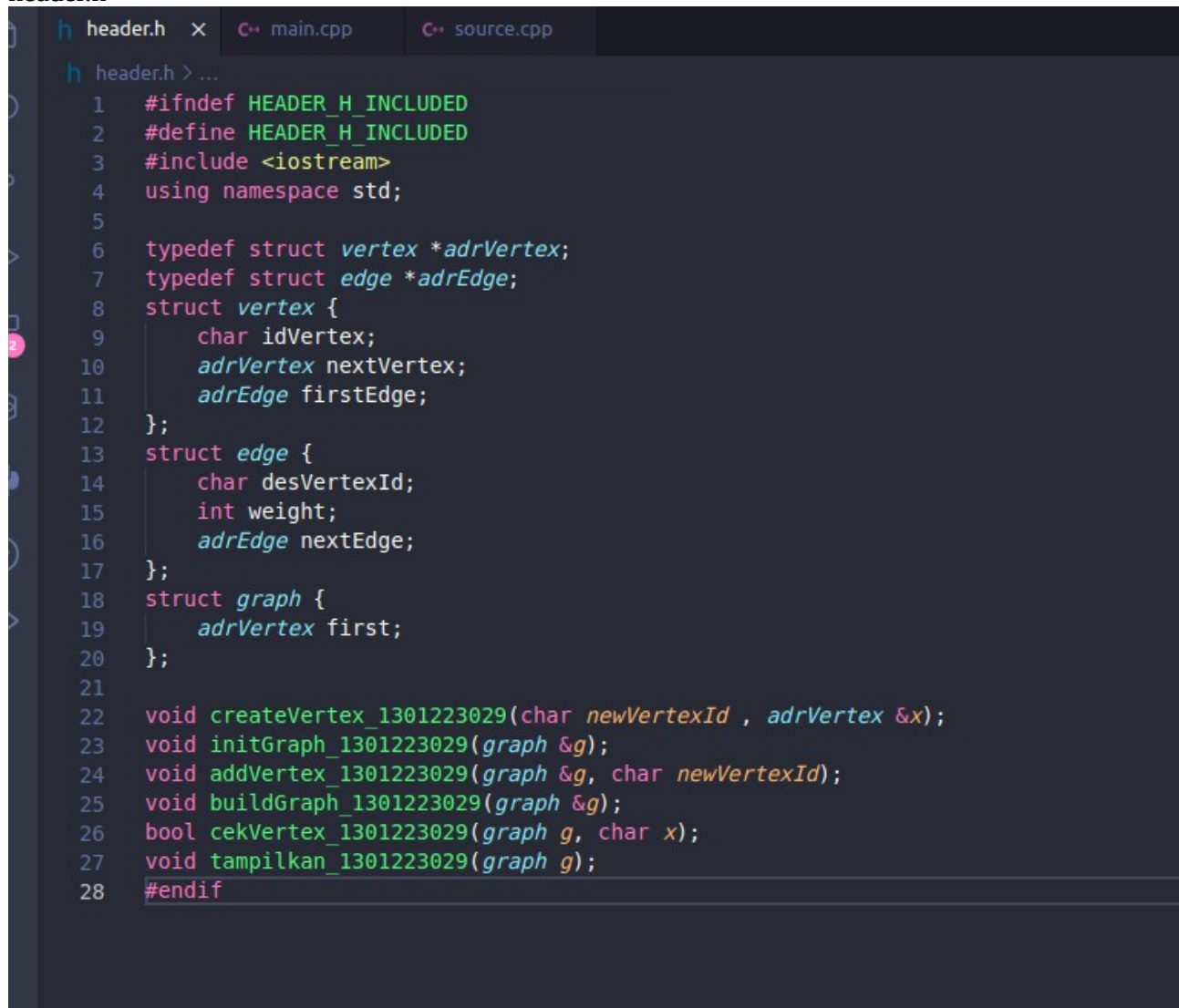


Deore Mufrad Hendrady
1301223029

header.h



```
h header.h x C++ main.cpp C++ source.cpp
h header.h > ...
1  #ifndef HEADER_H_INCLUDED
2  #define HEADER_H_INCLUDED
3  #include <iostream>
4  using namespace std;
5
6  typedef struct vertex *adrVertex;
7  typedef struct edge *adrEdge;
8  struct vertex {
9      char idVertex;
10     adrVertex nextVertex;
11     adrEdge firstEdge;
12 };
13 struct edge {
14     char desVertexId;
15     int weight;
16     adrEdge nextEdge;
17 };
18 struct graph {
19     adrVertex first;
20 };
21
22 void createVertex_1301223029(char newVertexId , adrVertex &x);
23 void initGraph_1301223029(graph &g);
24 void addVertex_1301223029(graph &g, char newVertexId);
25 void buildGraph_1301223029(graph &g);
26 bool cekVertex_1301223029(graph g, char x);
27 void tampilkan_1301223029(graph g);
28 #endif
```

source.cpp

```

C++ source.cpp > createVertex_1301223029(char, adrVertex &)
1  #include "header.h"
2  void createVertex_1301223029(char newVertexId , adrVertex &x){
3      x = new vertex;
4      x->firstEdge = NULL;
5      x->nextVertex = NULL;
6      x->idVertex = newVertexId;
7  }
8  void initGraph_1301223029(graph &g){
9      g.first = NULL;
10 }
11 void addVertex_1301223029(graph &g, char newVertexId){
12     adrVertex p;
13     createVertex_1301223029(newVertexId , p);
14     if(g.first == NULL){
15         g.first = p;
16     }else {
17         p->nextVertex = g.first;
18         g.first = p;
19     }
20 }
21 bool cekVertex_1301223029(graph g, char x){
22     // True kalo ada
23     bool duplikat = false;
24     adrVertex p;
25     p = g.first;
26     while(p != NULL && !duplikat){
27         if (p->idVertex == x){
28             duplikat = true;
29         }
30         p = p->nextVertex;
31     }
32
33     return duplikat;
34 }
35 void buildGraph_1301223029(graph &g){
36     char isi;
37     int i = 1;
38     initGraph_1301223029(g);
39     cout << "Masukkan vertex ke-" << i << " (A-Z) : ";
40     cin >> isi;
41     addVertex_1301223029(g,isi);
42     i++;
43     while (isi >= 'A' && isi <= 'Z'){
44         cout << "Masukkan vertex ke-" << i << " (A-Z) : ";
45         cin >> isi;
46         if (isi >= 'A' && isi <= 'Z'){
47             if (!cekVertex_1301223029(g,isi) ){
48                 addVertex_1301223029(g,isi);

```

```

source.cpp > createVertex_1301223029(char, adrVertex &)
31     ,
32
33     return duplikat;
34 }
35 void buildGraph_1301223029(graph &g){
36     char isi;
37     int i = 1;
38     initGraph_1301223029(g);
39     cout << "Masukkan vertex ke-" << i << " (A-Z) : ";
40     cin >> isi;
41     addVertex_1301223029(g,isi);
42     i++;
43     while (isi >= 'A' && isi <= 'Z'){
44         cout << "Masukkan vertex ke-" << i << " (A-Z) : ";
45         cin >> isi;
46         if (isi >= 'A' && isi <= 'Z'){
47             if (!cekVertex_1301223029(g,isi) ){
48                 addVertex_1301223029(g,isi);
49                 i++;
50             }else {
51                 cout << "Vertex sudah ada " << endl;
52             }
53         }else {
54             cout << "Input berakhir" << endl;
55         }
56     }
57 }
58 void tampilkan_1301223029(graph g){
59     adrVertex p ;
60     p = g.first;
61     int i = 1;
62     cout << "Isi Graph : " << endl;
63     while(p != NULL ){
64         cout << "    -Vertex ke-" << i << " : " << p->idVertex << endl;
65         p = p->nextVertex;
66         i++;
67     }
68
69 }
70
71

```

main.cpp

```
header.h  main.cpp  source.cpp
main.cpp > main()
1  #include "header.h"
2
3  int main(){
4      graph g;
5      buildGraph_1301223029(g);
6      tampilkan_1301223029(g);
7      return 0;
8  }
```

output

```
~/SID-TP/TP-14 > on P main ?3
./myprogram
Masukkan vertex ke-1 (A-Z) : E
Masukkan vertex ke-2 (A-Z) : D
Masukkan vertex ke-3 (A-Z) : C
Masukkan vertex ke-4 (A-Z) : B
Masukkan vertex ke-5 (A-Z) : A
Masukkan vertex ke-6 (A-Z) : -
Input berakhir
Isi Graph :
-Vertex ke-1 : A
-Vertex ke-2 : B
-Vertex ke-3 : C
-Vertex ke-4 : D
-Vertex ke-5 : E
```