

Nama : Bryan Yehuda Mannuel
NRP : 05311940000021

Tugas Pembuatan BFS

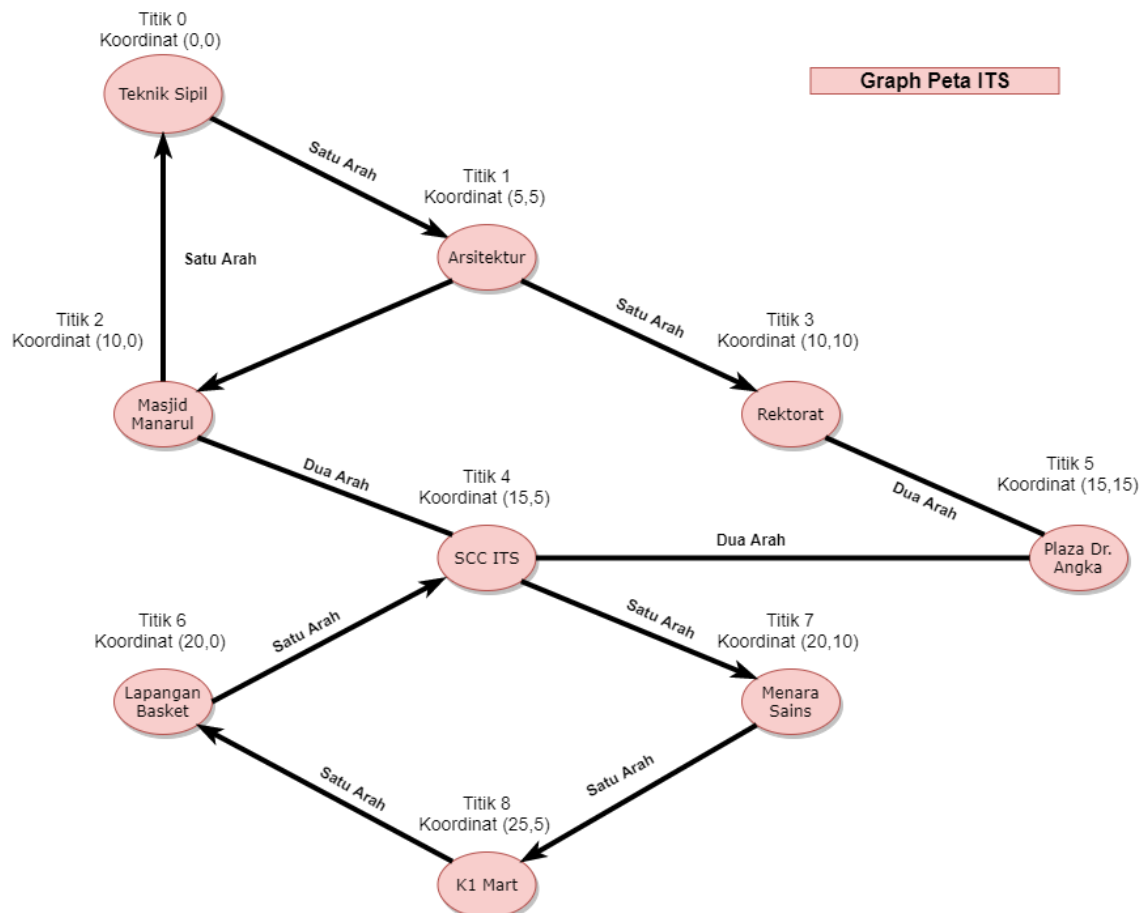
Mata Kuliah Struktur Data

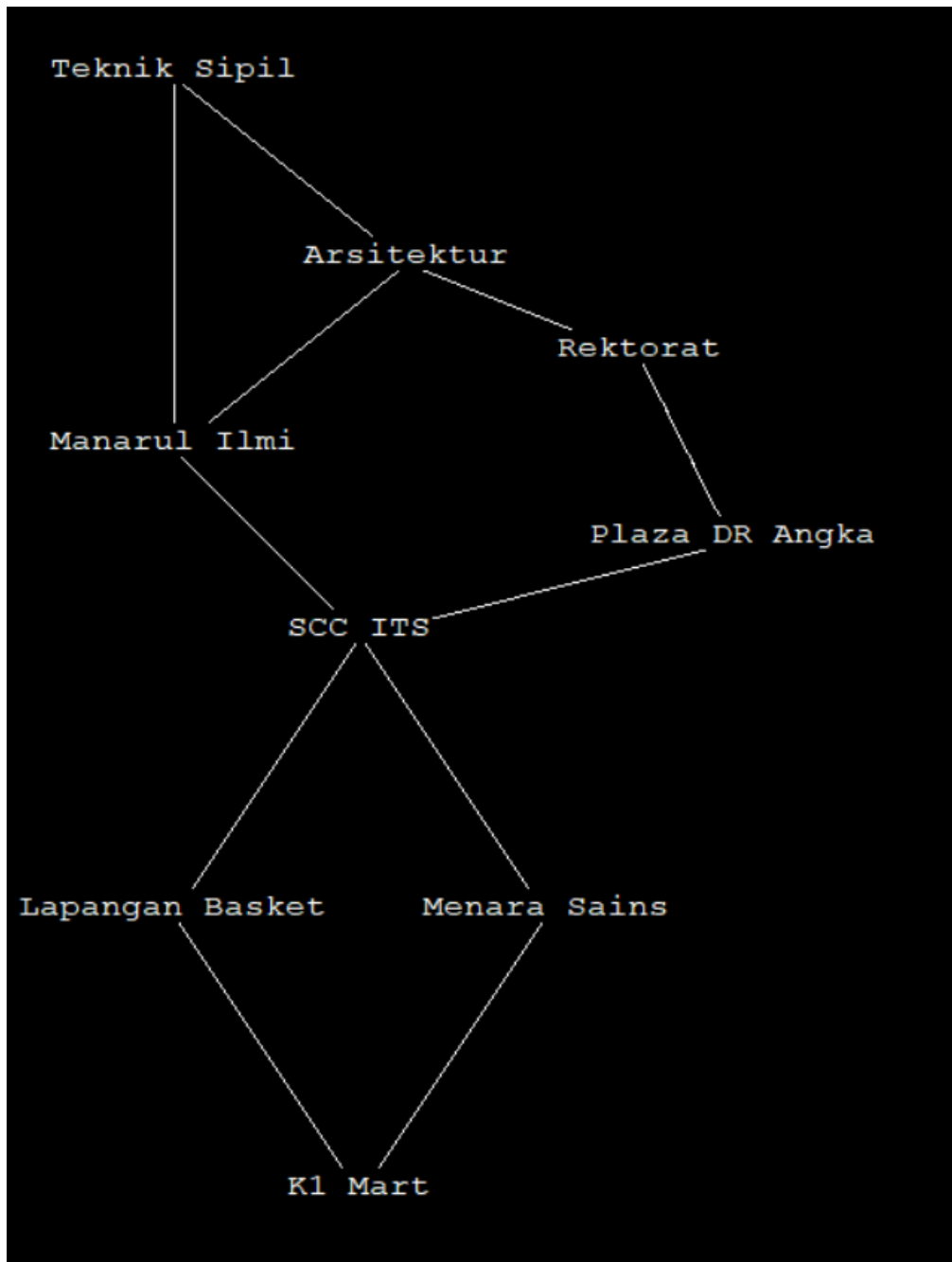
Apa itu BFS?

Breadth first search (atau BFS) adalah sebuah algoritma untuk melintasi atau mencari struktur data dalam sebuah pohon atau grafik. Algoritma ini dimulai di akar pohon (atau beberapa node acak dari sebuah grafik, kadang-kadang disebut sebagai 'tombol pencarian'), dan mengeksplorasi semua node tetangga pada kedalaman saat ini sebelum pindah ke simpul berikutnya pada tingkat yang lebih dalam.

Penugasan BFS berdasarkan Graph di Tugas Sebelumnya

Pada penugasan sebelumnya, saya diberikan tugas membuat sebuah graph Peta ITS yang memiliki 9 titik yang menggambarkan bagaimana keadaan nyata di daerah ITS. Graph tersebut sudah selesai dikerjakan menggunakan Bahasa Pemrograman C++ dan kita sudah berhasil menampilkan baik adjacency list dan bentuk graphnya. Berikut adalah gambar Graph yang dibuat secara manual dan dihasilkan oleh program.





Dan untuk penugasan pada kali ini saya diberikan tugas untuk membuat sebuah program yang bisa menghasilkan jalan terpendek (shortest path) dari graph tersebut. Program ini akan didasarkan pada algoritma BFS menggunakan Bahasa Pemrograman C++. Nantinya program ini akan menghasilkan sebuah output berupa jalan terpendek untuk melakukan Traversal (mengunjungi setiap node) yang ada pada graph tersebut. Untuk melakukan tugas ini, saya akan melakukan modifikasi kepada Source Code Graph pada penugasan sebelumnya dengan menambahkan fungsi BFS kedalamnya, dan berikut adalah fungsi BFS yang saya tambahkan agar bisa menghasilkan BFS:

```

//class untuk menampung graph
class Graph {
private:
    int V;
    list<int> *adj; //list hubungan antar titik
    list<Point> *coord; //list koordinat dan nama titik
public:
    Graph(int v);
    void addVertex(int v, int x, int y, string name);
    void addEdge(int v, int w);
    bool connected(int v, int w);
    void printList();
    void printEdge();
    void printPoint();
    void BFS(int s);
};

```

```

void Graph::BFS(int s)
{
    // Mark all the vertices as not visited
    bool *visited = new bool[V];
    for(int i = 0; i < V; i++)
        visited[i] = false;

    // Create a queue for BFS
    list<int> queue;

    // Mark the current node as visited and enqueue it
    visited[s] = true;
    queue.push_back(s);

    // 'i' will be used to get all adjacent
    // vertices of a vertex
    list<int>::iterator i;

    while(!queue.empty())
    {
        // Dequeue a vertex from queue and print it
        s = queue.front();
        cout << s << " ";
        queue.pop_front();

        // Get all adjacent vertices of the dequeued
        // vertex s. If a adjacent has not been visited,
        // then mark it visited and enqueue it
        for (i = adj[s].begin(); i != adj[s].end(); ++i)
        {
            if (!visited[*i])
            {
                visited[*i] = true;
                queue.push_back(*i);
            }
        }
    }
}

```

```

void showq(list<int> gq)
{
    list<int> g = gq;
    int i = 0;
    int lanjut = 1;
    if (lanjut == 1)
        return;

    while (!g.empty())
    {
        i++;
        cout << '\t' << (g.front() + 1);
        g.pop_front();
    }
    if (i == 0)
        cout << "\tempty";
    cout << '\n';
}

```

Potongan kode dibawah ini ditambahkan pada bagian Fungsi Main untuk memanggil fungsi BFS yang tadi sudah kita tambahkan diatas:

```

cout << "\n"
    << "Following is Breadth First Search | "
    << "(starting from vertex 2) \n";
g.BFS(2);
cout << "\n";

```

Hasil Run Code

Dan berikut adalah hasil Run Code dari program Graph yang sudah kita modifikasi dengan menambahkan fungsi BFS kedalamnya. Fungsi ini saya lakukan pemanggilan dari Vertex kedua atau Masjid Manarul dari Graph diatas. Dan kita bisa lihat bahwa program BFS sudah berhasil berjalan dengan menampilkan jalur terpendek untuk melakukan Traversal di Graph kita (2 1 0 4 3 5 7 6 8). Dengan demikian penugasan pembuatan BFS ini sudah berhasil dan selesai. Semoga laporan saya bisa berguna bagi kita semua. Terima kasih atas waktu dan perhatiannya.

```

Teknik Sipil -> Arsitektur -> Manarul Ilmi
Arsitektur -> Teknik Sipil -> Manarul Ilmi -> Rektorat
Manarul Ilmi -> Arsitektur -> Teknik Sipil -> SCC ITS -> SCC ITS
Rektorat -> Arsitektur -> Plaza DR Angka -> Plaza DR Angka
SCC ITS -> Manarul Ilmi -> Manarul Ilmi -> Plaza DR Angka -> Menara Sains -> Plaza DR Angka -> Lapangan Basket
Plaza DR Angka -> Rektorat -> SCC ITS -> Rektorat -> SCC ITS
Lapangan Basket -> SCC ITS -> K1 Mart
Menara Sains -> SCC ITS -> K1 Mart
K1 Mart -> Menara Sains -> Lapangan Basket

Following is Breadth First Search (starting from vertex 2)
2 1 0 4 3 5 7 6 8

Process returned 0 (0x0)   execution time : 1.451 s
Press any key to continue.

```