**LAPORAN TUGAS STRUKTUR DATA**

**APLIKASI PENERAPAN *GRAPH***



Dosen Pembimbing:

Dr. Ricky Eka Putra, S.Kom., M.Kom.

Disusun oleh:

Cornelius Louis Nathan 23051204085

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**UNIVERSITAS NEGERI SURABAYA**

**2024**

# KATA PENGANTAR

Puji syukur saya haturkan kepada Tuhan Yang Maha Esa sehingga atas rahmat-Nya. Saya dapat menyelesaikan laporan tugas struktur data yang berjudul “Aplikasi Penerapan *Graph*” sebagai tugas struktur data tepat pada waktunya.

Adapun tujuan penulisan laporan ini adalah untuk mengetahui pengaplikasian *graph* dan penerapan *graph* dalam suatu aplikasi. Pembuatan laporan ini sekaligus menjadi tugas yang mengisi nilai dari Mata Kuliah Struktur Data.

Penyusunan laporan ini tidak lepas dari bantuan dan dukungan. Sehingga, kami menyampaikan terima kasih kepada pihak yang telah mendukung pembuatan laporan ini. Terutama kepada Bapak Ricky Eka Putra, S.Kom., M.Kom., selaku Dosen Pembimbing Mata Kuliah Struktur Data, atas petunjuk, didikan, dan arahan yang sangat membantu dalam penyusunan laporan ini.

Kami menyadari bahwasanya penulisan laporan ini jauh dari kata sempurna, sehingga kami menerima saran dan kritik yang membangun dari pembaca guna untuk perbaikan dan perkembangan di masa mendatang. Semoga penyusunan laporan ini juga berguna bagi para pembaca dalam memahami pengaplikasian *graph* dan penerapan *graph* dalam suatu aplikasi

Surabaya, 29 April 2024

Penulis

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Penggunaan komputer tidak dapat lepas dari pengaturan, pengorganisasian, dan pengelolaan data. Proses pengaturan, pengorganisasian, dan pengelolaan data disebut sebagai struktur data. Salah satu struktur data yang banyak digunakan akhir-akhir ini adalah *graph*. *Graph* sendiri adalah struktur data yang merepresentasikan hubungan antar objek, objek-objek tersebut digambarkan terdiri atas *vertex* (simpul) yang saling terhubung oleh *edge* (sisi). Penggunaan *graph* digunakan untuk mencari jalur terpendek antara dua titik dalam suatu rute, analisis jaringan sosial dalam sosial media,

Dalam kasus ini, andaikan sebuah perusahaan *travel* bernama HaalandHaaland menerima banyak turis mancanegara yang mengunjungi Surabaya. HaalandHalaand ingin membuat aplikasi yang mampu memberikan rute perjalanan ke tempat-tempat wisata yang ada di Surabaya, yaitu *Surabaya Travelling Planner*. Rute perjalanan akan dimulai dari suatu titik dan mengarahkan para turis menuju ke tempat wisata yang dekat dari tujuan semula mereka.

Selain itu, perusahaan *travel* tersebut juga sering menerima permintaan dari sekolah-sekolah SMA untuk melakukan *study tour* ke kampus-kampus yang ada di Surabaya. Maka HaalandHaaland ingin membuat aplikasi yang mampu memberikan rute perjalanan ke kampus-kampus tersebut. Aplikasi akan membedakan rute antara sekolah yang ingin berkunjung ke kampus negeri maupun ke kampus swasta.

## 1.2 Rumusan Masalah

1. Apa itu *graph* dan apa saja komponen yang dimiliki *graph*?
2. Bagaimana pengimplementasian *graph* dalam kasus dunia nyata?
3. Bagaimana *output* yang dihasilkan dari program tersebut?

## 1.3 Tujuan

1. Memahami *graph* dan komponen yang dimiliki *graph*.
2. Memahami pengimplementasian *graph* dalam kasus dunia nyata.
3. Memahami *output* yang dihasilkan dari program tersebut.

# BAB II

# KAJIAN PUSTAKA

## 2.1 Memahami *Graph* dan Komponen yang Dimiliki *Graph*

*Graph* adalah struktur data yang merepresentasikan objek-objek diskrit dan hubungan antar objek, objek-objek tersebut digambarkan terdiri atas *vertex* (simpul), yang digambarkan dengan titik, bulatan, atau noktah, yang saling terhubung oleh *edge* (sisi). Contoh penggunaan *graph* secara sederhana adalah peta jaringan jalan raya yang menghubungkan antar kota, jaringan instalasi listrik, pipa air, dalam suatu bangunan.

*Graph* adalah kumpulan dari simpul dan busur yang secara matematis dinyatakan sebagai berikut :

Yang dimana:

*G* = *Graph,*

*V = Vertex* (simpul, node, atau titik),

*E = Edge* (garis penghubung antar simpul),

Contoh penerapan *graph* yang sering digunakan adalah jaringan pertemanan pada sosial media

**Grafik 2.1** Jaringan Pertemanan dalam sosial media

Apabila *graph* tersebut digambarkan dengan notasi matematis, maka berikut adalah hasilnya :

*G =* Jaringan pertemanan dalam sosial media

*V*  = {Andy, Ben, Cole, Doug, Eddy, Frank}

*E =* {Andy, Ben}, {Ben, Cole}, {Ben, Eddy},{Cole, Doug}, {Cole, Frank}, {Doug, Eddy}, {Doug, Frank}.

*Graph* memiliki banyak kemungkinan yang bisa terjadi. Suatu *graph* bisa saja terdiri atas satu *vertex* saja. Suatu *graph* bisa saja memiliki *vertex* yang tak terhubung dengan *vertex* lainnya. Suatu *graph* belum tentu semua *vertex* nya terhubung dengan *edge*, atau bisa jadi suatu *graph* memiliki *vertex* yang semuanya saling terhubung.

Jenis *graph* juga beragam macamnya, terdapat *directed graph, undirected graph,* dan *weight graph*. *Directed graph* adalah *graph* yang setiap *edge* nya memiliki arah panah ke masing-masing *vertex*. *Undirected graph* adalah *graph* yang setiap *edge* nya tidak memiliki arah panah ke masing-masing *vertex*. Sehingga mampu berlaku dua arah. *Undirected graph* masih terbagi lagi menjadi *complete undirected graph* dan *connected undirected graph*. *Complete undirected graph* adalah kondisi dimana setiap *vertex* memiliki *edge* ke semua *vertex* yang ada di dalam *graph,* sehingga saling terhubung satu dengan yang lain. Sedangkan *connected undirected graph* adalah kondisi dimana setiap *vertex* setidaknya memiliki satu *edge* dengan *vertex* lainnya. Terakhir adalah *weight graph, weight graph* adalah *graph* yang setiap *edge* memiliki *weight* atau satuan, seperti jarak.

*Graph* pasti akan dilakukan penelusuran untuk membaca setiap data pada *vertex* maupun *edge*. Sehingga setiap *vertex* setidaknya harus terhubung satu dengan yang lain. Penelusuran akan dimulai dari satu *vertex*, lalu menjalar ke *vertex-vertex* lain. Metode penelusuran terbagi menjadi 2, yaitu *Depth First Search/DFS* (Pencarian secara mendalam) dan *Breadth First Search/BFS* (Pencarian secara melebar).

## 2.2 Pengimplementasian *Graph* dalam Kehidupan Sehari-hari

Andaikan sebuah perusahaan *travel* bernama HaalandHaaland menerima banyak turis mancanegara yang mengunjungi Surabaya. HaalandHalaand ingin membuat aplikasi yang mampu memberikan rute perjalanan ke tempat-tempat wisata yang ada di Surabaya, yaitu *Surabaya Travelling Planner*. Rute perjalanan akan dimulai dari suatu titik dan mengarahkan para turis menuju ke tempat wisata yang dekat dari tujuan semula mereka.

Untuk penerapan penelusuran *graph* dari aplikasi tersebut, maka dapat menggunakkan metode *BFS.* Pada metode *BFS*, ditetapkanlah satu *vertex* awal yang akan menjadi *start,* yang menjadi *level* 0. Lalu, *vertex* yang terhubung dengan *vertex* awal akan ditelusuri oleh program, menjadi *level* 1. Lalu, *vertex* yang terhubung dengan *vertex level* 1 akan ditelusuri oleh program, menjadi *level* 2. Seterusnya sampai semua *vertex* telah ditelusuri. Berikut adalah ilustrasi yang dapat menjelaskan *BFS:*

0

1

1

1

2

2

3

3

3

3

3

4

4

4

5

5

**Grafik 2.2** Grafik Metode *Breadth First Search*

Maka, output yang dihasilkan adalah : A B E F C I D G J M N H K L O P.

Yang dimana,

Anggota dari level 0 adalah A.

Anggota dari level 1 adalah B E F

Anggota dari level 2 adalah I C

Anggota dari level 3 adalah D G J M N

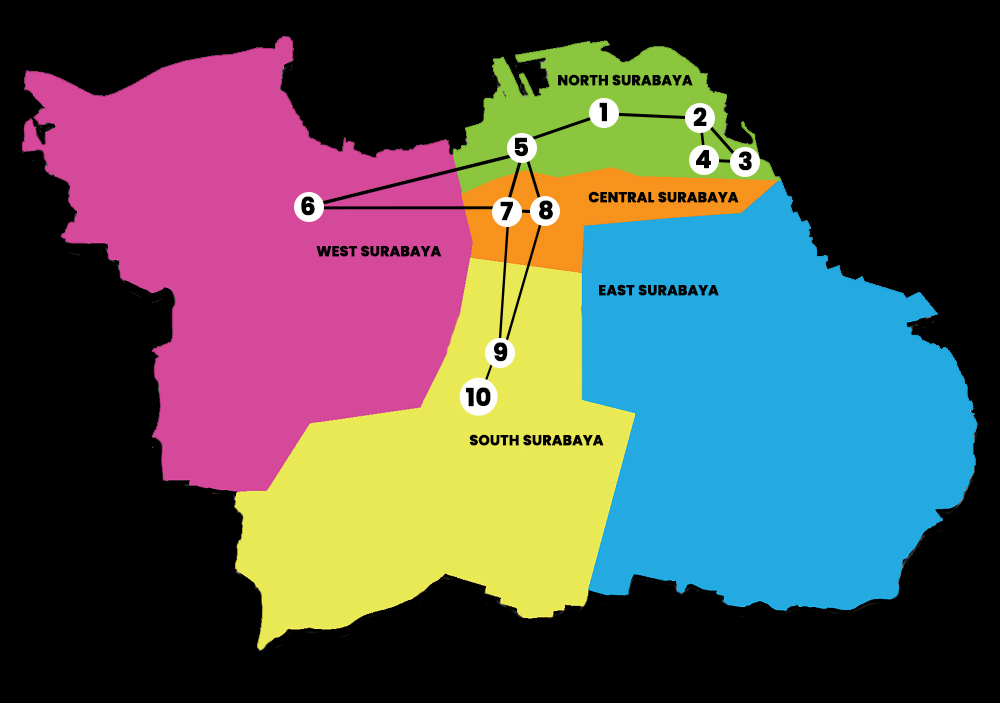
Anggota dari level 4 adalah H K L

Anggota dari level 5 adalah O P

Setelah melakukan perundingan, pertimbangan dan komunikasi dengan penanggung jawab dari tempat-tempat wisata yang ada di Surabaya. HaalandHaaland telah memutuskan mana destinasi wisata yang pantas untuk dikunjungi oleh para wisatawan, berikut adalah daftar listnya :

* + - 1. Arab Village (Kampoeng Arab), North Surabaya
      2. Bulak Village (Kampung Bulak), North Surabaya
      3. Sanggar Agung Temple (Klenteng Sanggar Agung), North Surabaya
      4. 4 Face Buddha Statue (Patung Buddha 4 Rupa), North Surabaya
      5. Museum House of Sampoerna, North Surabaya
      6. Food Junction Grand Pakuwon, West Surabaya
      7. Heroes Monument (Tugu Pahlawan), Central Surabaya
      8. Heroes Museum (Museum Pahlawan), Central Surabaya
      9. The Statue of Suro and Boyo (Patung Suro dan Boyo), South Surabaya
      10. Surabaya Zoo (Kebun Binatang Surabaya), South Surabaya

Yang apabila dipetakan dalam *graph* menjadi gambar berikut :



**Gambar 2.1** Grafik pemetaan destinasi wisata berdasarkan nomor di atas

Maka, dibuatlah *struct Graph* untuk menyimpan banyaknya *vertex* dan *unordered\_map* untuk menyimpan daftar ketetanggaan antar *vertex*. Serta membuat fungsi *void AddLine* untuk menghubungkan antar satu *vertex* dengan *vertex* lain. *Graph* dibuat berbasis *directed graph*.

struct Graph {

    int Destinations; // Jumlah destinasi (vertex)

    unordered\_map<string, vector<string>> Adj; // unordered\_map untuk merepresentasikan daftar ketetanggaan

};

void AddLine(Graph &Tour, string Start, string Finish){ // Untuk menambahkan edge ke dalam graf

    Tour.Adj[Start].push\_back(Finish); //terdapat garis penghubung antara Start dan Finish

}

Membuat fungsi *void BFS* dengan penerapan *queue* untuk melakukan pembacaan data *graph* serta menampilkan semua *vertex* yang dilalui*.*

void BFS(Graph &Tour, string Start) {

    unordered\_map<string, bool> Visited; // unordered\_map untuk menandai kota yang sudah dikunjungi

    queue<string> QueueTour; // Antrian untuk BFS

    Visited[Start] = true; // Tandai Start awal sebagai sudah dikunjungi

    QueueTour.push(Start); // Masukkan Start dalam Queue

    while (!QueueTour.empty()) {

        string StartTour = QueueTour.front(); // Ambil Start dari Queue

        QueueTour.pop();

        cout << StartTour << " ---> "; // Cetak destinasi yang dapat ditempuh

        // Iterasi destinasi

        for (string Finish : Tour.Adj[StartTour]) {

            // Jika destinasi lain belum dikunjungi, tandai sebagai sudah dikunjungi dan masukkan ke dalam antrian

            if (!Visited[Finish]) {

                Visited[Finish] = true;

                QueueTour.push(Finish);

            }

        }

    }

}

Setelah membuat semua fungsi tersebut, maka waktunya untuk membuat program utama dari *Surabaya Destination Planner,* sehingga dilakukan pendeklarasian *vertex* dan menghubungkan antara *vertex* satu dengan *vertex* yang lain dengan garis. Semakin dekat suatu *vertex* dengan *vertex* lain, maka hubungan tersebut semakin menjadi prioritas atau tingkatannya semakin di atas.

int main(){

    Graph SurabayaTourism;

    SurabayaTourism.Destinations = 10;

    AddLine(SurabayaTourism,"Arab Village", "Museum House of Sampoerna");

    AddLine(SurabayaTourism,"Arab Village", "Bulak Village");

    AddLine(SurabayaTourism,"Bulak Village", "4 Face Buddha Statue");

    AddLine(SurabayaTourism,"Bulak Village", "Sanggar Agung Temple");

    AddLine(SurabayaTourism,"Bulak Village", "Arab Village");

    AddLine(SurabayaTourism,"Sanggar Agung Temple", "4 Face Buddha Statue");

    AddLine(SurabayaTourism,"Sanggar Agung Temple", "Bulak Village");

    AddLine(SurabayaTourism,"4 Face Buddha Statue", "Sanggar Agung Temple");

    AddLine(SurabayaTourism,"4 Face Buddha Statue", "Bulak Village");

    AddLine(SurabayaTourism,"Museum House of Sampoerna","Arab Village");

    AddLine(SurabayaTourism,"Museum House of Sampoerna","Heroes Monument");

    AddLine(SurabayaTourism,"Museum House of Sampoerna","Heroes Museum");

    AddLine(SurabayaTourism,"Museum House of Sampoerna","Food Junction Grand Pakuwon");

    AddLine(SurabayaTourism,"Food Junction Grand Pakuwon","Museum House of Sampoerna");

    AddLine(SurabayaTourism,"Food Junction Grand Pakuwon","Heroes Monument");

    AddLine(SurabayaTourism,"Heroes Monument", "Heroes Museum");

    AddLine(SurabayaTourism,"Heroes Monument", "Museum House of Sampoerna");

    AddLine(SurabayaTourism,"Heroes Monument", "Food Junction Grand Pakuwon");

    AddLine(SurabayaTourism,"Heroes Monument", "The Statue of Suro and Boyo");

    AddLine(SurabayaTourism,"Heroes Museum", "Heroes Monument");

    AddLine(SurabayaTourism,"Heroes Museum", "The Statue of Suro and Boyo");

    AddLine(SurabayaTourism,"Heroes Museum", "Museum House of Sampoerna");

    AddLine(SurabayaTourism,"The Statue of Suro and Boyo", "Surabaya Zoo");

    AddLine(SurabayaTourism,"The Statue of Suro and Boyo", "Heroes Monument");

    AddLine(SurabayaTourism,"The Statue of Suro and Boyo", "Heroes Museum");

    AddLine(SurabayaTourism,"Surabaya Zoo", "The Statue of Suro and Boyo");

Pengguna dapat memilih destinasi yang diinginkan dari ke sepuluh pilihan tersebut. Destinasi dipilih dengan mengetikkan *string* nama tempat yang ingin dikunjungi.

string Destination;

    system("cls");

    cout << "========= Surabaya Destination Planner =========\n";

    cout << "Where is your destination?\n\n";

    cout << "1. Arab Village (Kampung Arab), North Surabaya\n";

    cout << "2. Bulak Village (Kampung Bulak), North Surabaya\n";

    cout << "3. Sanggar Agung Temple (Klenteng Sanggar Agung), North Surabaya\n";

    cout << "4. 4 Face Buddha Statue (Patung Buddha 4 Rupa), North Surabaya\n";

    cout << "5. Museum House of Sampoerna, North Surabaya\n";

    cout << "6. Food Junction Grand Pakuwon, West Surabaya\n";

    cout << "7. Heroes Monument (Tugu Pahlawan), Central Surabaya\n";

    cout << "8. Heroes Museum (Museum Pahlawan), Central Surabaya\n";

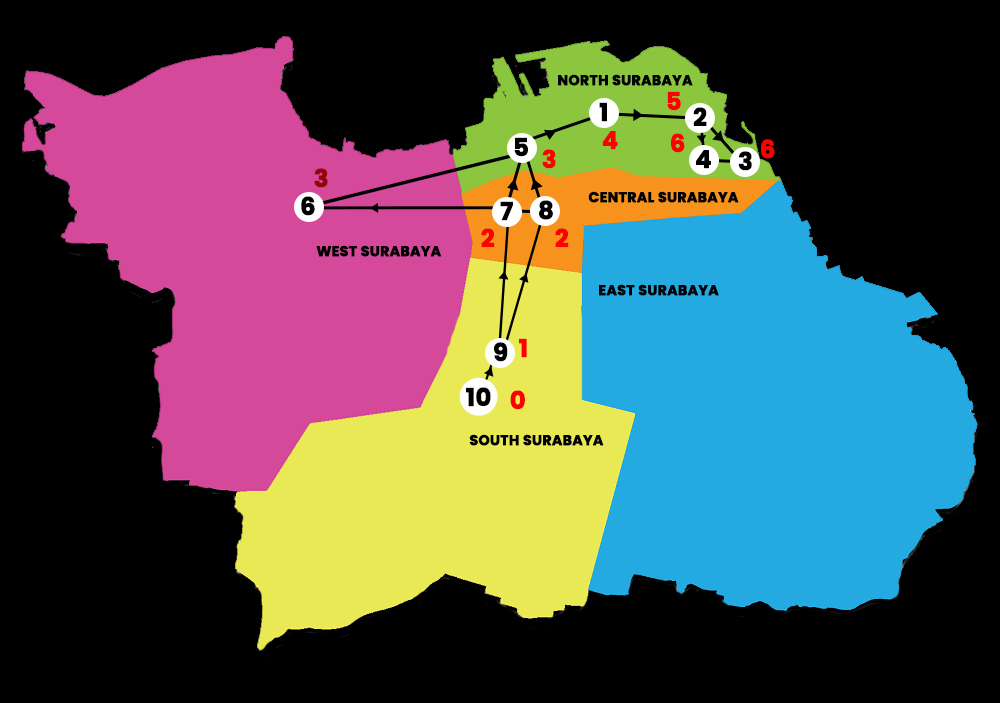
    cout << "9. The Statue of Suro and Boyo (Patung Suro dan Boyo), South Surabaya\n";

    cout << "10. Surabaya Zoo (Kebun Binatang Surabaya), South Surabaya\n\n";

    cout << "> ";

getline(cin, Destination);

Setelah memilih destinasi yang diinginkan, maka program akan menetapkan destinasi tersebut menjadi *vertex* awal atau level 0. Lalu dari *vertex 0*, menjelajah ke *vertex* lain yang terhubung langsung ke *vertex 1*. Sebagai contoh, apabila kita ingin destinasi awal kita adalah *Surabaya Zoo*, maka diharapkan proses *BFS* nya seperti berikut :



**Gambar 2.2** Grafik proses penelusuran *BFS* dari *Surabaya Zoo* sebagai *vertex* awal

*Surabaya Zoo* sebagai *vertex* level 0, kemudian program menelusuri *vertex* level 1, yaitu *The Statue of Suro and Boyo.* Program menelusuri *vertex* level 2, yaitu *Heroes Monument* dan *Heroes Museum*. Program menelusuri *vertex* level 3, yaitu *Food Junction Grand Pakuwon* dan *Museum House of Sampurna*, lanjut ke *Arab Village*. Pada *vertex 5*, terdapat *Bulak Village*. Pada *vertex 6,* destinasi terakhir sampai ke *Sanggar Agung Temple* ataupun *4 Face Buddha Statue*.

Beralih ke kasus selanjutnya, perusahaan *travel* HaalandHaaland juga sering menerima permintaan dari sekolah-sekolah SMA untuk melakukan *study tour* ke kampus-kampus yang ada di Surabaya. Maka HaalandHaaland ingin membuat aplikasi yang mampu memberikan rute perjalanan ke kampus-kampus tersebut. Aplikasi akan membedakan rute antara sekolah yang ingin berkunjung ke kampus negeri maupun ke kampus swasta.

Untuk penerapan penelusuran *graph* dari aplikasi tersebut, maka dapat menggunakkan metode *DFS.* Pada metode *DFS*, ditetapkanlah satu *vertex* awal yang akan menjadi *start*. Lalu, dipilih salah satu *vertex* yang terhubung langsung dengan *vertex* awal akan ditelusuri oleh program. Program akan menelusuri *vertex* tersebut secara terus menerus sampai *vertex* paling terakhir. Apabila sudah tidak ada *vertex* lain, maka program akan kembali ke *vertex* sebelumnya dan mencari *vertex* yang belum ditelusuri . Seterusnya sampai semua *vertex* yang terhubung telah selesai ditelusuri. Berikut adalah ilustrasi yang dapat menjelaskan *DFS.*



**Gambar 2.3** Grafik metode *Depth First Search*

Maka, output yang dihasilkan adalah : A B F H C D G I E

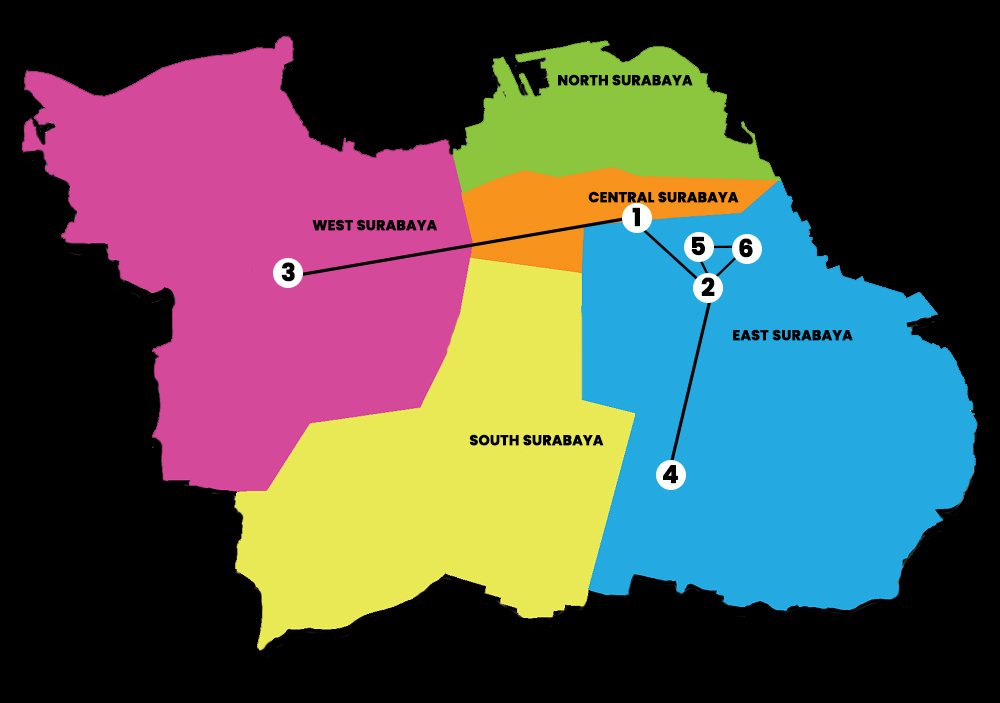
Yang dimana, *Start vertex* awal adalah A, lalu program memilih untuk menelusuri *vertex* B, walaupun sebenarnya dapat memilih untuk menelusuri *vertex* C, D, ataupun E. Dari *vertex B,* menelusuri ke *vertex* F dan *vertex* H. Karena sudah tidak ada lagi *vertex* yang bisa ditelusuri, maka program mundur ke *vertex* F, B*,* dan kembali ke A. Lalu program memilih *vertex* lain yang bisa ditelusuri, yaitu *vertex* C. Karena sudah tidak ada lagi *vertex* yang bisa ditelusuri, maka program mundur ke *vertex* A. Begitu seterusnya sampai *vertex* paling terakhir, yaitu *vertex* E sudah ditelusuri oleh program.

Setelah melakukan perundingan dan kerjasama dengan universitas-universitas yang ada di Surabaya. HaalandHaaland telah memutuskan mana universitas-universitas yang dapat dikunjungi oleh sekolah-sekolah untuk melakukan *study tour*, berikut adalah daftar listnya :

Daftar universitas negeri di Surabaya yang telah melakukan kerjasama dengan HaalandHaaland :

1. Universitas Airlangga (UNAIR)
2. Institut Teknologi Sepuluh November (ITS)
3. Universitas Negeri Surabaya (UNESA)
4. Universitas Pembangunan Negeri Veteran Jawa Timur (UPN)
5. Politeknik Elekronika Negeri Surabaya (PENS)
6. Politeknik Perkapalan Negeri Surabaya (PPNS)

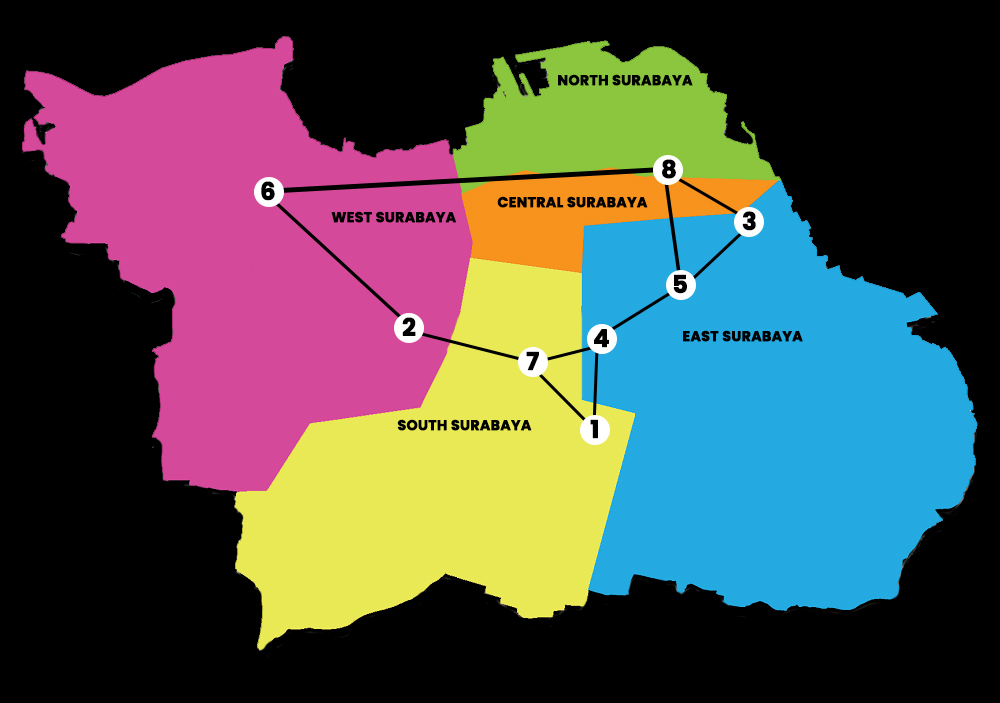
Yang apabila dipetakan dalam *graph* menjadi gambar berikut :



**Gambar 2.4** Grafik pemetaan universitas negeri di Surabaya berdasarkan penomoran di atas

Daftar universitas swasta di Surabaya yang telah melakukan kerjasama dengan HaalandHaaland :

1. Universitas Surabaya (UBAYA)
2. Universitas Wijaya Kusuma Surabaya (UWKS)
3. Universitas Katolik Widya Mandala (UWM)
4. Universitas 17 Agustus 1945 (UNTAG)
5. Universitas Hang Tuah (UHT)
6. Universitas Ciputra (UC)
7. Universitas Islam Negeri Sunan Ampel (UINSA)
8. Universitas Muhammadiyah Surabaya (UMS)



**Gambar 2.5** Grafik pemetaan universitas swasta di Surabaya berdasarkan penomoran di atas

Maka, dibuatlah *struct Graph* untuk menyimpan banyaknya *vertex* dan *unordered\_map* untuk menyimpan daftar ketetanggaan antar *vertex*. Serta membuat fungsi *void AddLine* untuk menghubungkan antar satu *vertex* dengan *vertex* lain. *Graph* dibuat berbasis *directed graph*.

// Struktur Graf

struct Graph {

    int Destinations; // Jumlah destinasi (vertices)

    unordered\_map<string, vector<string>> Adj; // unordered\_map untuk merepresentasikan daftar ketetanggaan

};

// Menambahkan edge ke dalam graf

void AddLine(Graph &Tour, string Start, string Finish) {

    Tour.Adj[Start].push\_back(Finish); // Terdapat garis penghubung antara Start dan Finish

}

Membuat fungsi *void DFS* dengan penerapan *stack* untuk melakukan pembacaan data *graph* serta menampilkan semua *vertex* yang dilalui

// Fungsi untuk melakukan pencarian secara mendalam (DFS) pada graf menggunakan stack

void DFS(const Graph &Tour, const string &Start) {

    // Membuat unordered\_map untuk melacak node yang telah dikunjungi

    unordered\_map<string, bool> Visited;

    // Inisialisasi semua node sebagai belum dikunjungi

    for (const auto &Pair : Tour.Adj) {

        Visited[Pair.first] = false;

    }

    // Buat stack untuk melacak node yang akan dikunjungi

    stack<string> ToVisit;

    // Masukkan node awal ke dalam stack

    ToVisit.push(Start);

    // Selama stack tidak kosong

    while (!ToVisit.empty()) {

        // Ambil node dari stack

        string Current = ToVisit.top();

        ToVisit.pop();

        // Jika node belum dikunjungi

        if (!Visited[Current]) {

            // Tandai node sebagai telah dikunjungi

            Visited[Current] = true;

            cout << Current << " ---> "; // Mengeluarkan node saat ini

            // Iterasi melalui semua tetangga dari node saat ini

            for (const string &Next : Tour.Adj.at(Current)) {

                // Jika tetangga belum dikunjungi, masukkan ke dalam stack

                if (!Visited[Next]) {

                    ToVisit.push(Next);

                }

            }

        }

    }

}

Setelah membuat semua fungsi tersebut, maka waktunya untuk membuat program utama dari *Surabaya Study Tour Planner.* Pengguna dapat memilih jenis kampus yang akan dituju, apakah ingin melakukan *study tour* ke perguruan tinggi negeri, perguruan tinggi swasta, ataupun secara bebas.

int main() {

    Graph StudyTour;

    StudyTour.Destinations = 14;

    int Choice;

    system("cls");

    cout << "========= Surabaya Study Tour Planner =========\n";

    cout << "Pilih jenis kampus yang ingin dituju?\n\n";

    cout << "1. Perguruan Tinggi Negeri\n";

    cout << "2. Perguruan Tinggi Swasta\n";

    cout << "3. Bebas\n";

    cout << "> ";

    cin >> Choice;

    cin.ignore();

    string Destination;

Pada menu 1. Perguruan Tinggi Negeri, dilakukan pendeklarasian *vertex* dan menghubungkan antara *vertex* satu dengan *vertex* yang lain dengan garis. Semakin dekat suatu *vertex* dengan *vertex* lain, maka hubungan tersebut semakin menjadi prioritas atau tingkatannya semakin di atas. Pengguna juga dapat menentukan universitas mana yang ingin pertama kali dituju dengan menginputkan singkatan dari universitas tersebut. Setelah itu, program akan melakukan pemanggilan fungsi *DFS* terhadap tujuan awal pengguna.

 switch (Choice){

    case 1:{

        AddLine(StudyTour,"UNAIR", "ITS");

        AddLine(StudyTour,"UNAIR", "UNESA");

        AddLine(StudyTour,"ITS", "PENS");

        AddLine(StudyTour,"ITS", "PPNS");

        AddLine(StudyTour,"ITS", "UPN");

        AddLine(StudyTour,"UNESA", "UNAIR");

        AddLine(StudyTour,"UPN", "ITS");

        AddLine(StudyTour,"PENS", "ITS");

        AddLine(StudyTour,"PENS", "PPNS");

        AddLine(StudyTour,"PPNS", "ITS");

        AddLine(StudyTour,"PPNS", "PENS");

        cout << "Tujuan pertama yang ingin dituju?\n\n";

        cout << "1. Universitas Airlangga (UNAIR)\n";

        cout << "2. Institut Teknologi Sepuluh November (ITS)\n";

        cout << "3. Universitas Negeri Surabaya (UNESA)\n";

        cout << "4. Universitas Pembangunan Negeri Veteran Jawa Timur (UPN)\n";

        cout << "5. Politeknik Elekronika Negeri Surabaya (PENS)\n";

        cout << "6. Politeknik Perkapalan Negeri Surabaya (PPNS)\n\n";

        cout << "(Masukkan singkatan kampus yang ingin dituju, cth : UNESA)\n";

        cout << "> ";

        getline(cin,Destination);

        DFS(StudyTour,Destination);

        cout << "Finish";

    }

Pada menu 2. Perguruan Tinggi Swasta, dilakukan pendeklarasian *vertex* dan menghubungkan antara *vertex* satu dengan *vertex* yang lain dengan garis. Semakin dekat suatu *vertex* dengan *vertex* lain, maka hubungan tersebut semakin menjadi prioritas atau tingkatannya semakin di atas. Pengguna juga dapat menentukan universitas mana yang ingin pertama kali dituju dengan menginputkan singkatan dari universitas tersebut. Setelah itu, program akan melakukan pemanggilan fungsi *DFS* terhadap tujuan awal pengguna.

    case 2:{

        AddLine(StudyTour,"UBAYA", "UNTAG");

        AddLine(StudyTour,"UBAYA", "UINSA");

        AddLine(StudyTour,"UWKS", "UINSA");

        AddLine(StudyTour,"UWKS", "UC");

        AddLine(StudyTour,"UWM", "UHT");

        AddLine(StudyTour,"UWM", "UMS");

        AddLine(StudyTour,"UNTAG", "UHT");

        AddLine(StudyTour,"UNTAG", "UBAYA");

        AddLine(StudyTour,"UNTAG", "UINSA");

        AddLine(StudyTour,"UHT", "UWM");

        AddLine(StudyTour,"UHT", "UNTAG");

        AddLine(StudyTour,"UHT", "UMS");

        AddLine(StudyTour,"UC", "UWKS");

        AddLine(StudyTour,"UC", "UMS");

        AddLine(StudyTour,"UMS", "UWM");

        AddLine(StudyTour,"UMS", "UHT");

        AddLine(StudyTour,"UMS", "UC");

        cout << "Tujuan pertama yang ingin dituju?\n\n";

        cout << "1. Universitas Surabaya (UBAYA)\n";

        cout << "2. Universitas Wijaya Kusuma Surabaya (UWKS)\n";

        cout << "3. Universitas Katolik Widya Mandala (UWM)\n";

        cout << "4. Universitas 17 Agustus 1945 (UNTAG)\n";

        cout << "5. Universitas Hang Tuah (UHT)\n";

        cout << "6. Universitas Ciputra (UC)\n";

        cout << "7. Universitas Islam Negeri Sunan Ampel (UINSA)\n";

        cout << "8. Universitas Muhammadiyah Surabaya (UMS)\n";

        cout << "(Masukkan singkatan kampus yang ingin dituju, cth : UBAYA)\n";

        cout << "> ";

        getline(cin,Destination);

        DFS(StudyTour,Destination);

        cout << "Finish";

    }

    break;

Pada menu 3. Bebas, dilakukan pendeklarasian *vertex* dan menghubungkan antara *vertex* satu dengan *vertex* yang lain dengan garis. Hasil diskusi yang dilakukan oleh HaalandHaaland telah menetapkan rute sebagai berikut

1. Universitas Surabaya (UBAYA)
2. Universitas Islam Negeri Surabaya (UINSA)
3. Universitas Negeri Surabaya (UNESA)
4. Universitas Ciputra (UC)
5. Universitas Airlangga (UNAIR)
6. Institut Teknologi Sepuluh November (ITS)

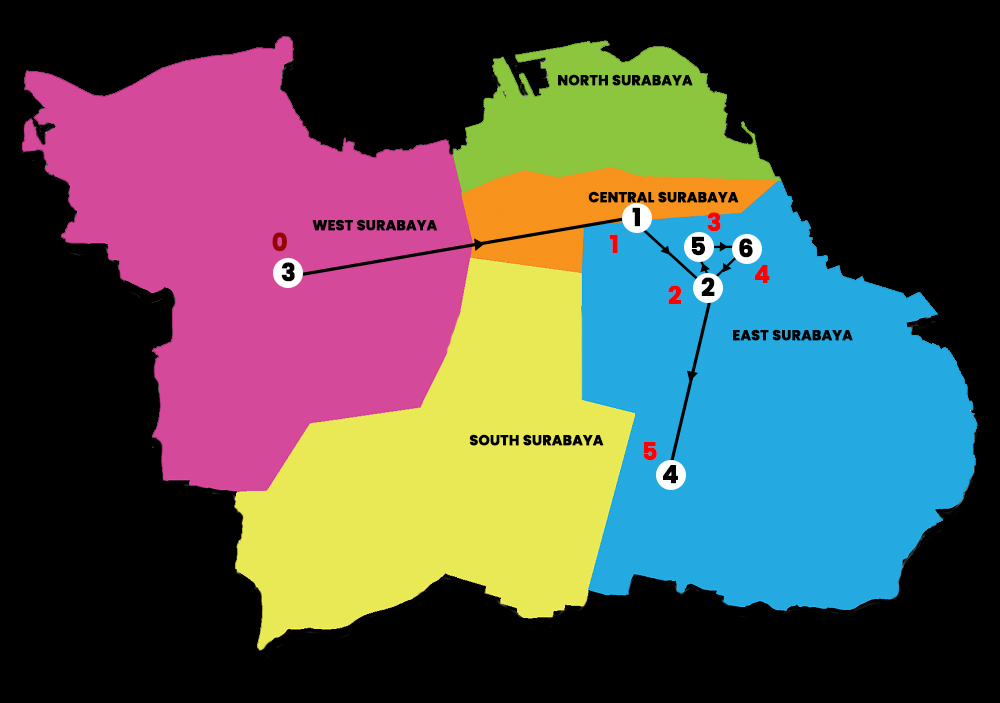
Dengan penggambaran grafik sebagai berikut:



**Gambar 2.6** Grafik pemetaan rute pemetaan universitas bebas di Surabaya berdasarkan penomoran di atas

UBAYAsebagai *vertex* awal, kemudian program menelusuri *vertex* yang berhubungan langsung dengan UBAYA, yaitu UINSA. Lalu program menelusuri *vertex* selanjutnya, yaitu UNESA. Setelah itu berlanjut ke UC, disusul dengan UNAIR. Maka program menelusuri ITS sebagai *vertex* terakhir.

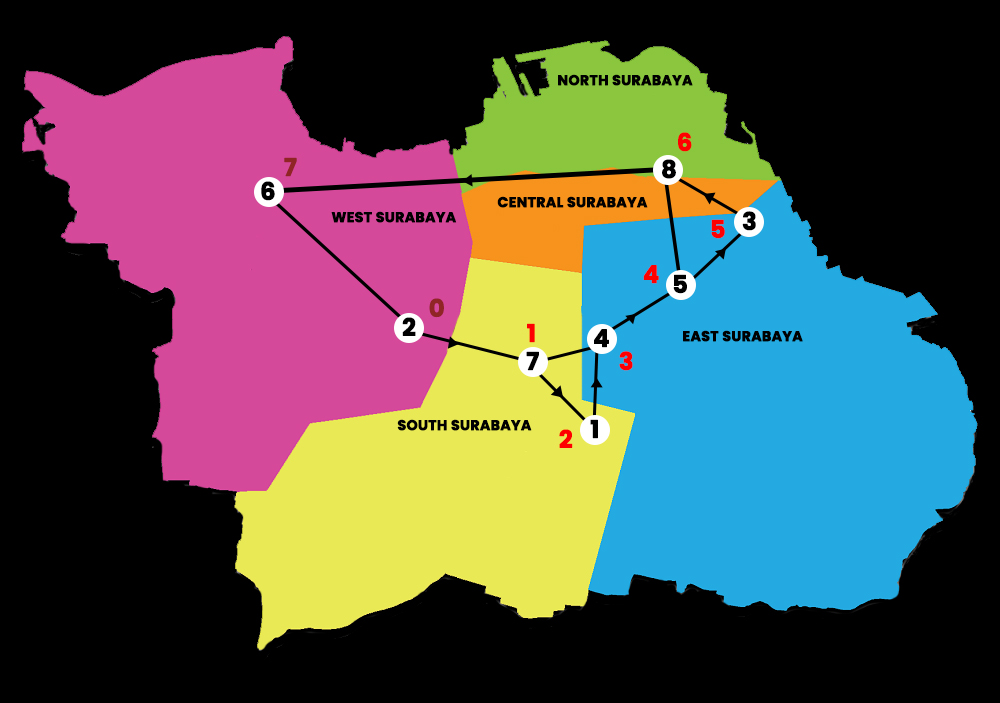
Setelah memilih salah satu menu dan menginputkan destinasi awal, maka program akan menetapkan destinasi tersebut menjadi *vertex* awal. Lalu dari *vertex* awal, menjelajah ke *vertex* lain yang terhubung langsung ke *vertex* awal. Sebagai contoh, apabila kita ingin destinasi awal kita adalah Universitas Negeri Surabaya (UNESA), pada menu 1, maka diharapkan proses *DFS* nya seperti berikut :



**Gambar 2.7** Grafik proses penelusuran *DFS* dari UNESA sebagai *vertex* awal

UNESAsebagai *vertex* awal, kemudian program menelusuri *vertex* yang berhubungan langsung dengan UNESA, yaitu UNAIR. Lalu program menelusuri *vertex* selanjutnya, yaitu ITS. Setelah itu berlanjut ke PENS, disusul dengan PPNS. Karena sudah tidak ada *vertex* lagi, maka program kembali menuju *vertex* sebelumnya, yakni ITS. Karena masih ada satu *vertex* yang belum dikunjungi, yakni UPN, maka program menelusuri UPN sebagai *vertex* terakhir.

Apabila kita ingin destinasi awal kita adalah Universitas Widya Kusuma Surabaya (UWKS), pada menu 2, maka diharapkan proses *DFS* nya seperti berikut :



**Gambar 2.8** Grafik proses penelusuran *DFS* dari UWKS sebagai *vertex* awal

UWKS sebagai *vertex* awal, kemudian program menelusuri *vertex* yang berhubungan langsung dengan UWKS, yaitu UINSA. Lalu program menelusuri *vertex* selanjutnya, yaitu UBAYA. Setelah itu berlanjut ke UNTAG, disusul dengan UHT. Program menelusuri *vertex* selanjutnya yakni UWM dan UMS. Maka program menelusuri UC sebagai *vertex* terakhir.

## 2.3 *Output* yang dihasilkan dari program tersebut

Pada program *Surabaya Destination Planner,* setelah menjalankan dan menginputkan jawaban sesuai dengan pilihan sebelumnya, program dapat berjalan dengan baik dan dapat memberikan output yang diharapkan. Berikut adalah hasil input berdasarkan program tersebut.

========= Surabaya Destination Planner =========

Where is your destination?

1. Arab Village (Kampung Arab), North Surabaya

2. Bulak Village (Kampung Bulak), North Surabaya

3. Sanggar Agung Temple (Klenteng Sanggar Agung), North Surabaya

4. 4 Face Buddha Statue (Patung Buddha 4 Rupa), North Surabaya

5. Museum House of Sampoerna, North Surabaya

6. Food Junction Grand Pakuwon, West Surabaya

7. Heroes Monument (Tugu Pahlawan), Central Surabaya

8. Heroes Museum (Museum Pahlawan), Central Surabaya

9. The Statue of Suro and Boyo (Patung Suro dan Boyo), South Surabaya

10. Surabaya Zoo (Kebun Binatang Surabaya), South Surabaya

> Surabaya Zoo

Your destination is Surabaya Zoo

Stay safe and happy travel, maybe we can suggest tourist destinations that are close to your destination:

Surabaya Zoo ---> The Statue of Suro and Boyo ---> Heroes Monument ---> Heroes Museum ---> Museum House of Sampoerna ---> Food Junction Grand Pakuwon ---> Arab Village ---> Bulak Village ---> 4 Face Buddha Statue ---> Sanggar Agung Temple ---> Finish

Output yang diberikan sesuai dengan penggambaran *BFS Surabaya Zoo* yang telah dijabarkan sebelumnya.

Apabila kita menginputkan *Museum House of Sampoerna* sebagai destinasi awal, maka program akan menunjukkan jawaban yang sesuai dengan penerapan *BFS*.

========= Surabaya Destination Planner =========

Where is your destination?

1. Arab Village (Kampung Arab), North Surabaya

2. Bulak Village (Kampung Bulak), North Surabaya

3. Sanggar Agung Temple (Klenteng Sanggar Agung), North Surabaya

4. 4 Face Buddha Statue (Patung Buddha 4 Rupa), North Surabaya

5. Museum House of Sampoerna, North Surabaya

6. Food Junction Grand Pakuwon, West Surabaya

7. Heroes Monument (Tugu Pahlawan), Central Surabaya

8. Heroes Museum (Museum Pahlawan), Central Surabaya

9. The Statue of Suro and Boyo (Patung Suro dan Boyo), South Surabaya

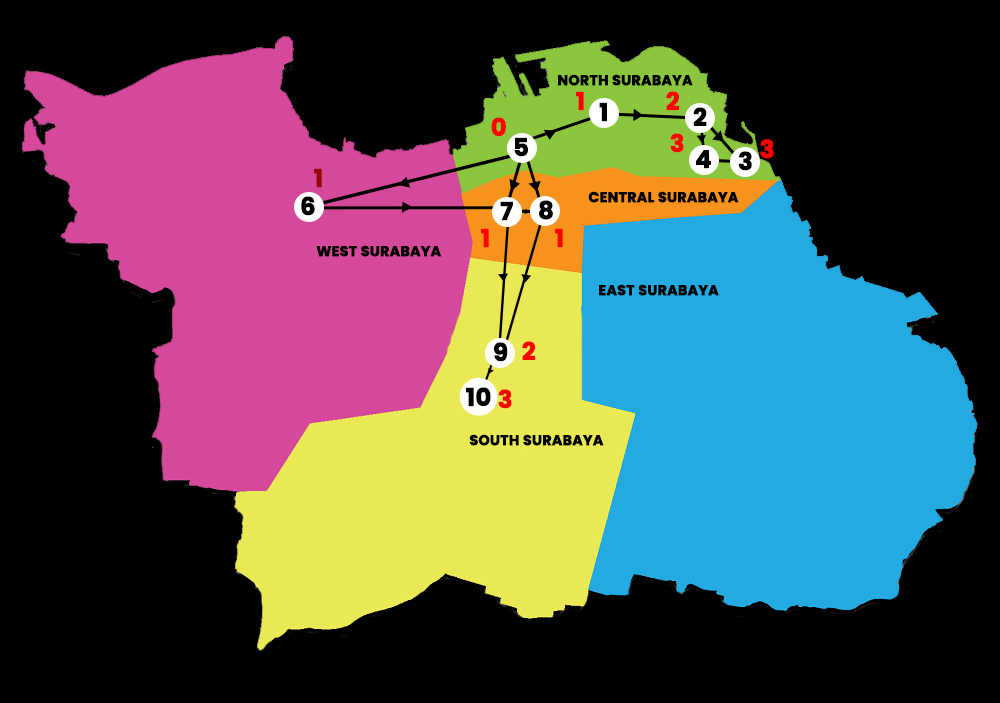
10. Surabaya Zoo (Kebun Binatang Surabaya), South Surabaya

> Museum House of Sampoerna

Your destination is Museum House of Sampoerna

Stay safe and happy travel, maybe we can suggest tourist destinations that are close to your destination:

Museum House of Sampoerna ---> Arab Village ---> Heroes Monument ---> Heroes Museum ---> Food Junction Grand Pakuwon ---> Bulak Village ---> The Statue of Suro and Boyo ---> 4 Face Buddha Statue ---> Sanggar Agung Temple ---> Surabaya Zoo ---> Finish



**Gambar 2.9** Grafik proses penelusuran *BFS* dari *Museum House of Sampoerna* sebagai *vertex* awal

Apabila kita menginputkan *Sanggar Agung Temple* sebagai destinasi awal, maka program akan menunjukkan jawaban yang sesuai dengan penerapan *BFS*.

========= Surabaya Destination Planner =========

Where is your destination?

1. Arab Village (Kampung Arab), North Surabaya

2. Bulak Village (Kampung Bulak), North Surabaya

3. Sanggar Agung Temple (Klenteng Sanggar Agung), North Surabaya

4. 4 Face Buddha Statue (Patung Buddha 4 Rupa), North Surabaya

5. Museum House of Sampoerna, North Surabaya

6. Food Junction Grand Pakuwon, West Surabaya

7. Heroes Monument (Tugu Pahlawan), Central Surabaya

8. Heroes Museum (Museum Pahlawan), Central Surabaya

9. The Statue of Suro and Boyo (Patung Suro dan Boyo), South Surabaya

10. Surabaya Zoo (Kebun Binatang Surabaya), South Surabaya

> Sanggar Agung Temple

Your destination is Sanggar Agung Temple

Stay safe and happy travel, maybe we can suggest tourist destinations that are close to your destination:

Sanggar Agung Temple ---> 4 Face Buddha Statue ---> Bulak Village ---> Arab Village ---> Museum House of Sampoerna ---> Heroes Monument ---> Heroes Museum ---> Food Junction Grand Pakuwon ---> The Statue of Suro and Boyo ---> Surabaya Zoo ---> Finish



**Gambar 2.10** Grafik proses penelusuran *BFS* dari *Sanggar Agung Temple* sebagai *vertex* awal

Apabila kita menginputkan *Heroes Monument* sebagai destinasi awal, maka program akan menunjukkan jawaban yang sesuai dengan penerapan *BFS*.

========= Surabaya Destination Planner =========

Where is your destination?

1. Arab Village (Kampung Arab), North Surabaya

2. Bulak Village (Kampung Bulak), North Surabaya

3. Sanggar Agung Temple (Klenteng Sanggar Agung), North Surabaya

4. 4 Face Buddha Statue (Patung Buddha 4 Rupa), North Surabaya

5. Museum House of Sampoerna, North Surabaya

6. Food Junction Grand Pakuwon, West Surabaya

7. Heroes Monument (Tugu Pahlawan), Central Surabaya

8. Heroes Museum (Museum Pahlawan), Central Surabaya

9. The Statue of Suro and Boyo (Patung Suro dan Boyo), South Surabaya

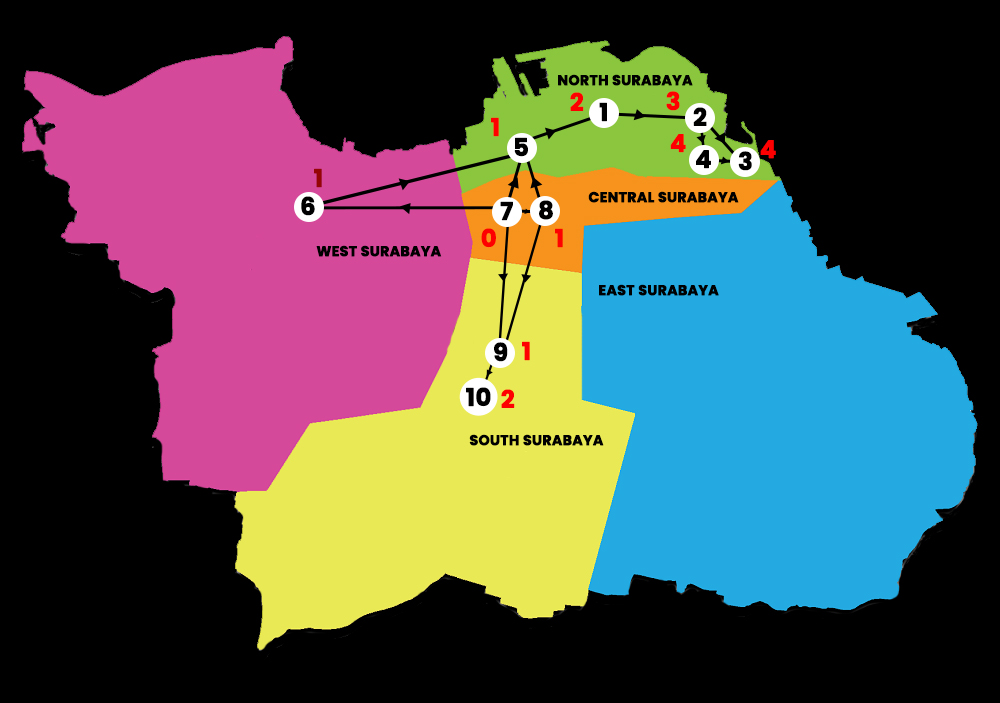
10. Surabaya Zoo (Kebun Binatang Surabaya), South Surabaya

> Heroes Monument

Your destination is Heroes Monument

Stay safe and happy travel, maybe we can suggest tourist destinations that are close to your destination:

Heroes Monument ---> Heroes Museum ---> Museum House of Sampoerna ---> Food Junction Grand Pakuwon ---> The Statue of Suro and Boyo ---> Arab Village ---> Surabaya Zoo ---> Bulak Village ---> 4 Face Buddha Statue ---> Sanggar Agung Temple ---> Finish



**Gambar 2.11** Grafik proses penelusuran *BFS* dari *Heroes Monument* sebagai *vertex* awal

Apabila kita menginputkan *Food Junction Grand Pakuwon* sebagai destinasi awal, maka program akan menunjukkan jawaban yang sesuai dengan penerapan *BFS*.

========= Surabaya Destination Planner =========

Where is your destination?

1. Arab Village (Kampung Arab), North Surabaya

2. Bulak Village (Kampung Bulak), North Surabaya

3. Sanggar Agung Temple (Klenteng Sanggar Agung), North Surabaya

4. 4 Face Buddha Statue (Patung Buddha 4 Rupa), North Surabaya

5. Museum House of Sampoerna, North Surabaya

6. Food Junction Grand Pakuwon, West Surabaya

7. Heroes Monument (Tugu Pahlawan), Central Surabaya

8. Heroes Museum (Museum Pahlawan), Central Surabaya

9. The Statue of Suro and Boyo (Patung Suro dan Boyo), South Surabaya

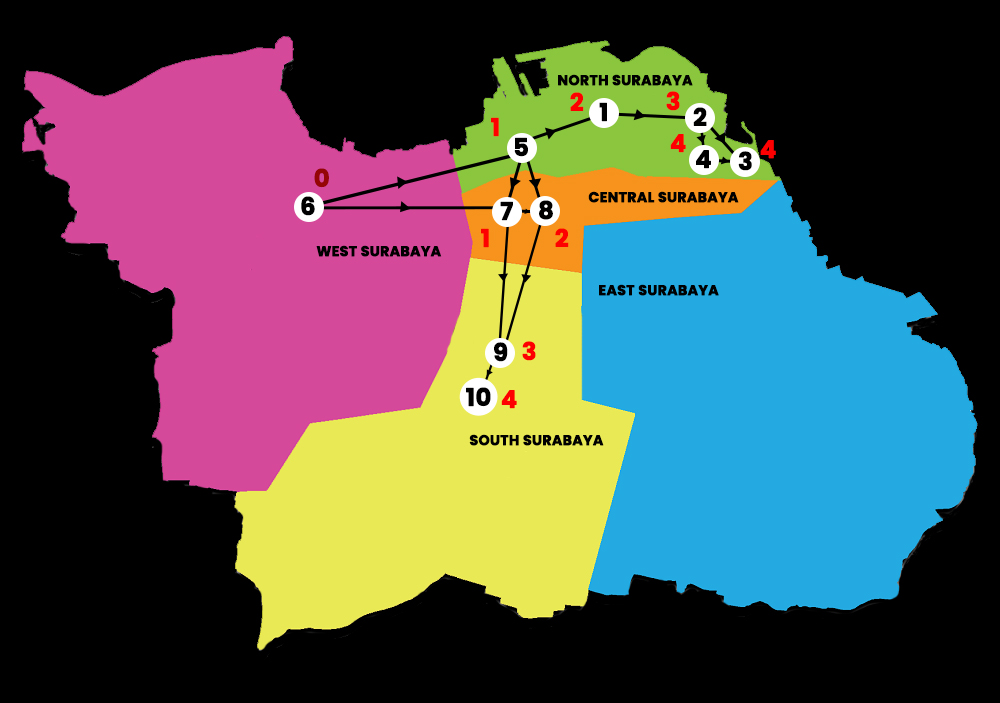
10. Surabaya Zoo (Kebun Binatang Surabaya), South Surabaya

> Food Junction Grand Pakuwon

Your destination is Food Junction Grand Pakuwon

Stay safe and happy travel, maybe we can suggest tourist destinations that are close to your destination:

Food Junction Grand Pakuwon ---> Museum House of Sampoerna ---> Heroes Monument ---> Arab Village ---> Heroes Museum ---> The Statue of Suro and Boyo ---> Bulak Village ---> Surabaya Zoo ---> 4 Face Buddha Statue ---> Sanggar Agung Temple ---> Finish



**Gambar 2.12** Grafik proses penelusuran *BFS* dari *Food Junction Grand Pakuwon* sebagai *vertex* awal

Sedangkan pada program *Surabaya Study Tour Planner*, setelah menjalankan dan menginputkan jawaban sesuai dengan pilihan sebelumnya, program dapat berjalan dengan baik dan dapat memberikan output yang diharapkan. Berikut adalah hasil input berdasarkan program tersebut.

========= Surabaya Study Tour Planner =========

Pilih jenis kampus yang ingin dituju?

1. Perguruan Tinggi Negeri

2. Perguruan Tinggi Swasta

3. Bebas

> 1

Tujuan pertama yang ingin dituju?

1. Universitas Airlangga (UNAIR)

2. Institut Teknologi Sepuluh November (ITS)

3. Universitas Negeri Surabaya (UNESA)

4. Universitas Pembangunan Negeri Veteran Jawa Timur (UPN)

5. Politeknik Elekronika Negeri Surabaya (PENS)

6. Politeknik Perkapalan Negeri Surabaya (PPNS)

(Masukkan singkatan kampus yang ingin dituju, cth : UNESA)

> UNESA

UNESA ---> UNAIR ---> ITS ---> PENS ---> PPNS ---> UPN ---> Finish

Output yang diberikan sesuai dengan penggambaran *DFS* UNESA yang telah dijabarkan sebelumnya.

Apabila kita menginputkan UNAIRsebagai destinasi awal, maka program akan menunjukkan jawaban yang sesuai dengan penerapan *DFS*.

========= Surabaya Study Tour Planner =========

Pilih jenis kampus yang ingin dituju?

1. Perguruan Tinggi Negeri

2. Perguruan Tinggi Swasta

3. Bebas

> 1

Tujuan pertama yang ingin dituju?

1. Universitas Airlangga (UNAIR)

2. Institut Teknologi Sepuluh November (ITS)

3. Universitas Negeri Surabaya (UNESA)

4. Universitas Pembangunan Negeri Veteran Jawa Timur (UPN)

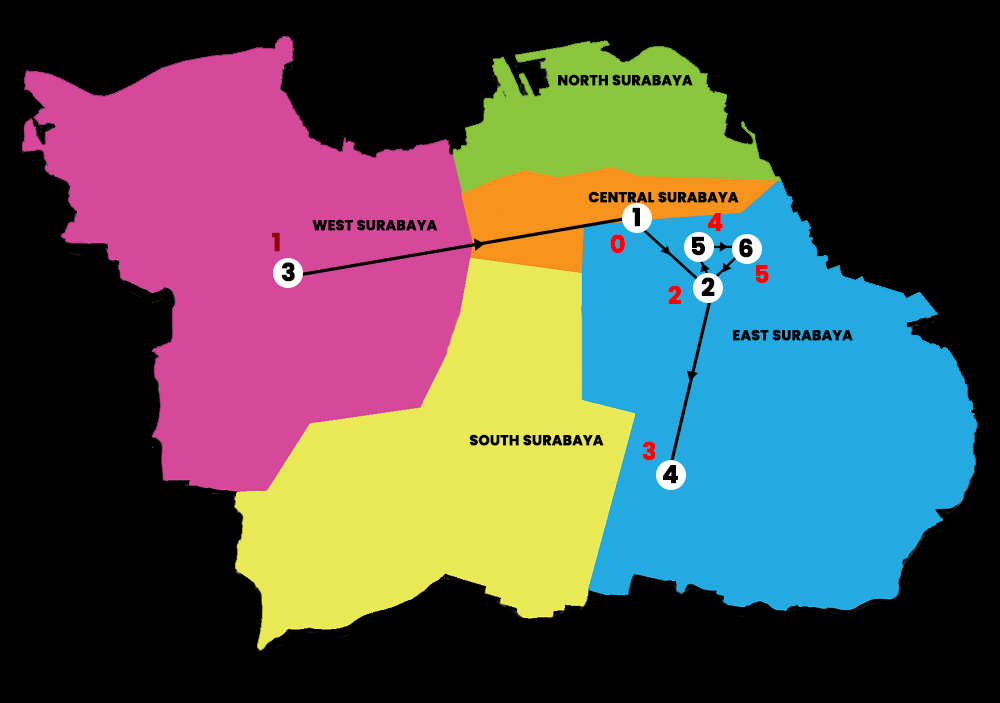
5. Politeknik Elekronika Negeri Surabaya (PENS)

6. Politeknik Perkapalan Negeri Surabaya (PPNS)

(Masukkan singkatan kampus yang ingin dituju, cth : UNESA)

> UNAIR

UNAIR ---> UNESA ---> ITS ---> UPN ---> PPNS ---> PENS ---> Finish



**Gambar 2.13** Grafik proses penelusuran *DFS* dari UNAIRsebagai *vertex* awal

Apabila kita menginputkan UWKSsebagai destinasi awal, maka program akan menunjukkan jawaban yang sesuai dengan penerapan *DFS,* sesuai pada penggambaraan *DFS* UKWS yang telah dijabarkan sebelumnya.

========= Surabaya Study Tour Planner =========

Pilih jenis kampus yang ingin dituju?

1. Perguruan Tinggi Negeri

2. Perguruan Tinggi Swasta

3. Bebas

> 2

Tujuan pertama yang ingin dituju?

1. Universitas Surabaya (UBAYA)

2. Universitas Wijaya Kusuma Surabaya (UWKS)

3. Universitas Katolik Widya Mandala (UWM)

4. Universitas 17 Agustus 1945 (UNTAG)

5. Universitas Hang Tuah (UHT)

6. Universitas Ciputra (UC)

7. Universitas Islam Negeri Sunan Ampel (UINSA)

8. Universitas Muhammadiyah Surabaya (UMS)

(Masukkan singkatan kampus yang ingin dituju, cth : UBAYA)

> UWKS

UWKS ---> UINSA ---> UBAYA ---> UNTAG ---> UHT ---> UWM ---> UMS ---> UC ---> Finish

Output yang diberikan sesuai dengan penggambaran *DFS* UWKS yang telah dijabarkan sebelumnya.

Apabila kita menginputkan UNTAGsebagai destinasi awal, maka program akan menunjukkan jawaban yang sesuai dengan penerapan *DFS*.



**Gambar 2.14** Grafik proses penelusuran *DFS* dari UNAIRsebagai *vertex* awal

Apabila kita memilih menu 3. Bebas, maka program akan menampilkan output sesuai dengan jabaran penerapan *DFS* sebelumnya

========= Surabaya Study Tour Planner =========

Pilih jenis kampus yang ingin dituju?

1. Perguruan Tinggi Negeri

2. Perguruan Tinggi Swasta

3. Bebas

> 3

Berikut adalah tujuan yang kami rekomendasikan:

UBAYA ---> UINSA ---> UNESA ---> UC ---> UNAIR ---> ITS ---> Finish



**Gambar 2.15** Grafik pemetaan rute pemetaan universitas bebas di Surabaya berdasarkan penomoran di atas