

**LAPORAN STRUKTUR DATA**  
**PROJECT UAS ADJACENCY**



Di susun oleh:

Nama: Riska Octavia Arianto (21091397059)

Kelas: A

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

Dalam laporan ini saya jabarkan tentang algoritma yang saya buat:

1. Membuat undirected graph menggunakan representasi adjacency list dengan input vertex dan edge dengan input :

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

Dan Output :

Satu per satu vertex, edge, dan weightnya

Dengan algoritma seperti berikut:

```
1  #include <iostream>
2  using namespace std;
3  // stores adjacency list items
4  struct adjNode {
5      int val, cost;
6      adjNode* next;
7  };
8  // structure to store edges
9  struct graphEdge {
10     int start_ver, end_ver, weight;
11 };
12 class DiaGraph{
13     // insert new nodes into adjacency list from given graph
14     adjNode* getAdjListNode(int value, int weight, adjNode* head) {
15         adjNode* newNode = new adjNode;
16         newNode->val = value;
17         newNode->cost = weight;
18
19         newNode->next = head;    // point new node to current head
20         return newNode;
21     }
22     int N, i; // number of nodes in the graph
23 public:
24     adjNode **head;           //adjacency list as array of pointers
25     // Constructor
26     DiaGraph(graphEdge edges[], int n, int N) {
27         // allocate new node
28         head = new adjNode*[N]();
29         this->N = N;
30         // initialize head pointer for all vertices
31         for (int i = 0; i < N; ++i);
32         head[i] = NULL;
```

```

33         // construct directed graph by adding edges to it
34         for (unsigned i = 0; i < n; i++) {
35             int start_ver = edges[i].start_ver;
36             int end_ver = edges[i].end_ver;
37             int weight = edges[i].weight;
38             // insert in the beginning
39             adjNode* newNode = getAdjListNode(end_ver, weight, head[start_ver]);
40
41             // point head pointer to new node
42             head[start_ver] = newNode;
43         }
44     }
45     // Destructor
46     ~DiaGraph() {
47         for (int i = 0; i < N; i++)
48             delete[] head[i];
49         delete[] head;
50     }
51 };
52 // print all adjacent vertices of given vertex
53 void display_AdjList(adjNode* ptr, int i)
54 {
55     while (ptr != NULL) {
56         cout << "(" << i << ", " << ptr->val
57             << ", " << ptr->cost << ") ";
58         ptr = ptr->next;
59     }
60     cout << endl;
61 }

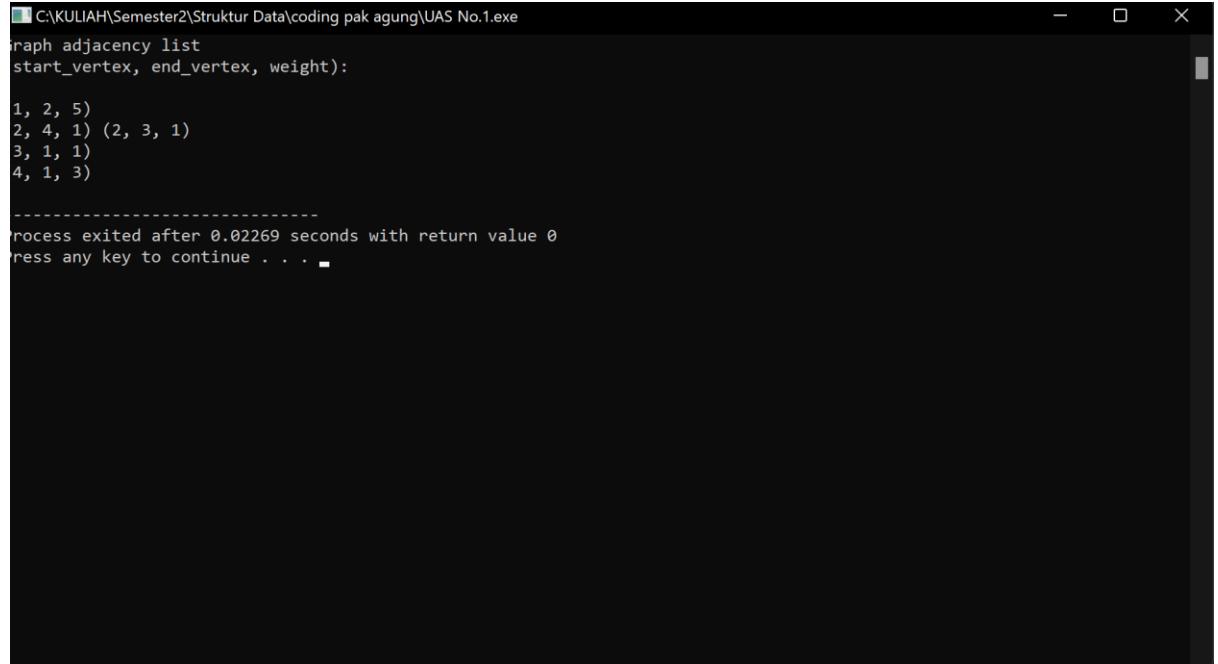
```

```

62 // graph implementation
63 int main()
64 {
65     // graph edges array.
66     graphEdge edges[] = {
67         // (x, y, w) -> edge from x to y with weight w
68         {1,2,5},{2,3,1},{4,1,3},{2,4,1},{3,1,1}
69     };
70     int N = 5;        // Number of vertices in the graph
71     // calculate number of edges
72     int n = sizeof(edges)/sizeof(edges[0]);
73     // construct graph
74     DiaGraph diagraph(edges, n, N);
75     // print adjacency list representation of graph
76     cout<<"Graph adjacency list "<<endl<<"(start_vertex, end_vertex, weight):"<<endl;
77     for (int i = 0; i < N; i++)
78     {
79         // display adjacent vertices of vertex i
80         display_AdjList(diagraph.head[i], i);
81     }
82     return 0;
83 }
84

```

Menghasilkan output seperti berikut:



```
C:\KULIAH\Semester2\Struktur Data\coding pak agung\UAS No.1.exe
graph adjacency list
start_vertex, end_vertex, weight):

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

-----
Process exited after 0.02269 seconds with return value 0
Press any key to continue . . .
```

## 2. Soal Cerita

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.

Dengan Input :

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

Dan Output :

Satu per satu vertex, edge, dan weightnya

Berikut adalah hasil algoritma yang saya buat:

```

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     //input jumlah kota
11     cout<<"* Jumlah kota yang berada di kerajaan Britan : "<< endl;
12     cin>>jumlah;
13
14     //input nama kota
15     cout<<"Kota Pertama : ";
16     cin>>kota1;
17     cout<<"Kota Kedua   : ";
18     cin>>kota2;
19     cout<<"Kota Ketiga  : ";
20     cin>>kota3;
21     cout<<"Kota Keempat : ";
22     cin>>kota4;
23     cout<<"Kota kelima  : ";
24     cin>>kota5;
25
26     cout<<endl;
27

```

```

28     //deklarasi graph
29     //menampilkan graph yang terjadi
30     cout<<"* Sisi-sisinya adalah : "<<endl<<endl;
31     cout<<kota1<<kota2<<" ";
32     cout<<kota1<<kota4<<" ";
33     cout<<kota1<<kota5<<" ";
34     cout<<kota2<<kota3<<" ";
35     cout<<kota3<<kota5<<" ";
36     cout<<kota3<<kota4<<" ";
37     cout<<kota4<<kota5<<endl<<endl;
38
39     //deklarasi edge
40     //menampilkan panjang jalan yang menghubungkan vertex
41     cout<<"* Panjang jalan antar kota : "<<endl;
42     cout<<"panjang "<<kota1<<" ke "<<kota2<<" : "; cin>> hasil1;
43     cout<<"panjang "<<kota1<<" ke "<<kota4<<" : "; cin>> hasil2;
44     cout<<"panjang "<<kota1<<" ke "<<kota5<<" : "; cin>> hasil3;
45     cout<<"panjang "<<kota2<<" ke "<<kota3<<" : "; cin>> hasil4;
46     cout<<"panjang "<<kota3<<" ke "<<kota5<<" : "; cin>> hasil5;
47     cout<<"panjang "<<kota3<<" ke "<<kota4<<" : "; cin>> hasil6;
48     cout<<"panjang "<<kota4<<" ke "<<kota5<<" : "; cin>> hasil7;
49
50     cout<<endl;
51

```

```

52 //deklarasi adjacent
53 //menampilkan jalan yang menghubungkan kedua simpul (x,y,z)
54 cout<<"* seluruh jalan yang ada dalam kerajaan britan dan panjang jalannya : "<< endl;
55 cout<<"("<<kota1<<","<<kota2<<","<<hasil1<<") ";
56 cout<<"("<<kota1<<","<<kota4<<","<<hasil2<<") ";
57 cout<<"("<<kota1<<","<<kota5<<","<<hasil3<<") ";
58 cout<<"("<<kota2<<","<<kota3<<","<<hasil4<<") ";
59 cout<<"("<<kota3<<","<<kota5<<","<<hasil5<<") ";
60 cout<<"("<<kota3<<","<<kota4<<","<<hasil6<<") ";
61 cout<<"("<<kota4<<","<<kota5<<","<<hasil7<<") ";
62
63 cout<<endl<<endl;
64
65 //hasil
66 //menampilkan tempat pedagang berada
67 cout<<"* kota tempat pedagang sekarang berada : "<<endl<<endl;
68 cout<<kota1;
69
70 cout<<endl<<endl;
71
72 //menampilkan kota yang diserang naga
73 cout<<"* kota yang diserang naga : "<<endl<<endl;
74 cout<<kota3;
75
76 cout<<endl<<endl;
77
78 //menampilkan kota yang terdapat kastil
79 cout<<"* kota yang memiliki kastil : "<<endl<<endl;
80 cout<<kota5;
81
82 cout<<endl<<endl;
83
84 //menampilkan vertex tercepat untuk selamat
85 cout<<"* jalur yang paling cepat ditempuh : "<<endl<<endl;
86 cout<<kota1<<"-"<<kota4<<"-"<<kota5<<endl;
87

```

```

88 cout<<endl<<endl;
89
90 //total edge yang harus ditempuh
91 cout<< "* dengan jarak : "<<endl<<endl;
92 cout<<hasil2+hasil7<<endl<<endl;
93
94 // pesan developer
95 cout << " "<<endl;
96 cout << "\ndevelop @riskaoctavia";
97
98 getch();
99 return 0;
100

```

Dengan output seperti berikut:

C:\KULIAH\Semester2\Struktur Data\coding pak agung\UAS No.2.exe

Jumlah kota yang berada di kerajaan Britan :

kota Pertama : a  
kota Kedua : b  
kota Ketiga : c  
kota Keempat : d  
kota kelima : e

Sisi-sisinya adalah :

b,ad,ae,bc,ce,cd,de

Panjang jalan antar kota :

panjang a ke b: 12  
panjang a ke d: 11  
panjang a ke e: 30  
panjang b ke c: 14  
panjang c ke e: 5  
panjang c ke d: 15  
panjang d ke e: 10

seluruh jalan yang ada dalam kerajaan britan dan panjang jalannya :

a,b,12) (a,d,11) (a,e,30) (b,c,14) (c,e,5) (c,d,15) (d,e,10)

kota tempat pedagang sekarang berada :

kota yang diserang naga :

C:\KULIAH\Semester2\Struktur Data\coding pak agung\UAS No.2.exe

a

\* kota yang diserang naga :

c

\* kota yang memiliki kastil :

e

\* jalur yang paling cepat ditempuh :

a-d-e

\* dengan jarak :

21

develop @riskaoctavia\_