowo
21091397070
Kelas B

**UNIVERSITAS NEGERI SURABAYA
PRODI D4 MANAJEMEN INFORMATIKA
2021**

Nomor 1

Kodingan:

UAS Nomor 1 (070).cpp UAS Nomor 2 (070).cpp

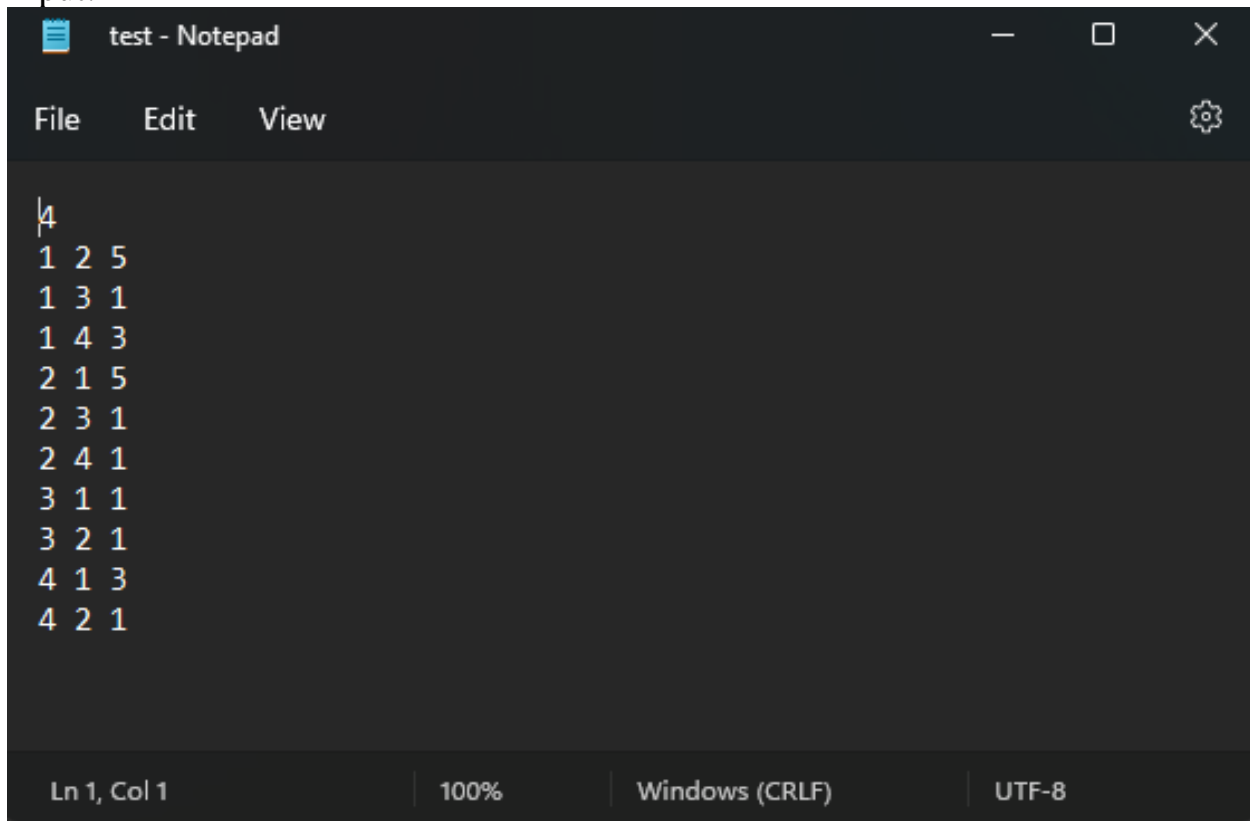
```
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <list>
5  using namespace std;
6
7  class edge
8  {
9  private:
10     int nodeid, weight;
11 public:
12     edge (int id, int w)
13     {
14         nodeid = id;
15         weight = w;
16     }
17     int getnodeid ()
18     {
19         return nodeid;
20     }
21     int getweightid ()
22     {
23         return weight;
24     }
25 };
26
27
```

```
28 int main ()
29 {
30     int totalnode, node1, node2, w;
31     ifstream input ("test.txt"); // untuk mengambil data dari file test.txt
32     input >> totalnode;
33     vector < list <edge> > adjlist (totalnode);
34     while (input >> node1 >> node2 >> w)
35     {
36         adjlist [node1-1] .push_back (edge (node2, w));
37     }
38     int c = 1;
39     vector < list <edge> > :: iterator i;
40     for (i = adjlist.begin(); i != adjlist.end(); i++)
41     {
42         cout << c << " -> ";
43         list <edge> li = *i;
44         list <edge> :: iterator iter;
45         for (iter = li.begin(); iter != li.end(); iter++)
46         {
47             cout << " [" << (*iter) .getnodeid() << ", " << (*iter) .getweightid() << "]" -> ";
48         }
49         cout << endl;
50         c++;
51     }
52     return 0;
53 }
```

Baris 31 terdapat sebuah perintah yang berfungsi untuk membaca sebuah data dari suatu file. Tempatkan file test.txt dan file kodingan dalam satu

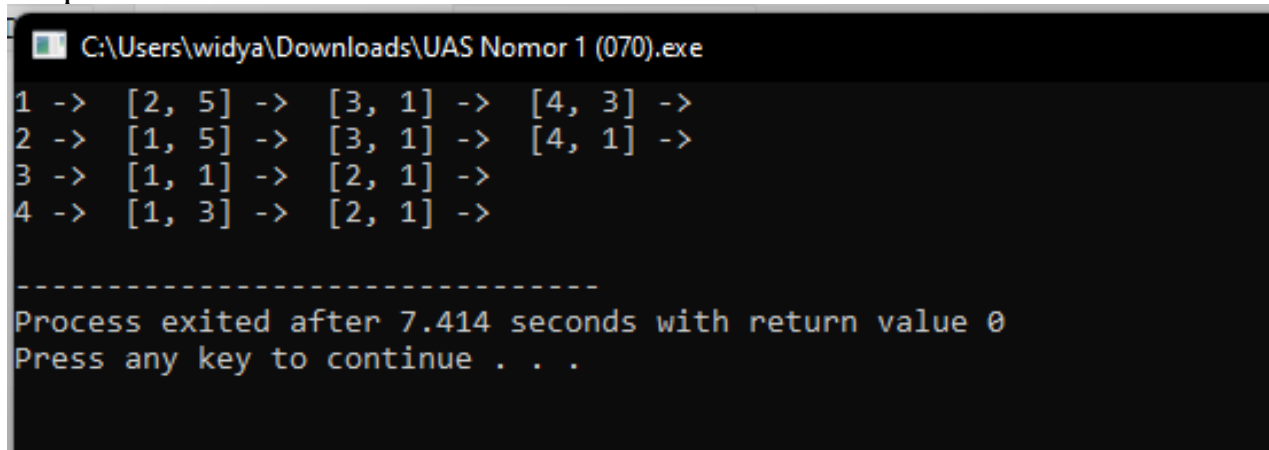
folder/directory yang sama agar data bisa terbaca sehingga dapat menghasilkan output graph.

Input:



```
test - Notepad
File Edit View
4
1 2 5
1 3 1
1 4 3
2 1 5
2 3 1
2 4 1
3 1 1
3 2 1
4 1 3
4 2 1
Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

Output:



```
C:\Users\widya\Downloads\UAS Nomor 1 (070).exe
1 -> [2, 5] -> [3, 1] -> [4, 3] ->
2 -> [1, 5] -> [3, 1] -> [4, 1] ->
3 -> [1, 1] -> [2, 1] ->
4 -> [1, 3] -> [2, 1] ->
-----
Process exited after 7.414 seconds with return value 0
Press any key to continue . . .
Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

Nomor 1 Revisi

Kodingan:

UAS Nomor 1 Revisi (070).cpp

```
1  #include<iostream>
2  #define Max 100
3
4  using namespace std;
5
6  int adjMat[Max][Max];
7
8  // Inisialisasi 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 // 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
34         for (int j = 1; j <= v; j++)
35         {
36             cout << adjMat[i][j] << "\t";
37         }
38
39         cout << endl;
40     }
41 }
42
43 int main()
44 {
45     int vertice = 4;
46
47     initializeMat(vertice);
48
49     addEdge(1,2,5);
50     addEdge(2,3,1);
51     addEdge(4,1,3);
52     addEdge(2,4,1);
53     addEdge(3,1,1);
54
55     displayMat(vertice);
56
57     return 0;
58 }
```

Output:

```
C:\Users\widya\Downloads\UAS Nomor 1 Revisi (070).exe

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

-----
Process exited after 6.28 seconds with return value 0
Press any key to continue . . .
```

Nomor 2

Kodingan:

```
UAS Nomor 1 (070).cpp  UAS Nomor 2 (070).cpp
1  #include <iostream>
2  #include <conio.h>
3  #include <string.h>
4  using namespace std;
5  int main()
6  {
7      char kotal, kota2, kota3, kota4, kota5;
8      int jumlah, panjang, hasil1, hasil2, hasil3, hasil4, hasil5, hasil6, hasil7;
9
10     cout << endl;
11
12     cout << "* Berapa jumlah kota dalam kerajaan Britan : " << endl;
13     cin >> jumlah;
14
15     // deklarasi vertex
16     // untuk menampilkan masing-masing vertex
17     cout << "Kota Pertama : ";
18     cin >> kotal;
19     cout << "Kota Kedua : ";
20     cin >> kota2;
21     cout << "Kota Ketiga : ";
22     cin >> kota3;
23     cout << "Kota Keempat : ";
24     cin >> kota4;
25     cout << "Kota kelima : ";
26     cin >> kota5;
27
28     cout << endl;
29
30     // deklarasi edge
31     // untuk menampilkan setiap edge yang terjadi
32     cout << "* Sisi-sisinya adalah : " << endl << endl;
33     cout << kotal << kota2 << ", ";
34     cout << kota2 << kota3 << ", ";
35     cout << kota3 << kota4 << ", ";
36     cout << kota4 << kota5 << ", ";
37     cout << kotal << kota3 << ", ";
38     cout << kotal << kota5 << ", ";
39     cout << kota2 << kota5 << ", ";
40     cout << endl;
```

```

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

```

```

64
65 cout << endl << endl;
66
67 // hasil
68 // untuk menampilkan tempat pedagang berada
69 cout << "* kota tempat pedagang sekarang berada : " << endl << endl;
70 cout << kotal;
71
72 cout << endl << endl;
73
74 // untuk menampilkan kota yang diserang naga
75 cout << "* kota yang diserang naga : " << endl << endl;
76 cout << kota3;
77
78 cout << endl << endl;
79
80 // untuk menampilkan kota yang memiliki kastil
81 cout << "* kota yang memiliki kastil : " << endl << endl;
82 cout << kota5;
83
84 cout << endl << endl;
85
86 // untuk menampilkan vertex tercepat untuk selamat
87 cout << "* jalur yang paling cepat ditempuh : " << endl << endl;
88 cout << kotal << "-" << kota4 << "-" << kota5 << endl;
89
90 cout << endl << endl;
91
92 // untuk menampilkan total edge yang harus ditempuh
93 cout << "* dengan jarak : " << endl << endl;
94 cout << hasil2+hasil7 << endl << endl;
95
96
97 getch();
98 return 0;
99

```

Output:

```
C:\Users\widya\Downloads\UAS Nomor 2 (070).exe

Berapa jumlah kota dalam 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 :

21
```