Nama/NIM/Kelas : Muhammad Zhafran Ilham / 103022330008 / SE-47-01

TP Modul 11 Struktur Data

graph.h

graph.h graph.cpp main.cpp

```
#ifndef GRAPH_H_INCLUDED
#define GRAPH_H_INCLUDED
#include <iostream>
#define firstVertex(G) G.firstVertex
#define idVertex(V) V->idVertex
#define nextVertex(V) V->next
#define firstEdge(V) V->firstEdge
using namespace std;

typedef struct vertex *adrVertex;
typedef struct edge *adrEdge;

struct vertex {
    char idVertex;
    adrVertex next;
    adrEdge firstEdge;
};

struct edge {
    char destVertexID;
    int weight;
    adrEdge nextEdge;
};

struct graph {
    adrVertex firstVertex;
};

void createVertex_103022330008(char newVertexID, adrVertex &v);

void initGraph_103022330008(graph &G);

void addVertex_103022330008(graph &G, char newVertexID);

void buildGraph_103022330008(graph &G);

void showVertex(graph G);


#endif // GRAPH_H_INCLUDED

```

Nama/NIM/Kelas : Muhammad Zhafran Ilham / 103022330008 / SE-47-01

graph.cpp

graph.h | graph.cpp | main.cpp

```cpp
#include <iostream>
#include "graph.h"
using namespace std;

void createVertex_103022330008(char newVertexID, adrVertex &v) {
    adrVertex newVertex = new vertex;
    idVertex(newVertex) = newVertexID;
    nextVertex(newVertex) = NULL;
    firstEdge(newVertex) = NULL;
    v = newVertex;
}

void initGraph_103022330008(graph &G) {
    firstVertex(G) = NULL;
}

void addVertex_103022330008(graph &G, char newVertexID) {
    adrVertex newVertex;
    createVertex_103022330008(newVertexID, newVertex);

    if(firstVertex(G) == NULL) {
        firstVertex(G) = newVertex;
    } else {
        adrVertex vertex = firstVertex(G);
        while (nextVertex(vertex) != NULL) {
            vertex = nextVertex(vertex);
        }
        nextVertex(vertex) = newVertex;
    }
}

```

Nama/NIM/Kelas : Muhammad Zhafran Ilham / 103022330008 / SE-47-01

```cpp
void buildGraph_103022330008(graph &G) {
    char inputVertexID;

    while (true) {
        cout << "Masukkan ID Simpul (A-Z) : ";
        cin >> inputVertexID;

        if(!(inputVertexID >= 'A' && inputVertexID <= 'Z')) {
            cout << "ID Simpul yang anda masukkan bukan (A-Z)." << endl;
            break;
        }

        bool isUnique = true;
        adrVertex v = firstVertex(G);

        while (v != NULL) {
            if(idVertex(v) == inputVertexID) {
                isUnique = false;
                break;
            }
            v = nextVertex(v);
        }

        if(isUnique) {
            addVertex_103022330008(G, inputVertexID);
            cout << "Simpul " << inputVertexID << " berhasil ditambahkan." << endl;
        } else {
            cout << "ID Simpul " << inputVertexID << " yang anda masukkan sudah ada." << endl;
        }
    }
}
```

```cpp
void showVertex(graph G) {
    adrVertex v = firstVertex(G);

    while (v != NULL) {
        cout << idVertex(v) << " ";
        v = nextVertex(v);
    }
}
```

Nama/NIM/Kelas : Muhammad Zhafran Ilham / 103022330008 / SE-47-01

main.cpp

graph.h | graph.cpp | main.cpp

```cpp
#include <iostream>
#include "graph.h"
using namespace std;

int main()
{
    graph G;
    initGraph_103022330008(G);

    buildGraph_103022330008(G);

    showVertex(G);

    return 0;
}
```

Running Program

```
"E:\Telkom University\Aset Kuliah\Semester 3\Praktikum Struktur Data\TP_Modul_11\bin\Debug\TP_Modul_11.exe"
Masukkan ID Simpul (A-Z) : A
Simpul A berhasil ditambahkan.
Masukkan ID Simpul (A-Z) : A
ID Simpul A yang anda masukkan sudah ada.
Masukkan ID Simpul (A-Z) : B
Simpul B berhasil ditambahkan.
Masukkan ID Simpul (A-Z) : C
Simpul C berhasil ditambahkan.
Masukkan ID Simpul (A-Z) : C
ID Simpul C yang anda masukkan sudah ada.
Masukkan ID Simpul (A-Z) : D
Simpul D berhasil ditambahkan.
Masukkan ID Simpul (A-Z) : E
Simpul E berhasil ditambahkan.
Masukkan ID Simpul (A-Z) : 2
ID Simpul yang anda masukkan bukan (A-Z).
A B C D E
Process returned 0 (0x0)   execution time : 11.339 s
Press any key to continue.
```