

MVC

Anggota:

- **2440068902 Leonardo Wijaya**
- **2440047980 Benyamin Kristiawan Aliwinoto**
- **2440021161 Matthew Tirtasaputra**
- **2440062810 Bunga Sari Ariansyah**

Definisi dari MVC:

MVC atau yang sering dikenal sebagai Model View Controller adalah sebuah pola design dalam system yang terdiri dari:

Model : bagian yang terhubung langsung dengan database.

View : bagian yang akan jadi tampilan informasi dari sebuah model ke pengguna.

Controller : bagian yang menghubungkan model dan view dalam setiap proses yang dirquest oleh pengguna.

Dari project yang kami buat, kami akan menampilkan dan menjelaskan bagian mana saja yang merupakan MVC pada project kami.

1.

Kita akan mulai dengan View dalam kode kami, kami bisa melihat bagian view dari MotorcycleView.java yang memberikan kemampuan kepada user untuk melihat informasi/data yang kami berikan.

```
1 public class MotorcycleView {
2
3     public void printMotorcycleDetails(int motorcycleID, String motorcycleName, String motorcycleBrand,
4         String motorcycleColor, int motorcycleCC, int motorcyclePrice) {
5         System.out.println("| " + motorcycleID + " | " + motorcycleName + " | " + motorcycleBrand + " | "
6             + motorcycleColor + " | " + motorcycleCC + " | Rp. " + motorcyclePrice + " |");
7     }
8 }
```

2.

Yang kedua ada Controller yaitu yang mengendalikan alur program secara keseluruhan seperti snippet code dibawah yang berasal dari MotorcycleController.java

```
import java.sql.ResultSet;
import java.sql.SQLException;

public class MotorcycleController {

    Connect con = Connect.getConnection();

    public void addMotorcycle(String motorcycleName, String
motorcycleBrand, String motorcycleColor, int motorcycleCC,
        int motorcyclePrice) {
        String query = String.format(
            "INSERT INTO `motorcycles`(`name`, `brand`, `color`,
`cc`, `price`) VALUES ('%s', '%s', '%s', '%d' , '%d')",
            motorcycleName, motorcycleBrand, motorcycleColor,
motorcycleCC, motorcyclePrice);
        con.executeUpdate(query);
    }

    public void showMotorcycle() {
        String query = String.format("SELECT * FROM motorcycles");
        ResultSet rs = con.executeQuery(query);

        System.out.println(
            "| Motorcycle ID | Motorcycle Name | Motorcycle Brand
| Motorcycle Color | Motorcycle CC | Motorcycle OTR Price |");
        try {
            while (rs.next()) {
```

```

        Motorcycle model = new Motorcycle(rs.getInt("id"),
rs.getString("name"), rs.getString("brand"),
        rs.getString("color"), rs.getInt("cc"),
rs.getInt("price"));
        MotorcycleView view = new MotorcycleView();
        view.printMotorcycleDetails(model.getId(),
model.getName(), model.getBrand(), model.getColor(),
        model.getCc(), model.getPrice());
    }

    } catch (SQLException e) {
        // TODO: handle exception
        e.printStackTrace();
    }
};

public void updateMotorcycle(int motorcycleID, String
motorcycleName, String motorcycleBrand,
        String motorcycleColor, int motorcycleCC, int
motorcyclePrice) {
    String query = String.format(
        "UPDATE `motorcycles` SET `name` = '%s', `brand` =
'%s', `color` = '%s', `cc` = '%d', `price` = '%d' WHERE
`motorcycles`.`id` = %d",
        motorcycleName, motorcycleBrand, motorcycleColor,
motorcycleCC, motorcyclePrice, motorcycleID);
    con.executeUpdate(query);
}
}

```

3. Yang terakhir adalah Model yang menggambarkan struktur data pada database seperti pada Motorcycle.java

```

4. public class Motorcycle {
5.
6.     private int id;
7.     private String name;
8.     private String brand;
9.     private String color;
10.    private int cc;
11.    private int price;
12.
13.    public Motorcycle(int id, String name, String brand,
String color, int cc, int price) {
14.        this.id = id;
15.        this.name = name;
16.        this.brand = brand;
17.        this.color = color;
18.        this.cc = cc;
19.        this.price = price;
20.    }
21.

```

```
22.     public int getId() {
23.         return id;
24.     }
25.
26.     public void setId(int id) {
27.         this.id = id;
28.     }
29.
30.     public String getName() {
31.         return this.name;
32.     }
33.
34.     public void setName(String name) {
35.         this.name = name;
36.     }
37.
38.     public String getBrand() {
39.         return this.brand;
40.     }
41.
42.     public void setBrand(String brand) {
43.         this.brand = brand;
44.     }
45.
46.     public String getColor() {
47.         return this.color;
48.     }
49.
50.     public void setColor(String color) {
51.         this.color = color;
52.     }
53.
54.     public int getCc() {
55.         return this.cc;
56.     }
57.
58.     public void setCc(int cc) {
59.         this.cc = cc;
60.     }
61.
62.     public int getPrice() {
63.         return this.price;
64.     }
65.
66.     public void setPrice(int price) {
67.         this.price = price;
68.     }
```