

Nama : Frederico Steven Kwok

NPM : 242310037

Kelas : TI-24-PA

Matkul : Lab. Desain dan Analisis Algoritma

Github : https://github.com/RicoSteven120206/Desain_Analisis_Algoritma.git

Tugas : Membuat program C++ dengan konsep Graph 2

A. Membuat program sesuai dengan output yang diberikan

1. Screenshot Source Code

```
1  #include <bits/stdc++.h>
2  #include <iostream>
3  #define MAX 100005
4  #define INF INT_MAX
5  using namespace std;
6
7  vector<pair<int, int> > adj[MAX];
8  bool vis[MAX];
9  int dist[MAX];
10
11 class graph {
12 public:
13     //membuat function dijkstra
14     void djikstra(int start) {
15         memset(vis, false, sizeof vis);
16         for (int i = 0; i < MAX; i++) {
17             dist[i] = INF;
18         }
19         dist[start] = 0;
20
21         priority_queue<pair<int, int>, vector<pair<int, int> >, greater<pair<int, int> > > pq;
22         pq.push({0, start});
23         while(!pq.empty()) {
24             pair<int, int> p = pq.top();
25             pq.pop();
26             int x = p.second;
27             if (vis[x]) {
28                 continue;
29             }
30             vis[x] = true;
31         }
```

```

32 for (int i = 0; i < adj[x].size(); i++) {
33     int e = adj[x][i].first;
34     int w = adj[x][i].second;
35     if (dist[x] + w < dist[e]) {
36         dist[e] = dist[x] + w;
37         pq.push({dist[e], e});
38     }
39 }
40 }
41 }
42 };
43
44 int main() {
45     graph g;
46     int n;
47     cout<<"Masukkan jumlah vertex : ";
48     cin>>n;
49
50     for(int i = 0; i < n + 1; i++) {
51         int value;
52         cout<<"Distance : ";
53         cin>>value;
54         adj[0].push_back({i, value});
55     }
56     g.dijkstra(0);
57
58     cout<<"\n-----\n";
59     cout<<"Vertex\tDistance from Source"<<endl;
60
61     for (int i = 0; i < n + 1; i++) {
62         cout<<" " <<adj[0][i].first<<"\t\t"<<dist[i]<<endl;
63     }
64
65     return 0;
66 }

```

2. Screenshot Output

```

Masukkan jumlah vertex : 8
Distance : 0
Distance : 4
Distance : 12
Distance : 19
Distance : 21
Distance : 11
Distance : 9
Distance : 8
Distance : 14

```

```

-----
Vertex Distance from Source
0         0
1         4
2        12
3        19
4        21
5        11
6         9
7         8
8        14
-----

```

3. Screenshot pengerjaan

21.26
27/06/2025

6

- B. Dari modul, membuat program C++ berdasarkan gambar dengan konsep rute terpendek hal. 87

1. Screenshot Source Code

```
1  #include <bits/stdc++.h>
2  #include <iostream>
3  #define MAX 100005
4  #define INF INT_MAX
5  using namespace std;
6
7  vector<pair<int, int> > adj[MAX];
8  bool vis[MAX];
9  int dist[MAX];
10 int route[MAX];
11
12 class graph {
13 public:
14     // membuat function dijkstra dengan parameter start
15     void dijkstra(int start) {
16         memset(vis, false, sizeof vis);
17         for (int i = 0; i < MAX; i++) {
18             dist[i] = INF;
19         }
20         dist[start] = 0;
21
22         priority_queue<pair<int, int>, vector<pair<int, int> >, greater<pair<int, int> > > pq;
23         pq.push({0, start});
24
25         while(!pq.empty()) {
26             pair<int, int> p = pq.top();
27             pq.pop();
28             int x = p.second;
29             if (vis[x]) {
30                 continue;
31             }
32
33             vis[x] = true;
34
35             for (int i = 0; i < adj[x].size(); i++) {
36                 int e = adj[x][i].first;
37                 int w = adj[x][i].second;
38                 if (dist[x] + w < dist[e]) {
39                     dist[e] = dist[x] + w;
40                     route[e] = x;
41                     pq.push({dist[e], e});
42                 }
43             }
44         }
45     }
46
47     void addEdge(vector<pair<int, int> > *adj, int from, int to, int weight) {
48         adj[from].push_back({to, weight});
49         adj[to].push_back({from, weight}); // undirected graph
50     }
51
52     // membuat function print rute untuk membuat rute
53     void printRoute(int start, int end) {
54         if (dist[end] == INF) {
55             cout << "Tidak ada rute dari node " << start << " ke node " << end << endl;
56             return;
57         }
58     }
```

```

59     vector<int> path; // deklarasi path dengan vector
60     int currentNode = end;
61     while (currentNode != 0) {
62         path.push_back(currentNode);
63         currentNode = route[currentNode];
64     }
65
66     reverse(path.begin(), path.end());
67     for (size_t i = 0; i < path.size(); ++i) {
68         cout << path[i] << (i == path.size() - 1 ? "" : " -> "); // fungsi ternary
69     }
70 }
71 };
72
73 int main() {
74     graph g;
75     int dari, ke, n;
76     cout<<"Masukkan jumlah node: ";
77     cin>>n;
78
79     for (int i = 0; i < n; i++) {
80         int x, y, z;
81         cout<<"Simpul asal : ";
82         cin>>x;
83         cout<<"Simpul Tujuan : ";
84         cin>>y;
85         cout<<"Bobot : ";
86         cin>>z;
87         cout<<endl;
88         g.addEdge(adj, x, y, z);
89     }
90     cout << "> Masukkan simpul asal: ";
91     cin >> dari;
92     cout << "> Masukkan simpul tujuan: ";
93     cin >> ke;
94
95     g.dijkstra(dari);
96
97     cout << "Jarak terpendek dari node " << dari << " ke node " << ke << " adalah: " << dist[ke] << endl;
98     cout << "Rute: " << endl;
99     g.printRoute(dari, ke);
100    cout << endl;
101
102    system("pause");
103
104    return 0;
105 }

```

2. Screenshot Output

```
Masukkan jumlah node: 9  
Simpul asal : 1  
Simpul Tujuan : 2  
Bobot : 7
```

```
Simpul asal : 1  
Simpul Tujuan : 3  
Bobot : 9
```

```
Simpul asal : 1  
Simpul Tujuan : 6  
Bobot : 14
```

```
Simpul asal : 2  
Simpul Tujuan : 3  
Bobot : 10
```

```
Simpul asal : 2  
Simpul Tujuan : 4  
Bobot : 15
```

```
Simpul asal : 3  
Simpul Tujuan : 4  
Bobot : 11
```

```
Simpul asal : 3  
Simpul Tujuan : 6  
Bobot : 2
```

```
Simpul asal : 4  
Simpul Tujuan : 5  
Bobot : 6
```

```
Simpul asal : 5  
Simpul Tujuan : 6  
Bobot : 9
```

```
=> Masukkan simpul asal: 1  
=> Masukkan simpul tujuan: 5  
Jarak terpendek dari node 1 ke node 5 adalah: 20  
Rute:  
1 -> 3 -> 6 -> 5  
Press any key to continue . . . █
```

3. Screenshot pengerjaan

22.04
27/06/2025 8