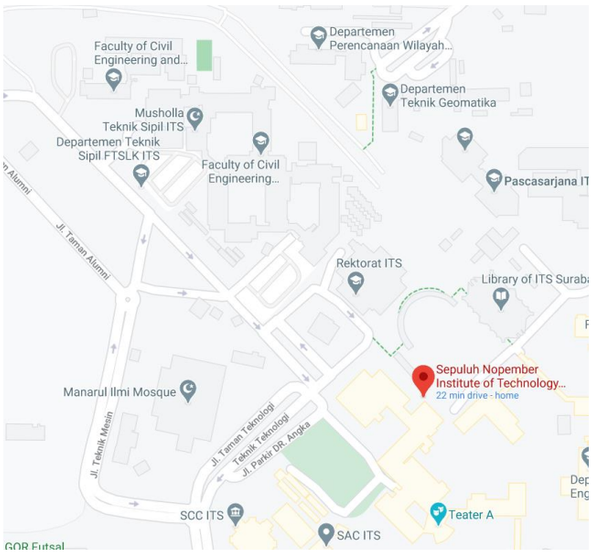
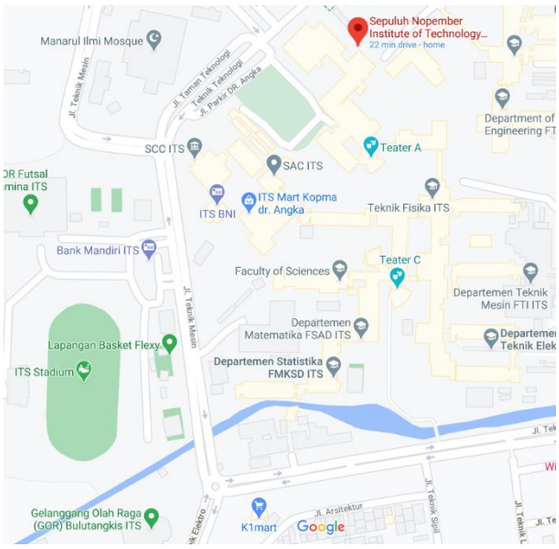
Nama : Bryan Yehuda Mannuel
NRP : 05311940000021

# Tugas Pembuatan Graph
# Mata Kuliah Struktur Data

## Graph Peta ITS

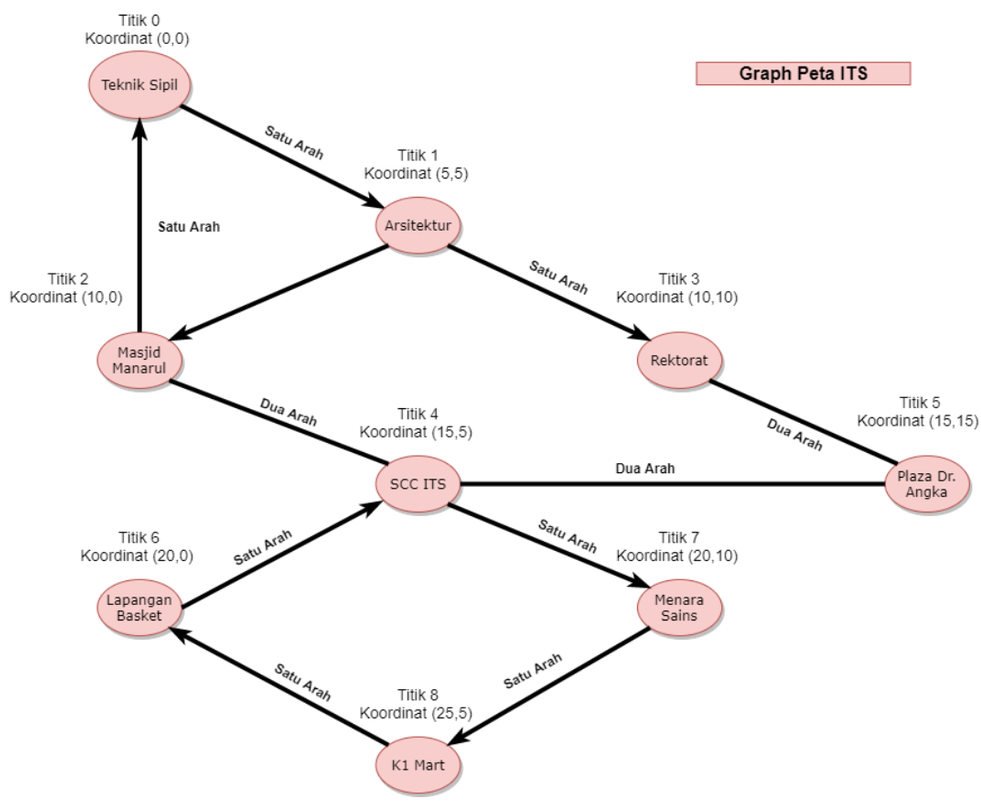



**Bagian Atas**



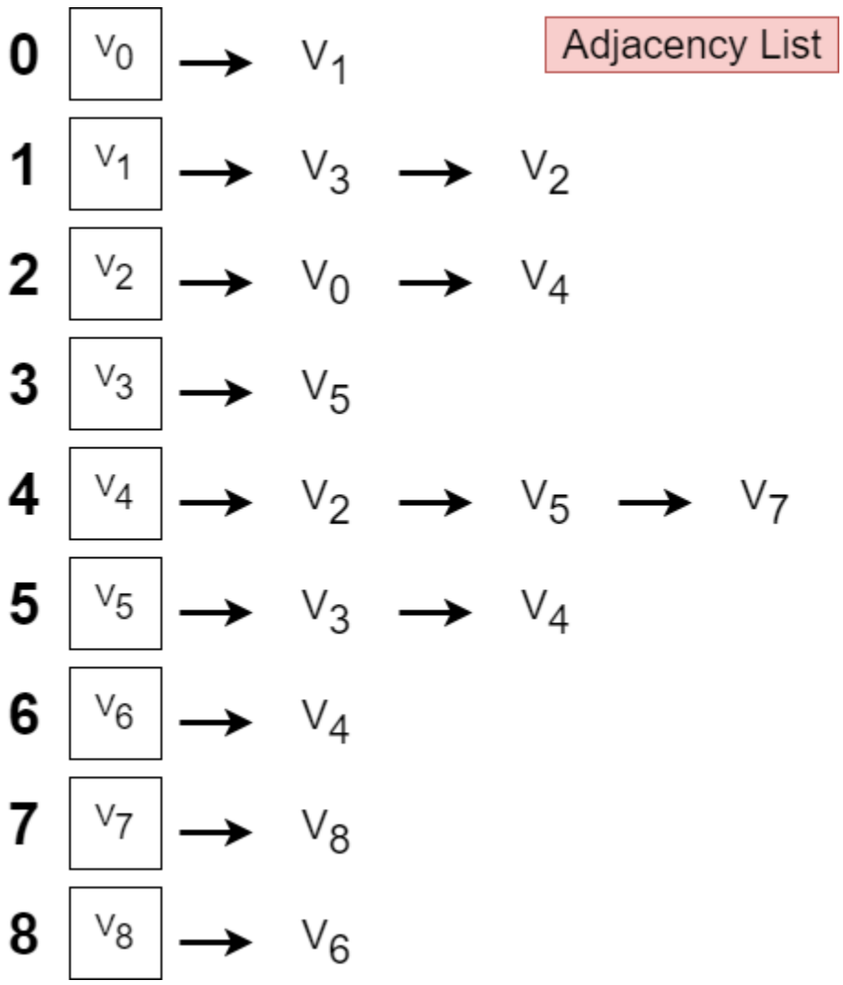

**Bagian Bawah**

Dari Peta ITS yang saya ambil dari Google Maps pada gambar diatas (ada 2 bagian yaitu bagian atas yang melingkupi Teknik Sipil hingga SCC, dan bagian bawah yang melingkupi SCC hingga K1 Mart) saya mengambil 9 lokasi yang menarik yaitu, Departemen Teknik Sipil, Departemen Arsitektur, Rektorat, Plaza Dr. Angka, Masjid Manarul Ilmi, SCC ITS, Menara Sains, Lapangan Basket, dan K1 Mart. Ketika ke Sembilan tempat tadi diubah menjadi sebuah Graph lengkap dengan arah jalan yang bisa ditempuh, maka akan tampil bentuk seperti di bawah ini:



## Adjacency List

Ketika Graph di atas diubah menjadi sebuah Adjacency List, maka akan berbentuk seperti pada adjacency list yang ada di bawah ini. Bisa dilihat bahwa titik 0 adalah Teknik Sipil, titik 1 adalah Arsitektur, titik 2 adalah Masjid Manarul, titik 3 adalah Rektorat, titik 4 adalah SCC ITS, titik 5 adalah Plaza Dr. Angka, titik 6 adalah Lapangan Basket, titik 7 adalah Menara Sains, dan titik 8 adalah K1 Mart.




## Program Pembuatan Graph

Kemudian, setelah kita selesai dalam pembuatan Graph maupun Adjacaceny List untuk Peta ITS diatas, sekarang sudah saatnya kita untuk masuk ke dalam pembuatan program yang bisa menerima vertex-vertex dan edge kemudian mengoutpurkan sebuah adjacency list dan graph ke layar output ketika dilakukan print. Program yang dibuat menggunakan Library Graphics.h agar bisa menghasilkan gambar Graph yang bagus dan bisa dioutpukan ke layar. Karena library Graphics.h adalah library yang cukup tua, maka kita harus melakukan sedikit pengaturan di Compiler kita (saya menggunakan compiler Code Blocks untuk mengerjakan tugas ini). Cara pengaturan tersebut adalah melakukan download GCC Compiler versi 32 bit, mengapa demikian? Karena library Graphics.h dibuat pada saat semua compiler dan OS yang ada masih berbentuk versi 32 bit dan kita tidak bisa menggunakan Compiler versi 64 bit (versi OS yang saya gunakan sekarang) untuk melakukan compilation terhadap library Graphics.h. Setelah melakukan download kepada GCC Compiler versi 32 bit, kita melakukan tambahan download library Graphics.h karena GCC Compiler secara default belum memasukkan library ini dan kita harus mendownloadnya dan memasukkannya secara manual. Dan setelah kita memasukkan library graphics.h maka sudah saatnya kita melakukan pembuatan program yang menampilkan adjacency list dan juga graph tersebut. Dan berikut adalah program yang kita buat:

```cpp
#include <iostream>
#include <list>
#include <cmath>
#include <graphics.h>
using namespace std;

class Point {
    private:
        int x, y;
        string name;
    public:
        Point();
        Point(int x, int y, string name);
        int getX();
        int getY();
        string getName();
};
//default
Point::Point() {
    this->x = -1;
    this->y = -1;
    this->name = "";
}
//constructor titik
Point::Point(int x, int y, string name) {
    this->x = x;
    this->y = y;
    this->name = name;
}
int Point::getX() {
    return x;
}
int Point::getY() {
    return y;
}
string Point::getName() {
    return name;
}
//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 addVertice(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();
};
//constructor graph
Graph::Graph(int V) {
    this->V = V;
    adj = new list<int>[V];
    coord = new list<Point>[V];
}
```

```
//menambahkan titik dan update ukuran matriks
void Graph::addVertice(int v,int x,int y, string name) {
    Point p(x, y, name);
    coord[v].push_back(p);

}
void Graph::addEdge(int v, int w) {
    adj[v].push_back(w);
    adj[w].push_back(v);
}
//mengecek apakah dua titik terhubung
bool Graph::connected(int v, int w) {
    list<int>::iterator j;

    for (j = adj[v].begin(); j != adj[v].end(); j++) {
        if (*j == w) return true;
    }

    for (j = adj[w].begin(); j != adj[w].end(); j++) {
        if (*j == v) return true;
    }

    return false;
}
//print hubungan dari setiap titik
void Graph::printList() {
    list<int>::iterator i;
    for (int v = 0; v < V; v++) {
        cout << coord[v].begin()->getName();
        for (i = adj[v].begin(); i != adj[v].end(); ++i) {
            cout << " -> " << coord[*i].begin()->getName();
        }
        cout << endl;
    }
}
void Graph::printEdge() {
    list<Point>::iterator i;
    for (int v = 0; v < V; v++) {
        for (int val = v + 1; val < V; val++) {
                if (!connected(v, val)) continue;

                i = coord[v].begin();
                int x1 = i->getX();
                int y1 = i->getY();

                i = coord[val].begin();
                int x2 = i->getX();
                int y2 = i->getY();

                line(x1, y1, x2, y2);
        }
    }
}
void Graph::printPoint() {
    list<Point>::iterator i;
    for (int v = 0; v < V; v++) {
        i = coord[v].begin();
        int point_x = i->getX();
        int point_y = i->getY();
        string name = i->getName();
        int n = name.length();
        char char_array[n + 1];
        strcpy(char_array, name.c_str());

        outtextxy(point_x, point_y, char_array);
    }
}
```

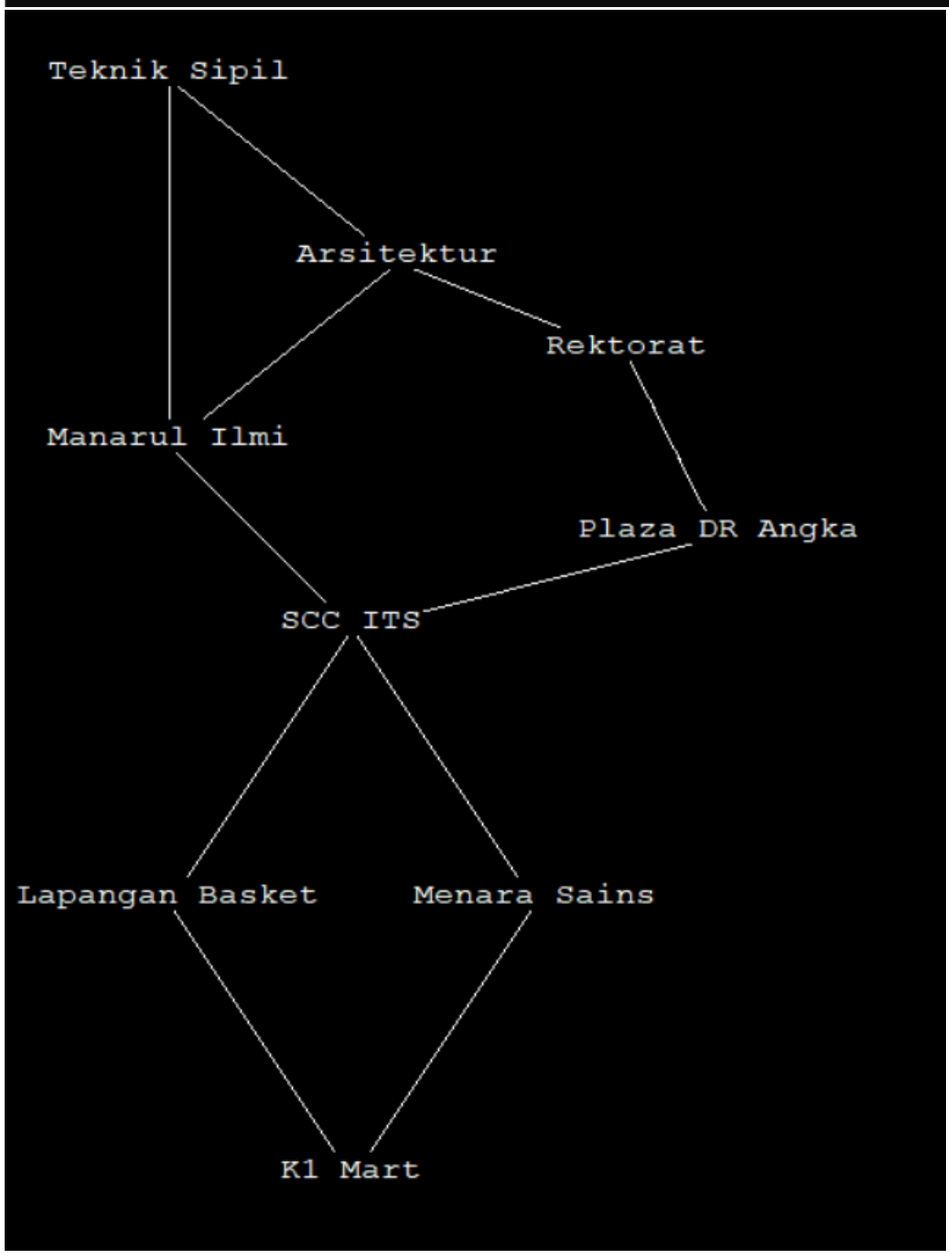
```
int main() {
    //membuat objek graph
    Graph g(9);
    int gd = DETECT, gm;

    // initgraph initializes the graphics system
    // by loading a graphics driver from disk
    initwindow(1920,1080);

    // font style
    int font = 8;

    // font direction
    int direction = 0;

    // font size
    int font_size = 1;

    // for setting text style
    settextstyle(font, direction, font_size);
    settextjustify(1, 1);
    //menambahkan titik dengan namanya
    g.addVertice(0, 100, 100, "Teknik Sipil");
    g.addVertice(1, 225, 200, "Arsitektur");
    g.addVertice(2, 100, 300, "Manarul Ilmi");
    g.addVertice(3, 350, 250, "Rektorat");
    g.addVertice(4, 200, 400, "SCC ITS");
    g.addVertice(5, 400, 350, "Plaza DR Angka");
    g.addVertice(6, 100, 550, "Lapangan Basket");
    g.addVertice(7, 300, 550, "Menara Sains");
    g.addVertice(8, 200, 700, "K1 Mart");
```

```
    //menambahkan hubungan titik
    g.addEdge(0, 1);
    g.addEdge(1, 2);
    g.addEdge(1, 3);
    g.addEdge(2, 0);
    g.addEdge(2, 4);
    g.addEdge(3, 5);
    g.addEdge(4, 2);
    g.addEdge(4, 5);
    g.addEdge(4, 7);
    g.addEdge(5, 3);
    g.addEdge(5, 4);
    g.addEdge(6, 4);
    g.addEdge(7, 8);
    g.addEdge(8, 6);
    //print hubungan dan graph
    g.printList();
    g.printEdge();
    g.printPoint();

    getch();

    // closegraph function closes the graphics
    // mode and deallocates all memory allocated
    // by graphics system .
    closegraph();
    return 0;
}
```

## Output Program

Dan berikut adalah output dari program ketika program di jalankan:

```
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

Process returned 0 (0x0)   execution time : 7.088 s
Press any key to continue.
```




Bisa dilihat dari output di atas bahwa program kita sudah berhasil mengoutputkan baik Adjacency List maupun gambar dari graph sesuai dengan yang sudah kita buat secara manual pada bagian atas dari laporan ini. Karena hasil output baik dari Adjacency List maupun dari Graph sudah sesuai dengan apa yang kita inginkan berarti kita sudah bisa menyebutkan bahwa program graph.cpp ini sudah bisa berjalan dengan baik dalam menjalankan fungsinya.

Demikian laporan tugas ini. Semoga apa yang ada dalam laporan ini bisa berguna bagi kita semua. Atas waktu dan perhatiannya saya ucapkan terima kasih.