

Agregasi

```
public class Admin extends Account{
    private User[] users; //agregasi
    private Admin[] admins; //agregasi
    private Place[] places; //agregasi
```

Collection

```
ObservableList<UserView> list = FXCollections.observableArrayList();
for (Admin adminItm : admin.getAdmins()) {
    list.add(new UserView(adminItm.getEmail(), adminItm.getName(),0));
}
```

Database

```
public class Database {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/tubespbo";
    static final String USER = "root";
    static final String PASS = "";

    public static String email = "";

    public static Connection conn;
    public static Statement stmt;
    public static ResultSet rs;

    public Database() throws SQLException{
        getConnection();
        stmt = conn.createStatement(); //buat objek statment
    }

    public void getConnection() {
        try {
            Class.forName(JDBC_DRIVER);
            conn = DriverManager.getConnection(DB_URL, USER, PASS); //buat connection
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }

    public ResultSet query(String sql) throws SQLException{
        ResultSet s = stmt.executeQuery(sql); //execute query
        return s;
    }

    public void update(String sql) throws SQLException{
        stmt.executeUpdate(sql); //execute query
    }
}
```

Enkapsulasi

```
public class PlaceView {
    private SimpleStringProperty name;
    private SimpleDoubleProperty x, y;
    public PlaceView(Place place){
        this.name = new SimpleStringProperty(place.getName());
        this.x = new SimpleDoubleProperty(place.getCoordinate().getX());
        this.y = new SimpleDoubleProperty(place.getCoordinate().getY());
    }

    public String getName() {
        return name.get();
    }

    public Double getX() {
        return x.get();
    }

    public Double getY() {
        return y.get();
    }
}
```

Generic

```
@FXML
private TableView<UserView> table;

@FXML
private TableColumn<UserView, String> email, name;
```

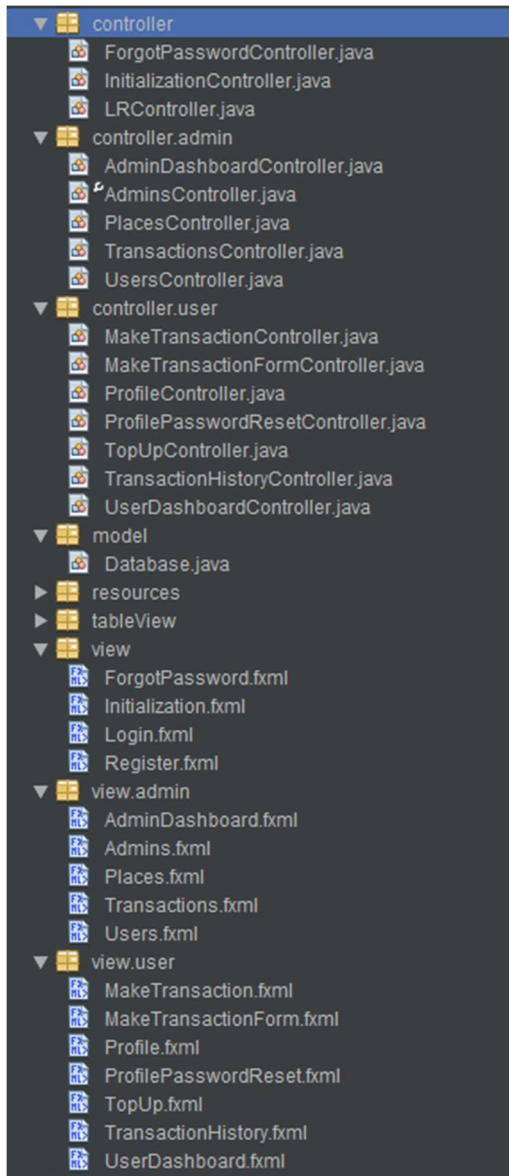
Inheritance

```
public class Admin extends Account{
    private User[] users; //agregasi
    private Admin[] admins; //agregasi
    private Place[] places; //agregasi
```

Komposisi & Abstract

```
public abstract class Account implements Nameable, Transactionable {
    protected boolean isAdmin;
    protected SimpleStringProperty password;
    protected SimpleStringProperty email;
    protected SimpleStringProperty name;
    protected Transaction[] transactions; // komposisi
```

MVC



Polimorfisme Overloading

```
public Place(){
    this.name = new SimpleStringProperty("");
    coordinate = new Point(0,0);
}

public Place(String name,double x,double y){
    this.name = new SimpleStringProperty(name);
    coordinate = new Point(x,y);
}
```