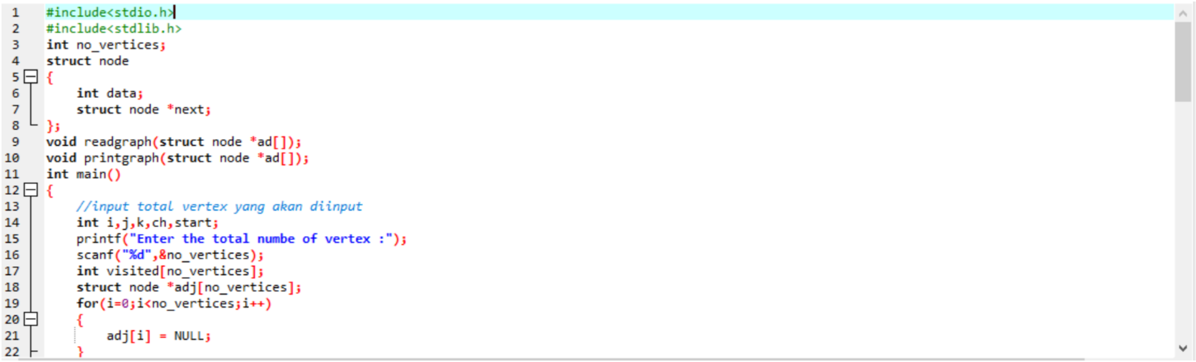
LAPORAN UAS STRUKTUR DATA

UNDIRECTED GRAPH

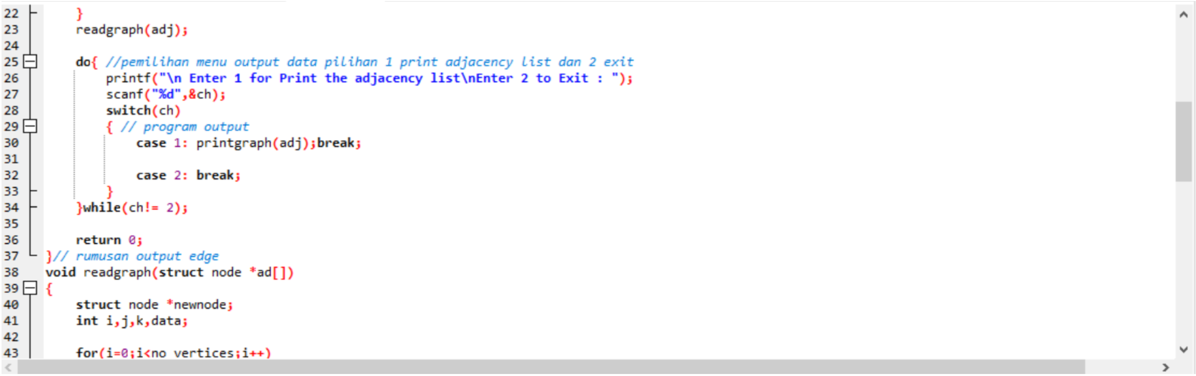
Nama : Putri Dwinatryska A.R.F

Kelas/ Nim :A / 21091397075

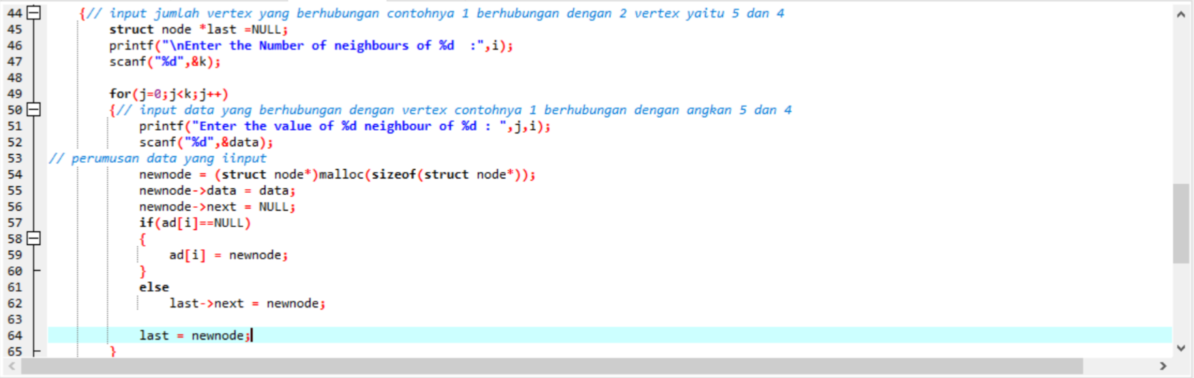
Membuat kodingan dan laporan sebuah algoritma yang membuat undirected graph menggunakan representasi adjacency list dengan input vertex dan edge.
Input :
1) int jumlah vertex yang ada dalam graph
2) (x,y,w) dengan x = vertex 1, y = vertex 2, w = weight
Output :
Satu per satu vertex, edge, dan weightnya
HASIL



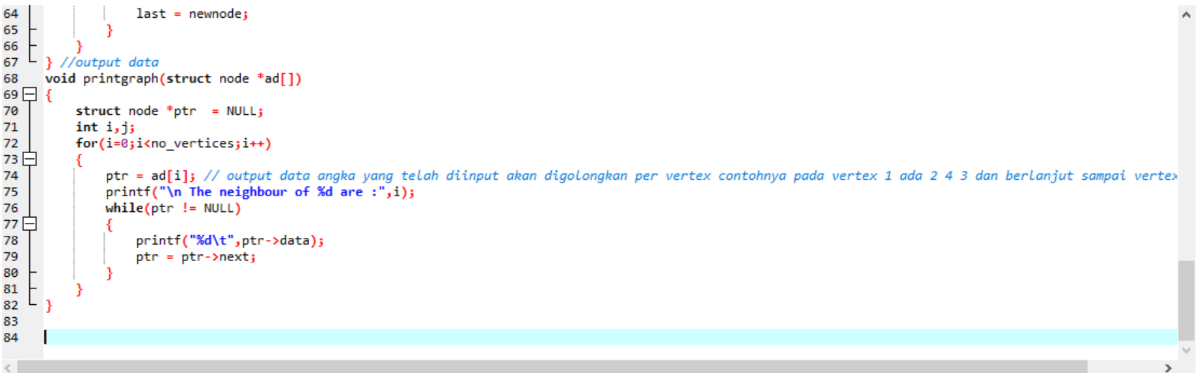
```c
#include<stdio.h>
#include<stdlib.h>
int no_vertices;
struct node
{
    int data;
    struct node *next;
};
void readgraph(struct node *ad[]);
void printgraph(struct node *ad[]);
int main()
{
    //input total vertex yang akan diinput
    int i,j,k,ch,start;
    printf("Enter the total numbe of vertex :");
    scanf("%d",&no_vertices);
    int visited[no_vertices];
    struct node *adj[no_vertices];
    for(i=0;i<no_vertices;i++)
    {
        adj[i] = NULL;
    }
```



```c
    }
    readgraph(adj);

    do{ //pemilihan menu output data pilihan 1 print adjacency list dan 2 exit
        printf("\n Enter 1 for Print the adjacency list\nEnter 2 to Exit : ");
        scanf("%d",&ch);
        switch(ch)
        { // program output
            case 1: printgraph(adj);break;

            case 2: break;
        }
    }while(ch!= 2);

    return 0;
}// rumusan output edge
void readgraph(struct node *ad[])
{
    struct node *newnode;
    int i,j,k,data;

    for(i=0;i<no_vertices;i++)
```

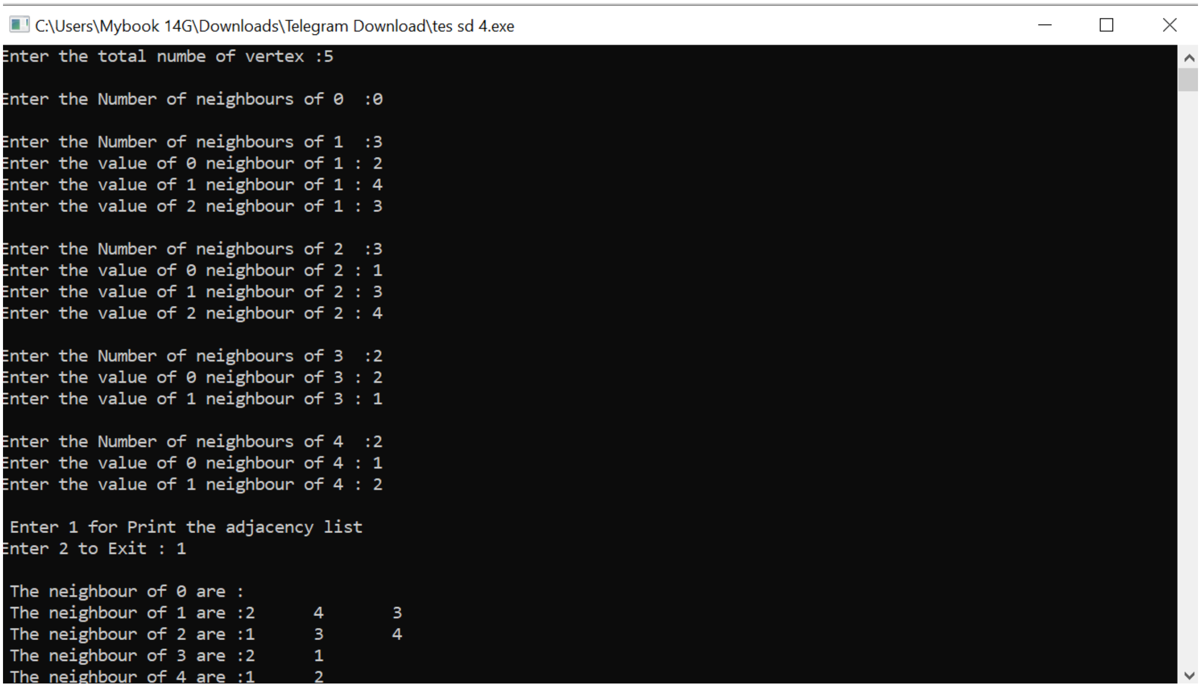


```c
    {// input jumlah vertex yang berhubungan contohnya 1 berhubungan dengan 2 vertex yaitu 5 dan 4
        struct node *last =NULL;
        printf("\nEnter the Number of neighbours of %d  :",i);
        scanf("%d",&k);

        for(j=0;j<k;j++)
        {// input data yang berhubungan dengan vertex contohnya 1 berhubungan dengan angkan 5 dan 4
            printf("Enter the value of %d neighbour of %d : ",j,i);
            scanf("%d",&data);
// perumusan data yang iinput
            newnode = (struct node*)malloc(sizeof(struct node*));
            newnode->data = data;
            newnode->next = NULL;
            if(ad[i]==NULL)
            {
                ad[i] = newnode;
            }
            else
                last->next = newnode;

            last = newnode;
        }
```



```c
            last = newnode;
        }
    }
} //output data
void printgraph(struct node *ad[])
{
    struct node *ptr  = NULL;
    int i,j;
    for(i=0;i<no_vertices;i++)
    {
        ptr = ad[i]; // output data angka yang telah diinput akan digolongkan per vertex contohnya pada vertex 1 ada 2 4 3 dan berlanjut sampai vertex
        printf("\n The neighbour of %d are :",i);
        while(ptr != NULL)
        {
            printf("%d\t",ptr->data);
            ptr = ptr->next;
        }
    }
}

```

A. Output Program



C:\Users\Mybook 14G\Downloads\Telegram Download\tes sd 4.exe

```
Enter the total numbe of vertex :5

Enter the Number of neighbours of 0  :0

Enter the Number of neighbours of 1  :3
Enter the value of 0 neighbour of 1 : 2
Enter the value of 1 neighbour of 1 : 4
Enter the value of 2 neighbour of 1 : 3

Enter the Number of neighbours of 2  :3
Enter the value of 0 neighbour of 2 : 1
Enter the value of 1 neighbour of 2 : 3
Enter the value of 2 neighbour of 2 : 4

Enter the Number of neighbours of 3  :2
Enter the value of 0 neighbour of 3 : 2
Enter the value of 1 neighbour of 3 : 1

Enter the Number of neighbours of 4  :2
Enter the value of 0 neighbour of 4 : 1
Enter the value of 1 neighbour of 4 : 2

 Enter 1 for Print the adjacency list
Enter 2 to Exit : 1

 The neighbour of 0 are :
 The neighbour of 1 are :2       4       3
 The neighbour of 2 are :1       3       4
 The neighbour of 3 are :2       1
 The neighbour of 4 are :1       2
```

Hasil dari program setisp input data yang dimasukan akan digolongkan di setiap jenis vertex seperti contoh hasil output diatas.