



TUGAS PERTEMUAN KE – 8
(PRAKTIKUM OOP 2022-202)

NAMA	Tesalonika Dua Nurak
NIM	2218015
KELAS	A
PEMBERI TUGAS	Muhammad Ridho Putra Syalabi (2118014)

8.1 Tugas Rumah : Menerapkan Konsep Interface

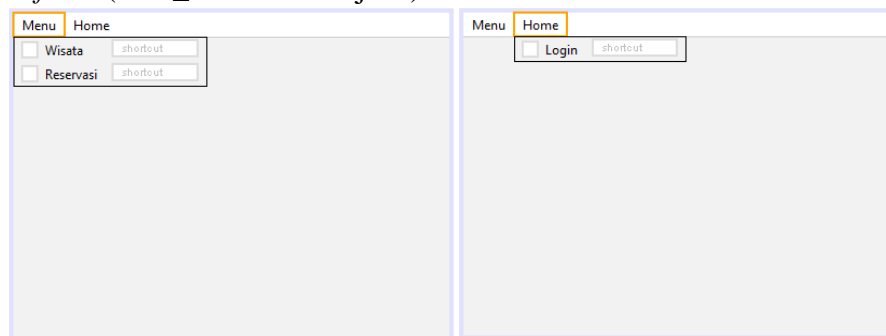
Judul : Informasi Wisata

Tema : Sistem Informasi

List
<ul style="list-style-type: none">• Nama• Kota• Deskripsi• Harga• Jumlah• Total
<ul style="list-style-type: none">+ void dataNama+ void dataKota+ void dataDeskripsi+ String cetakNama+ String cetakKota+ String cetakDeskripsi

Interface Info
Void getInfo

Desain *form* (GUI_MenuUtama.java):



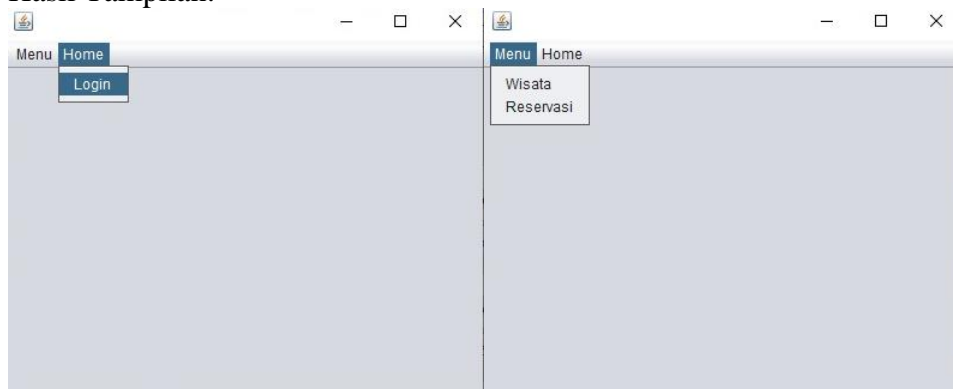
Gambar 8.1 Desain GUI_MenuUtama .java



Tabel 8.1 Properti GUI_MenuUtama.java

No	Nama Komponen	Properti	Value
1	jLabel1	Text	PROGRAM WISATA
2	jMenuBar	jMenu	1] = Menu [2] = Home
		jMenuItem	1] = Wisata [2] = Reservasi [3] = Login

Hasil Tampilan:



Gambar 8.2 Tampilan GUI_MenuUtama .java

Source Code GUI_MenuUtama :

```
private void
jMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Wisata mhs = new Wisata();mhs.show();
}

private void
jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    Daftar mhs = new Daftar();mhs.show();
}

private void
jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    Reservasi mk = new Reservasi();mk.show();
}
```

Source code Object Class (List):

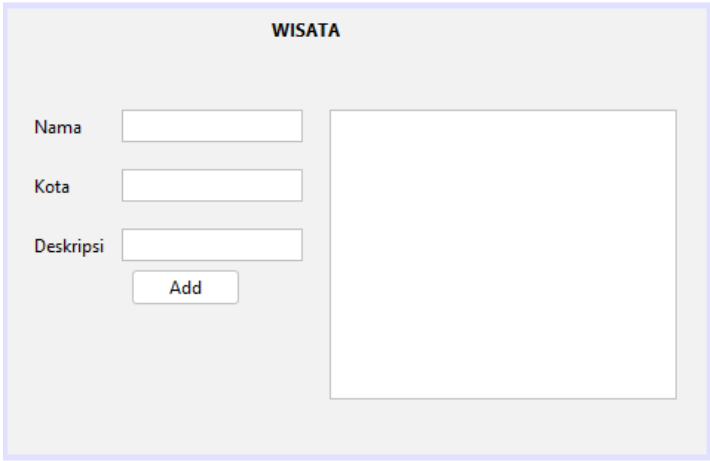
```
public class List implements Info {
    String Nama, Deskripsi, Kota;

    public void dataKota(String Kota){
        this.Kota = Kota;
    }
}
```



```
    }
    void dataNama(String Nama){
        this.Nama = Nama;
    }
    void dataDeskripsi(String Deskripsi){
        this.Deskripsi = Deskripsi;
    }
    public void getInfo() {
        System.out.println("List Wisata");
        System.out.println("-----");
        System.out.println(" Nama : " + Nama);
        System.out.println(" Kota: " + Kota);
        System.out.println(" Deskripsi: " + Deskripsi);
    }
}
```

Desain form (Wisata.java):



Gambar 8.3 Desain GUI_Wisata .java

Tabel 8.2 Properti GUI_Wisata.java

No	Nama Komponen	Properti	Value
1	jLabel1	Text	Wisata
2	jLabel2	Text	Nama
3	jLabel3	Text	Kota
4	jLabel4	Text	Deskripsi
5	jTextField1	Name	txtNama
		Text	“ “
6	jTextField2	Name	txtKota
		Text	“ “
7	jTextField3	Name	txtDes
		Text	“ “

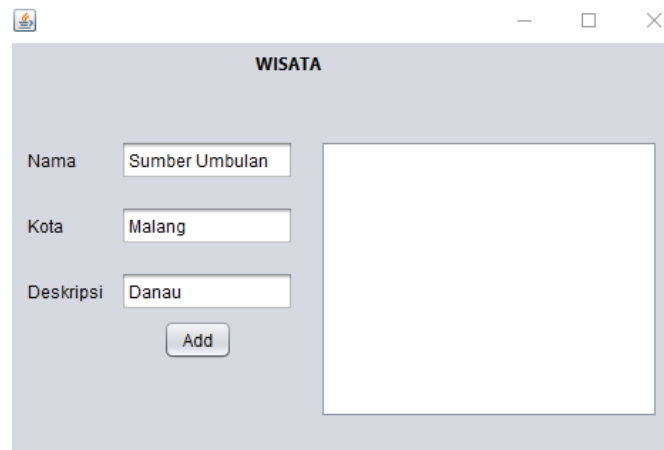


8	jButton1	Name Text	Add btnData
9	jScrollPane1	Name	memoDta
		Text	“ “

Source Code button Add :

```
private void  
btnDataActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    memoData.setText("");  
  
    // Use polymorphism by declaring the variable as  
    the interface type  
    Info displayable = new List();  
  
    // Populate the data  
    ((List) displayable).dataNama(txtNama.getText());  
    ((List) displayable).dataKota(txtKota.getText());  
    ((List)  
displayable).dataDeskripsi(txtDes.getText());  
  
    // Use polymorphism to display the data  
    displayable.getInfo();  
  
    Reservasi r = new Reservasi();  
    r.setVisible(true);;  
}
```

Hasil Tampilan:



Gambar 8.4 Tampilan Hasil *Running* GUI

Source Code Object Class/Abstract (Login.java) :

```
public class Login {  
    private String username, password;  
    public String nama;  
  
    public Login()  
    {  
        nama = "Tesa";  
        username = "tesalonika";  
    }  
}
```



```
        password = "12345";
    }

    public String getUsername()
    {
        return username;
    }
    public void setUsername(String username)
    {
        this.username = username;
    }
    public String getPassword()
    {
        return password;
    }
    public void setPassword(String password)
    {
        this.password = password;
    }

    boolean    CekLogin(String    username,    String
password)
    {
        if(username.equals(getUsername())    &&
password.equals(getPassword()))
        {
            return true;
        }
        return false;
    }
}
```

Desain form (Login.java) :

LOGIN

Nama

Password

Gambar 8.5 Desain Gui Login .java

Tabel 8.3 Properti Desain Login.java

No	Nama Komponen	Properti	Value
1	jLabel1	Text	Login
2	jLabel2	Text	Nama :
3	jLabel3	Text	Password
4	jTextField1	Name	txtnama
		Text	“ “

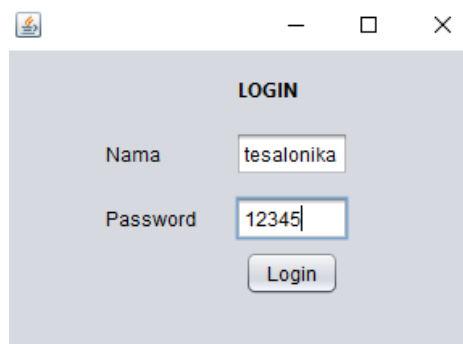


5	jTextField2	Name	txtPass
		Text	“ “
6	jButton1	Name	txtPass
		Text	Login

Source code Button/combobox btnLogin :

```
private void  
btnLoginActionPerformed(java.awt.event.ActionEvent  
evt) {  
    // TODO add your handling code here:  
    Login l = new Login();  
    String username = txtNama.getText();  
    String password = txtPass.getText();  
    boolean Authenticated =  
l.CekLogin(username, password);  
    if (Authenticated)  
    {  
        JOptionPane.showMessageDialog(rootPane,  
"LOGIN BERHASIL, "+l.nama + "!");  
        Wisata w = new Wisata();  
        w.setVisible(true);  
        this.dispose();  
    }else  
    {  
        JOptionPane.showMessageDialog(rootPane,  
"LOGIN GAGAL. Silahkan periksa kembali username dan  
password Anda.");  
    }  
}
```

Hasil Tampilan :



Gambar 8.6 Hasil Tampilan Login.java

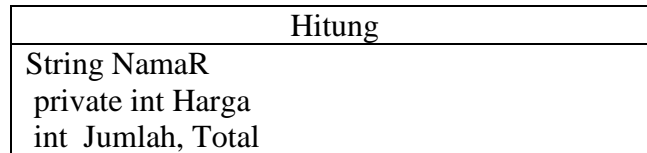
Analisa:

Program diatas merupakan tampilan gui wisata dari class list,pada bagian ini akan tampil nama,kota dan deskripsi.pada tampilan gui kedua tampilan login.pada login ada nama dan password.Pada bagian class list yang menjadi private yaitu kota dan pada bagian login private String username, password.Pada class list terdapat Overriding dan berfungsi mendefinisikan kembali method superclass-nya dan dapat di modifikasi kembali.Pada bagian atas



juga kita membuat menu bar yang bisa tersambung dengan gui login,wisata dan reservasi.

Class diagram (Class Hitung.java):



Source code Object Class (Hitung.java):

```
public class Hitung extends Data{
    String NamaR;
    private int Harga;
    int  Jumlah, Total;

    void dataNama(String NamaR)
    {
        this.NamaR = NamaR;
    }
    public int getHarga()
    {
        return Harga;
    }
    public void setHarga(int Harga)
    {
        this.Harga = Harga;
    }

    public Hitung(){
        this.Harga = 25000;
        this.Jumlah = Jumlah;
        this.Total = Total;
    }
    public int Total()
    {
        Total = (Harga * Jumlah);
        return Total;
    }
}
```

Source code Object Class (Info.java):

```
public interface Info {
    void getInfo();
}
```

Desain *form* (GUI_Reservasi.java):



Gambar 8.7 Desain Gui Reservasi.javaTabel

Tabel 8.4 Properti Desain GUI_Reservasi.java

No	Nama Komponen	Properti	Value
1	jLabel1	Text	Reservasi
2	jLabel2	Text	Nama
3	jLabel3	Text	Harga Tiket
4	jLabel4	Text	Jumlah Orang
5	jTextField1	Name	txtR
		Text	“ “
6	jTextField2	Name	txtHarga
		Text	“ “
7	jTextField3	Name	txtHarga
		Text	“ “
8	jButton1	Name	btnHitung
		Text	Hitung
9	jScrollPane1	Name	memo
		Text	

Source code Button btnHitung :

```
private void
btnHitungActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    memo.setText("");
    Hitung h = new Hitung();
    h.dataNama(txtR.getText());
    h.Jumlah = Integer.parseInt(txtJumlah.getText());

    memo.append("                Reservasi\n");
    memo.append("Nama : "+h.NamaR+"\n");
    memo.append("Jumlah Orang : "+h.Jumlah+"\n");
    memo.append("Total : "+Integer.toString(h.Total()));
}
```




```
}
```

Hasil Tampilan:

RESERVASI

Nama

Harga Tiket

Jumlah Orang

Reservasi
Nama : Tesa
Jumlah Orang : 4
Total :100000

Gambar 8.8 Desain Gui Reservasi.javaTabel

Analisa:

Polimorfisme adalah salah satu konsep dalam pemrograman komputer yang memungkinkan suatu entitas, seperti fungsi, operator, atau objek, untuk dapat memiliki beberapa bentuk atau perilaku yang berbeda. Dengan kata lain, polimorfisme memungkinkan suatu entitas untuk bersifat banyak bentuk. Sebagai contoh pada Program diatas merupakan implementasi dari class hitung yang di hubungkan gui reservasi. Tampil yang muncul nama pemesan, harga tiket dan jumlah orang. Setelah itu akan muncul di memo sesuai apa yang di inputkan. Pada class ini yang di jadikan private yaitu Harga. Pada bagian ini juga kita buat class baru yaitu kelas data yang berisi Info dan Info di buat interface . Pada gui wisata terjadi masalah yang membuat yang membuat memo tidak keluar.



8.2 Kesimpulan

1. Interface merupakan sebuah antarmuka ,yang secara umum interface berfungsi sebagai penghubung antar sesuatu yang ‘Abstract’ dengan sesuatu yang nyata. Terdapat istilah ‘able’ atau ‘mampu’ yaitu istilah yang sering digunakan untuk menunjukkan bahwa , objek yang mengimplementasikan interface tersebut memiliki sesuatu atau memiliki kemampuan tertentu. Dalam OOP, sebuah interface dapat dianggap sebagai prototipe atau templat untuk sebuah kelas. Analoginya, jika sebuah class abstrak adalah kerangka dasar untuk kelas-kelas lain, maka interface adalah templat yang memberikan struktur khusus untuk kelas tertentu.
2. Interface secara struktur mirip dengan class, tetapi dengan perbedaan penting. Isi dari interface terdiri dari deklarasi method yang bersifat abstrak, yaitu metode yang hanya didefinisikan tanpa isi (badan metode). Deklarasi metode dalam interface mirip dengan deklarasi metode pada kelas abstrak. Ketika kita membuat sebuah method dalam interface maka otomatis sama dengan kita membuat class abstract. variabel yang dideklarasikan dalam interface otomatis dianggap sebagai static (artinya, terkait dengan kelas itu sendiri, bukan instance) dan final (artinya, nilainya tidak dapat diubah setelah diinisialisasi). Sedangkan Metode dalam interface secara otomatis dianggap sebagai public dan abstract.
3. Sintak untuk menciptakan interface serupa dengan cara menciptakan sebuah class tetapi terdapat beberapa pengecualian, yaitu : Seluruh method yang di deklarasikan pada Interface pasti bersifat public dan abstract. Variable selalu bersifat public, final dan static. Bentuk Deklarasi Interface. Struktur multiple implements terdapat yaitu Sebuah class dapat meng extends dan implements interface secara bersamaan, Sebuah Interface dapat extends dari interface lain, dan Sebuah Class dapat extends dari kelas lain.

