

## 1.1 Judul Proyek

Judul proyek kami yaitu “FIVEL (*Find My Travel*)”

## 1.2 Latar Belakang

Pariwisata adalah perjalanan yang dilakukan oleh seseorang dalam jangka waktu tertentu dari suatu tempat ke tempat lain dengan melakukan perencanaan sebelumnya, tujuannya untuk rekreasi atau untuk suatu kepentingan sehingga keinginannya dapat terpenuhi [1]. Pariwisata merupakan salah satu sektor penting dalam kehidupan manusia, dikarenakan memberikan kesempatan untuk berekreasi, eksplorasi, dan membuat pengalaman baru. Seiring dengan meningkatnya minat masyarakat untuk bepergian, maka kebutuhan akan cara yang lebih mudah dalam merencanakan perjalanan pun terus berkembang. Dalam beberapa tahun terakhir, teknologi telah memberikan dampak yang besar dalam bidang pariwisata, di mana teknologi membantu para wisatawan dalam memilih destinasi, mencari informasi, hingga mengatur jadwal perjalanannya.

Kemajuan teknologi ini didukung dengan penerapan berbagai metode untuk meningkatkan pengalaman wisata. Salah satunya adalah metode *graph*. Dalam konteks perjalanan, *graph* digunakan untuk merepresentasikan hubungan antar lokasi wisata serta menentukan jalur terbaik antara satu destinasi dengan destinasi wisata lainnya. Metode lainnya yang dapat digunakan untuk menyelesaikan permasalahan pencarian jalur terpendek adalah algoritma Dijkstra. Algoritma Dijkstra merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi yang lain. Prinsip dari algoritma Dijkstra adalah dengan pencarian dua lintasan yang paling kecil [2]. Dengan begitu para wisatawan dapat dengan mudah menemukan jalur terpendek menuju destinasi yang diinginkan, serta mengoptimalkan waktu perjalanan.

Berkaitan dengan hal tersebut, maka penulis menggagas sebuah inovasi baru yang dapat menjadi solusi untuk mempermudah perencanaan perjalanan wisata dengan judul program yaitu “FIVEL (*Find My Travel*)”. FIVEL dirancang agar dapat membantu orang-orang yang ingin berpergian dalam membuat, mengedit, dan menghapus jadwal perjalanannya. FIVEL juga dapat dilakukan pencarian dan pengurutan destinasi wisata. Dengan begitu, FIVEL diharapkan mampu memberikan kenyamanan yang lebih baik bagi penggunaanya dalam merencanakan perjalanan wisata. Selain itu, pada FIVEL juga tertera informasi terkait jarak, harga, dan *rating* dari setiap destinasi wisata guna mendukung keputusan para wisatawan dalam memilih tempat yang akan dikunjungi.

### 1.3 Deskripsi Program

FIVEL (*Find My Travel*) merupakan program yang dirancang untuk memudahkan para wisatawan dalam merencanakan perjalanan wisata mereka. Di mana program ini menggabungkan berbagai elemen interaktif, sehingga pengguna dapat mencari destinasi wisata, mengelola jadwal perjalanan, serta memperoleh informasi terkait jarak, harga, dan *rating* dari objek wisata yang ada. FIVEL dapat memudahkan pengguna untuk mencari destinasi wisata dengan didukung oleh metode *sorting* berdasarkan *rating*, jarak, atau harga. *User* atau pengguna dapat memilih destinasi berdasarkan preferensi mereka, kemudian program akan menampilkan hasil yang relevan. Dengan metode *sorting* ini, pengguna dapat menemukan tempat wisata yang paling sesuai dengan kriteria yang diinginkan dan mempercepat proses pencarian. Program ini juga menggunakan *graph* untuk menampilkan hubungan antara berbagai destinasi wisata dan jarak antar tempat. Pengguna dapat melihat jalur terdekat untuk mencapai suatu tempat wisata yang membantu mereka merencanakan perjalanan dengan lebih efisien. Setiap objek wisata juga dilengkapi dengan informasi berupa *rating* dan harga, sehingga pengguna akan mendapat gambaran terhadap tempat destinasi yang akan dikunjungi.

Dengan program yang interaktif, maka akan dapat membuat pengguna untuk mengelola jadwal perjalanan mereka dengan mudah. Pengguna dapat menambah, mengedit, dan menghapus destinasi wisata dari jadwal yang telah dibuat. Setelah mengunjungi tempat wisata, pengguna dapat menandai jadwal sebagai selesai, dan jadwal yang selesai akan dipindahkan ke riwayat perjalanan yang disimpan menggunakan struktur data *stack*. Dengan begitu, pengguna dapat melihat kembali perjalanan mereka. Selain itu, program juga dilengkapi dengan *login* dan *logout*, serta membedakan akses antara pengguna biasa dan admin. Pengguna hanya dapat mengelola jadwal mereka, sementara admin memiliki akses penuh untuk melakukan CRUD (*create, read, update, delete*) pada destinasi wisata, riwayat perjalanan, serta data pengguna lainnya. Admin dapat menambah atau menghapus destinasi wisata yang tersedia dan mengelola data yang ada pada program. Dengan begitu, diharapkan program FIVEL ini akan membantu dalam merencanakan perjalanan wisata pengguna dengan lebih baik.

### 1.4 Algoritma

Berikut ini merupakan algoritma yang digunakan untuk menyelesaikan permasalahan pada program:

## 1.5 Penjelasan Kode

### 1.5.1 User Class

```
public class User{
    String username;
    String password;
    String role;
    User next;
    public User(String username, String password, String role){
        this.username = username;
        this.password = password;
        this.role = role;
        this.next = null;
    }
}
```

*Script* di atas merupakan *class* “User”, di mana pada kelas “User” ini terdapat variabel seperti “username”, “password”, dan “role” untuk menyimpan informasi terkait pengguna. *Class* “User” dirancang untuk menyimpan informasi pengguna dan mengelola data dengan cara saling terhubung menggunakan atribut “next”. Konstruktor pada *class* ini digunakan untuk menginisialisasi atribut “username”, “password”, dan “role” sesuai dengan nilai yang diberikan saat objek dibuat. Selain itu, “next” diatur ke *null* yang berarti objek tersebut belum terhubung dengan objek “User” lainnya. “next” diatur ke objek “User” lain untuk membuat daftar pengguna yang saling terhubung pada struktur data *single linked list*. Sehingga dapat digunakan untuk menyimpan dan mengelola data pengguna.

### 1.5.2 LinkedList Class

```
public class LinkedList {
    User head;
    void add(User user){
        User newUser = new User(user.username, user.password,
user.role);
        newUser.next = head;
        head = newUser;
    }
    User find(String username, String password){
        User current = head;
        while(current != null){
            if(current.username.equals(username)    &&
current.password.equals(password)){
                return current;
            }
            current = current.next;
        }
        return null;}
    void display(){
        User current = head;
        while(current != null){
            System.out.println("- " + current.username + " (" +
current.role + ")");
            current = current.next;        }}}}
```

*Script* di atas adalah *class* “LinkedList” yang mengimplementasikan struktur data *single linked list* untuk menyimpan objek “User”. Pada kelas “LinkedList” terdapat variabel “head” yang digunakan sebagai referensi ke objek “User” pertama dalam daftar. Kemudian ada *method* “add” yang digunakan untuk menambahkan pengguna baru ke awal daftar dengan membuat salinan objek “User” dan mengatur atribut “next” dari pengguna baru tersebut ke “head”, lalu memperbarui “head” ke pengguna baru. *Method* “find” digunakan untuk mencari pengguna berdasarkan “username” dan “password” dengan cara memeriksa setiap *node* dalam daftar secara berurutan hingga menemukan kecocokan atau mencapai akhir daftar, dan mengembalikan pengguna yang sesuai. *Method* “display” digunakan untuk menampilkan semua pengguna dalam daftar mulai dari “head”, mencetak “username” dan “role” setiap pengguna hingga akhir daftar

### 1.5.3 Schedule Class

```
class Schedule {
    String judulPerjalanan;
    String lokasiAwal;
    String lokasiTujuan;
    LinkedList jalur;
    int totalJarak;
    String waktuMulai;
    String waktuSelesai;
    public Schedule(String judulPerjalanan, String lokasiAwal,
String lokasiTujuan, LinkedList jalur, int totalJarak, String
waktuMulai) {
        this.judulPerjalanan = judulPerjalanan;
        this.lokasiAwal = lokasiAwal;
        this.lokasiTujuan = lokasiTujuan;
        this.jalur = jalur;
        this.totalJarak = totalJarak;
        this.waktuMulai = waktuMulai;
        this.waktuSelesai = null; }
    void tandaiSelesai(String waktuSelesai) {
        this.waktuSelesai = waktuSelesai;}}
```

*Script* di atas mendefinisikan kelas “Schedule” yang digunakan untuk menggambarkan jadwal perjalanan. Yang dimana dalam kelas tersebut terdapat Variabel seperti “judulPerjalanan”, “lokasiAwal”, “lokasiTujuan” yang bertipe data String dan variable “jalur” menggunakan struktur data LinkedList, yang menyimpan urutan rute atau jalur yang akan ditempuh selama perjalanan. Selain itu terdapat juga variabel “totalJarak” yang bertipe integer dan “waktuMulai”, “waktuSelesai” dalam bentuk string. Konstraktor “Schedule” digunakan untuk menginisialisasi objek baru dengan parameter meliputi

“judulPerjalanan”, “lokasiAwal”, “lokasitujuan”, “jalur”, “totalJarak”, dan waktuMulai” yang dimana variabel “waktuSelesai” diinisialisai dengan “null”. Metode “tandaiSelesai” untuk memperbarui nilai “waktuSelesai” setelah perjalanan selesai.

## 1.6 Output Program

Berikut merupakan *output* dari program pengimplementasian *graph* pada program FIVEL (*Find My Travel*) yang telah kami buat:

## 1.7 Kesimpulan

Berdasarkan hasil proyek praktikum Algoritma dan Struktur Data, dapat disimpulkan bahwa

## 1.8 Referensi

- [1] C. M. Lengkong, R. Sengkey, and B. A. Sugiarto, "Sistem informasi pariwisata berbasis web di Kabupaten Minahasa," *Jurnal Teknik Informatika*, vol. 14, no. 1, pp. 15-20, 2019.
- [2] V. H. Pranatawijaya, W. Widiatry, N. N. K. Sari, and P. B. A. A. Putra, "Sistem Informasi Geografis Mencari Rute Lokasi Travel Di Kota Palangka Raya Berbasis Website," *Jurnal Teknologi Informasi: Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika*, vol. 13, no. 1, pp. 76-82, 2019.