LAPORAN TUGAS STRUKTUR DATA

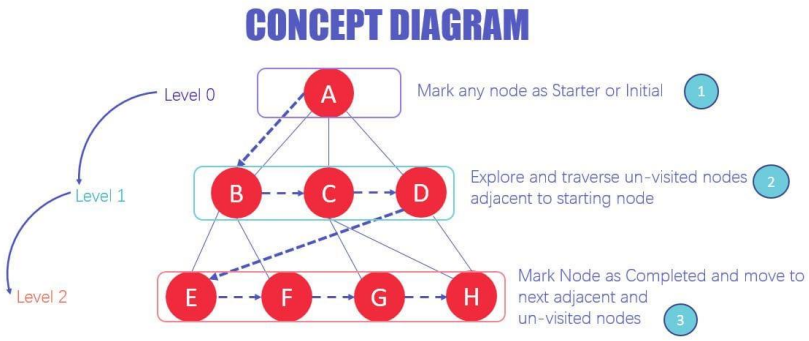
Kelompok 2

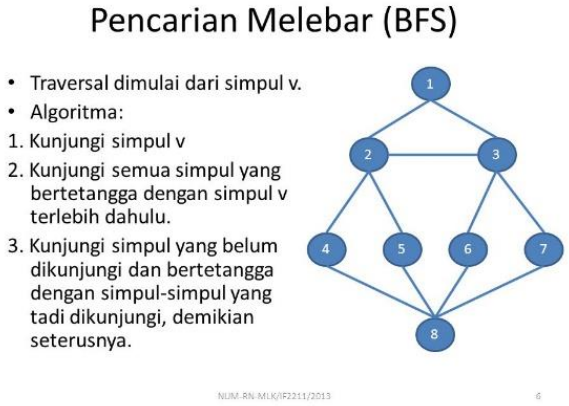
1) Handika Dio Pradana	(21091397021)
2) Muhammad Fikri Ramadhana	(21091397033)
3) Aditya Putra Pratama	(21091397043)
4) A’Ahmes Osama Firmansyah	(21091397051)
5) Irfan Rahmat Firmansyah	(21091397063)

A. Pendahuluan Laporan

Breadth-first search (BFS) adalah algoritma traversal yang menemukan jalur dari node tertentu ke semua node lain dalam grafik struktur data yang diberikan dengan terlebih dahulu memeriksa hanya penerus langsung dari node dan kemudian menggunakan penerusnya sebagai titik awal baru. Algoritme secara efisien mengunjungi dan menandai semua simpul kunci dalam grafik secara luas dan akurat. Algoritma ini memilih satu node (titik awal atau sumber) dalam grafik dan kemudian mengunjungi semua node yang berdekatan dengan node yang dipilih. Ingat, BFS mengakses node ini satu per satu. Setelah algoritme mengunjungi dan menandai simpul awal, ia bergerak menuju simpul terdekat yang belum dikunjungi dan menganalisisnya. Setelah dikunjungi, semua node ditandai. Iterasi ini berlanjut sampai semua node grafik telah berhasil dikunjungi dan ditandai.

B. Algoritma BFS



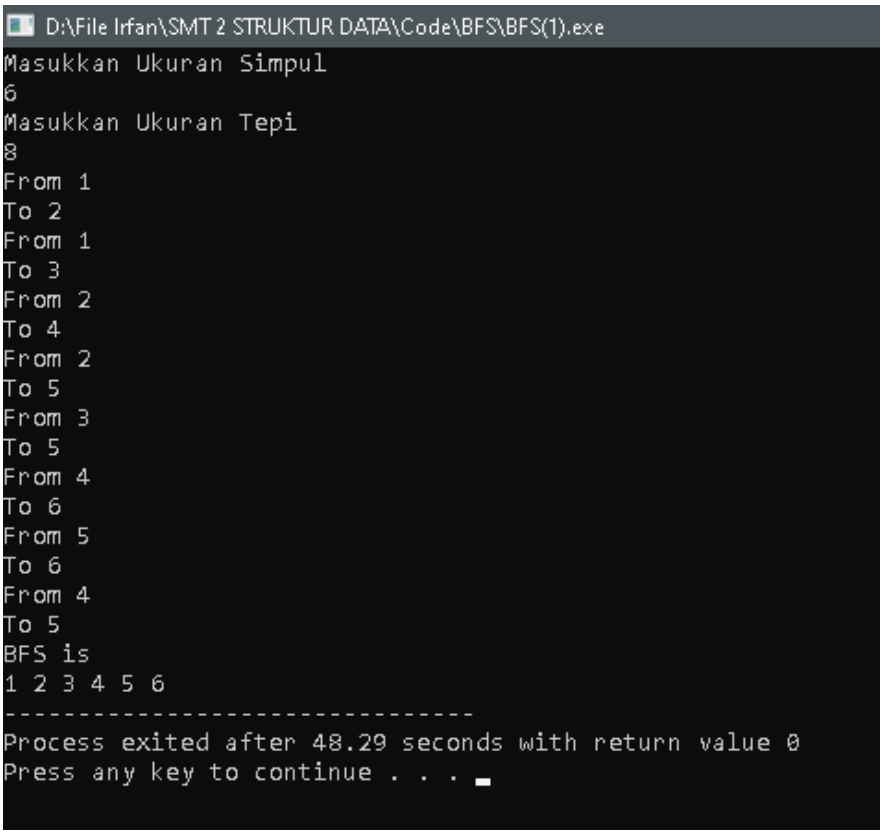


C. Coding Binary Search Tree menggunakan konsep BFS

Input :

```cpp
#include<iostream>
#include<vector>
#include<queue>
using namespace std;
//Untuk graf berarah , hapus kondisi kedua
void addEdge(vector<int>* graph,int u,int v)
{
    graph[u].push_back(v);
    graph[v].push_back(u);   // (2) Grafik Tidak Berarah
}
/*Function*/
int main()
{
    int n; //Simpul
    int e; //tepi
    int v,u; //from-to
    cout<<"Masukkan Ukuran Simpul"<<endl;
    cin>>n;
    cout<<"Masukkan Ukuran Tepi"<<endl;
    cin>>e;
    int copy;
    copy=n; // copy
    int a[n+1]={0}; //array untuk node yang dikunjungi atau tidak
    vector<int> graph[n+1]; //untuk mengurutkan simpul
    for(int i=1;i<=e;i++)
    {
        cout<<"From ";
        cin>>u;
        cout<<"To ";
        cin>>v;
        addEdge(graph,u,v); //fungsi untuk menambahkan tepi
    }
    vector<int>::iterator it;
    queue<int> q; //queue
    q.push(1); //awaln menyimpan elemen pertama dalam antrian
    a[1] = 1;
    int x;
    cout<<"BFS is"<<endl;
    while(!q.empty())
    {
        x = q.front();
        q.pop();
        cout<<x<<" ";
        for(it=graph[x].begin();it!=graph[x].end();it++)
        {
            if(a[*it]!=1)
            {
                q.push(*it);
                a[*it]=1;
            }
        }
    }
    return 0;
}
```

Output :


D:\File Irfan\SMT 2 STRUKTUR DATA\Code\BFS\BFS(1).exe
Masukkan Ukuran Simpul
6
Masukkan Ukuran Tepi
8
From 1
To 2
From 1
To 3
From 2
To 4
From 2
To 5
From 3
To 5
From 4
To 6
From 5
To 6
From 4
To 5
BFS is
1 2 3 4 5 6

Process exited after 48.29 seconds with return value 0
Press any key to continue . . .