

Laporan UAS Struktur Data

Nama : Muhammad Yuda Pratama
NIM : 21091397025
Kelas : A

NO 1

```
1  #include <iostream>
2  using namespace std;
3
4  //menyimpan item daftar kedekatan
5  struct adjNode {
6      int val, cost;
7      adjNode* next;
8  };
9  //struktur untuk menyimpan edges
10 struct graphEdge {
11     int start_ver, end_ver, weight;
12 };
13 class DiaGraph{
14     //masukkan node baru ke dalam daftar kedekatan dari grafik yang diberikan
15     adjNode* getAdjListNode(int value, int weight, adjNode* head){
16         adjNode* newNode = new adjNode;
17         newNode->val = value;
18         newNode->cost = weight;
19
20         newNode->next = head; // arahkan simpul baru ke kepala saat ini
21         return newNode;
22     }
23     int N; //jumlah node dalam grafik
24 public:
25     adjNode **head; //daftar kedekatan sebagai array pointer
26     // Constructor
27     DiaGraph(graphEdge edges[], int n, int N) {
28         // mengalokasikan simpul baru
29         head = new adjNode*[N]();
30         this->N = N;
31         // initialize head pointer for all vertices
32         for (int i = 0; i < N; ++i)
33             head[i] = nullptr;
34         //inisialisasi penunjuk kepala untuk semua simpul
35         for (unsigned i = 0; i < n; i++) {
36             int start_ver = edges[i].start_ver;
37             int end_ver = edges[i].end_ver;
38             int weight = edges[i].weight;
39             // masukkan di awal
40             adjNode* newNode = getAdjListNode(end_ver, weight, head[start_ver]);
41
42             // arahkan penunjuk kepala ke simpul baru
43             head[start_ver] = newNode;
44         }
45     }
```

```

46     // Destructor
47     ~DiaGraph() {
48         for (int i = 0; i < N; i++)
49             delete[] head[i];
50         delete[] head;
51     }
52 };
53 // cetak semua simpul yang berdekatan dari simpul yang diberikan
54 void display_AdjList(adjNode* ptr, int i)
55 {
56     while (ptr != nullptr) {
57         cout << i << " -> " << "[" << ptr->val
58             << ", " << ptr->cost << "]" << " ";
59         ptr = ptr->next;
60     }
61     cout << endl;
62 }
64 int main()
65 {
66     // array tepi grafik
67     graphEdge edges[] = {
68         // (x, y, w) -> tepi dari x ke y dengan bobot w
69         {1,2,5},{2,3,1},{4,1,3},{2,4,1},{3,1,1}
70     };
71     int N = 4; // Jumlah simpul dalam grafik
72     // hitung jumlah rusuknya
73     int n = sizeof(edges)/sizeof(edges[0]);
74     // construct graph
75     DiaGraph diagraph(edges, n, N);
76     // cetak representasi daftar kedekatan grafik
77     for (int i = 0; i < N; i++)
78     {
79         // menampilkan simpul yang berdekatan dari simpul i
80         display_AdjList(diagraph.head[i], i);
81     }
82     return 0;
83 }

```

Output :

```

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

```