**PERTEMUAN KE 13 STRUKTUR DATA Graph Operation**



**Disusun oleh:**

**Muhammad Rendi Al Sungkar 2022320010**

**Muhamamad Jaryas Kurillah 2022310001**

**Aditya Nugroho 2022320001**

**Firman Maulana Firdaus 2022310032**

**Imanuel Selan 2022310046**

**JURUSAN INFORMATIKA**

**UNIVERSITAS BINA INSANI**

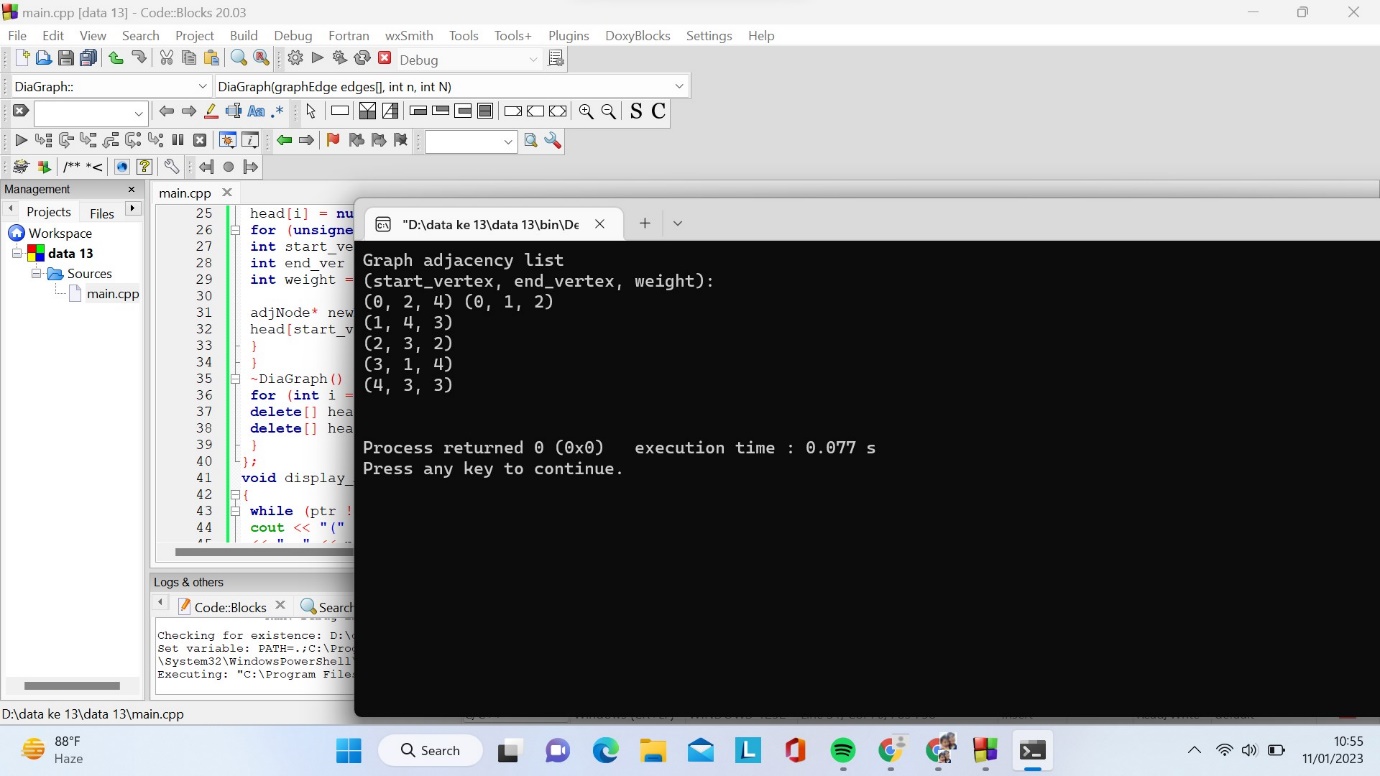
**BEKASI**

**2022**

**Membuat program “Graph Operation”**

Dibawah ini adalah tampilan pertama di program STRUKTUR DATA “**Graph Operation”**,

**Cokntoh tampilan awal**

****

Program diatas memiliki penjelasan dibawah ini;

Penjelasan

#include <iostream>

using namespace std;

struct adjNode {

int val, cost;

adjNode\* next;

};

struct graphEdge {

int start\_ver, end\_ver, weight;

};

class DiaGraph{

adjNode\* getAdjListNode(int value, int weight, adjNode\* head) {

adjNode\* newNode = new adjNode;

newNode->val = value;

newNode->cost = weight;

newNode->next = head;

return newNode;

}

int N;

public:

adjNode \*\*head;

DiaGraph(graphEdge edges[], int n, int N) {

head = new adjNode\*[N]();

this->N = N;

for (int i = 0; i < N; ++i)

head[i] = nullptr;

for (unsigned i = 0; i < n; i++) {

int start\_ver = edges[i].start\_ver;

int end\_ver = edges[i].end\_ver;

int weight = edges[i].weight;

adjNode\* newNode = getAdjListNode(end\_ver, weight, head[start\_ver]);

head[start\_ver] = newNode;

}

}

~DiaGraph() {

for (int i = 0; i < N; i++)

delete[] head[i];

delete[] head;

}

};

void display\_AdjList(adjNode\* ptr, int i)

{

while (ptr != nullptr) {

cout << "(" << i << ", " << ptr->val

<< ", " << ptr->cost << ") ";

ptr = ptr->next;

}

cout << endl;

}

int main()

{

graphEdge edges[] = {

{0,1,2},{0,2,4},{1,4,3},{2,3,2},{3,1,4},{4,3,3}

};

int N = 6;

int n = sizeof(edges)/sizeof(edges[0]);

DiaGraph diagraph(edges, n, N);

cout<<"Graph adjacency list "<<endl<<"(start\_vertex, end\_vertex, weight):"<<endl;

for (int i = 0; i < N; i++)

{

display\_AdjList(diagraph.head[i], i);

}

return 0;

}