|  |  |
| --- | --- |
| Nama | : La Ode Muhammad Gazali |
| NIM | : 222212696 |
| Kelas | : 2KS2 |

**TUGAS PRA-PERTEMUAN 7 PEMROGRAMAN BERORIENTASI OBJEK**

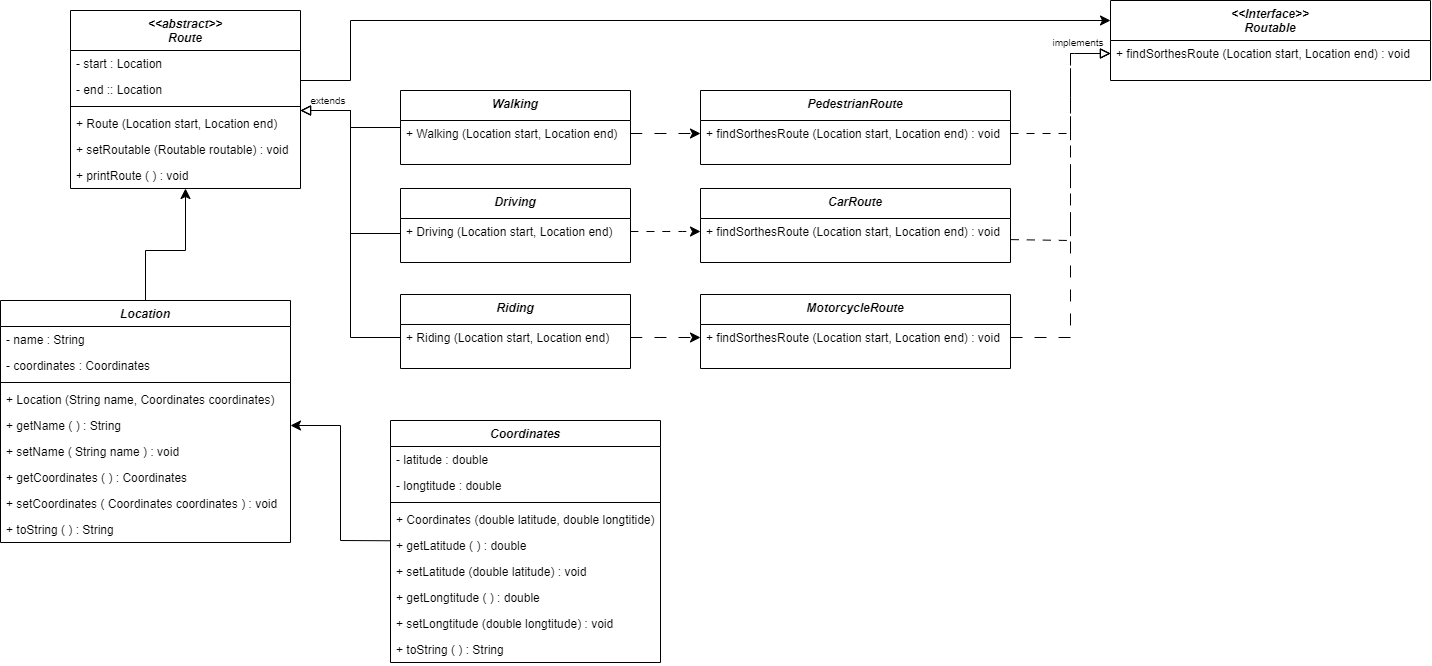
**(Design Pattern)**

**Case Study**

|  |  |
| --- | --- |
|  | * Sebuah aplikasi memiliki fitur untuk menentukan rute dua titik lokasi. Fitur ini memungkinkan pengguna untuk memilih rute terbaik sesuai moda transportasi yang dipilih. * Buatlah desain kelas dari fitur aplikasi tersebut menggunakan class diagram dengan menerapkan strategry pattern. Selanjutnya implementasikan dengan bahasa pemrograman Java. |

**Penyelesaian :**

1. **Class Diagram**

****

1. **Implementasi Program**

* **Coordinates.java**

public class Coordinates {

    private double latitude;

    private double longitude;

    public Coordinates(double latitude, double longitude) {

      this.latitude = latitude;

      this.longitude = longitude;

    }

    public double getLatitude() {

      return latitude;

    }

    public void setLatitude(double latitude) {

      this.latitude = latitude;

    }

    public double getLongitude() {

      return longitude;

    }

    public void setLongitude(double longitude) {

      this.longitude = longitude;

    }

    @Override

    public String toString() {

      return (

        "Coordinates[latitude=" + latitude + ", longitude=" + longitude + "]"

      );

    }

}

* **Location.java**

public class Location {

    private String name;

    private Coordinates coordinates;

    public Location(String name, Coordinates coordinates) {

      this.name = name;

      this.coordinates = coordinates;

    }

    public String getName() {

      return name;

    }

    public void setName(String name) {

      this.name = name;

    }

    public Coordinates getCoordinates() {

      return coordinates;

    }

    public void setCoordinates(Coordinates coordinates) {

      this.coordinates = coordinates;

    }

    @Override

    public String toString() {

      return "Location[name=" + name + ", coordinates=" + coordinates + "]";

    }

}

* **Routable.java (Interface)**

public interface Routable{

    public void findShortestRoute(Location start, Location end);

}

/\*\*

\* CarRoute

 \*/

class CarRoute implements Routable{

    @Override

    public void findShortestRoute(Location start, Location end){

        System.out.println("Ini rute untuk mobil");

    }

}

/\*\*

\* MotorcycleRoute

 \*/

class MotorcycleRoute implements Routable{

    @Override

    public void findShortestRoute(Location start, Location end){

        System.out.println("Ini rute untuk motor");

    }

}

/\*\*

\* PedestrianRoute

 \*/

class PedestrianRoute implements Routable{

    @Override

    public void findShortestRoute(Location start, Location end){

        System.out.println("Ini rute untuk pejalan kaki/pedestrian");

    }

}

* **Route.java (Abstract Class)**

public abstract class Route {

    private Location start;

    private Location end;

    private Routable routable;

    protected Route(Location start, Location end){

        this.start = start;

        this.end = end;

    }

    public void setRoutable(Routable routable){

        this.routable = routable;

    }

    public void printRoute() {

        routable.findShortestRoute(start, end);

    }

}

/\*\*

\* Driving

 \*/

class Driving extends Route {

    public Driving (Location start, Location end){

        super(start, end);

        super.setRoutable(new CarRoute());

    }

}

/\*\*

  \* Riding

  \*/

class Riding extends Route {

    public Riding(Location start, Location end){

        super(start, end);

        super.setRoutable(new MotorcycleRoute());

    }

}

/\*\*

 \* Walking

 \*/

class Walking extends Route{

    public Walking(Location start, Location end){

        super(start, end);

        super.setRoutable(new PedestrianRoute());

    }

}

* **Main\_Test.java**

public class Main\_Test {

    public static void main (String args[]){

        Location STIS = new Location(

            "POLSTAT STIS",

            new Coordinates(-6.231d, 106.867)

        );

        Location STAN = new Location(

            "PKN STAN",

            new Coordinates(-6.267d, 106.732)

        );

        Route drivingRoute = new Driving(STIS, STAN);

        drivingRoute.printRoute();

        Route ridingRoute = new Riding(STIS, STAN);

        ridingRoute.printRoute();

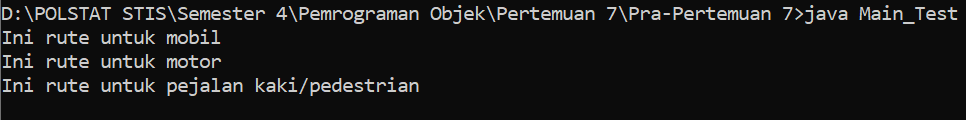
        Route walkingRoute = new Walking(STIS, STAN);

        walkingRoute.printRoute();

  }

}

* **Hasil Compile**

****