TUGAS PERTEMUAN KE – 7

(PRAKTIKUM OOP 2022-202)

NAMA	Tesalonika Dua Nurak
NIM	2218015
KELAS	A
PEMBERI TUGAS	Raden Adityo Priyo Harjuno (2218093)

5.1 Tugas Rumah : Menerapkan Konsep polimorfisme

Judul : Informasi Wisata

Tema : Sistem Informasi

List
• Nama • Kota • Deskripsi • Harga • Jumlah • Total
+ void dataNama + void dataKota + void dataDeskripsi + String cetakNama + String cetakKota + String cetakDeskripsi

interface Info
void getInfo

abstract class Data
int Total

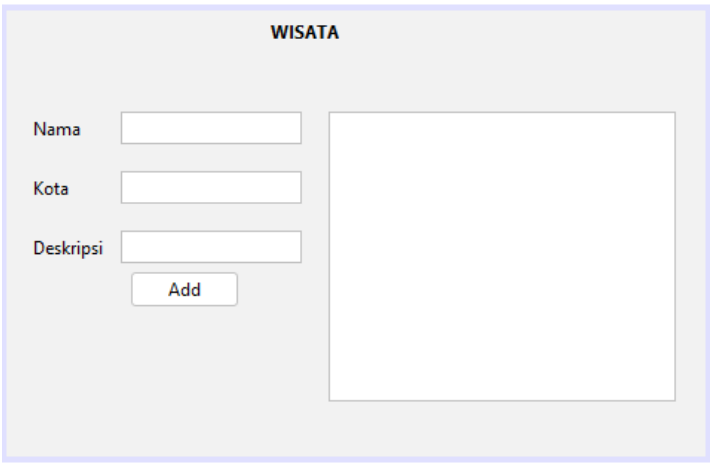
Hitung
String NamaR int Harga int Jumlah, Total

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);
    }
}
```

#PraktikumOOP2023

Desain *form* (Wisata.java):



Gambar 6.1 Desain GUI_Wisata .java

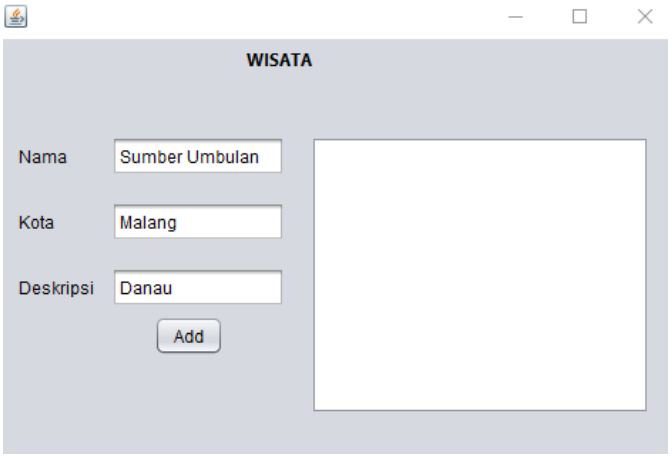
Tabel 4.1 Properti GUI_Wisata.java

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:



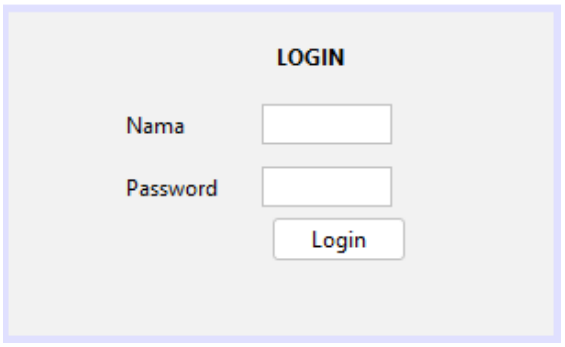
Gambar 6.2 Tampilan Hasil *Running* GUI

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) :



Gambar 6.3 Desain Gui Login .java

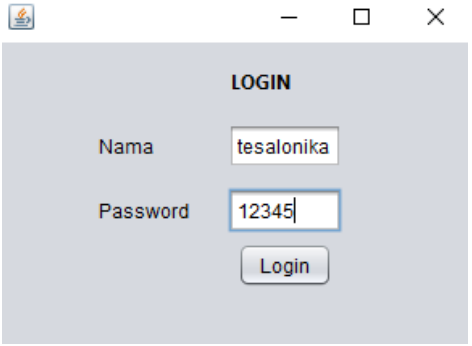
Tabel 3.2 Properti Desain Login,java

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 Text	txtPass Login

Source code Button/combobox bttnLogin :

```
    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 6.4 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.

Class diagram (Class Hitung.java):

Hitung
String NamaR private int Harga int Jumlah, Total

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;
     }

}
```

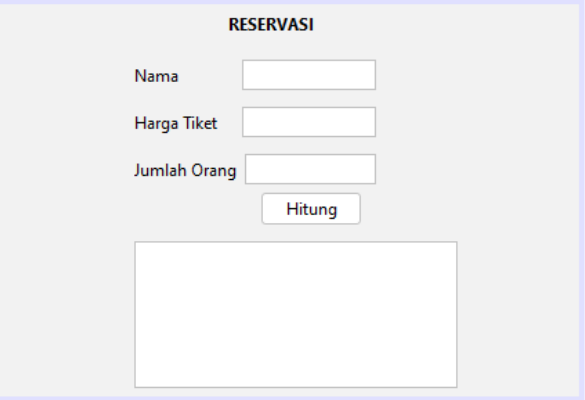
Source code Object Class (Data.java):

```
public abstract class Data {
    abstract int Total();
}
```

Source code Object Class (Info.java):

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

Desain *form* (GUI_Reservasi.java):



Gambar 6.5 Desain Gui Reservasi.javaTabel

Properti Desain GUI_Reservasi.java

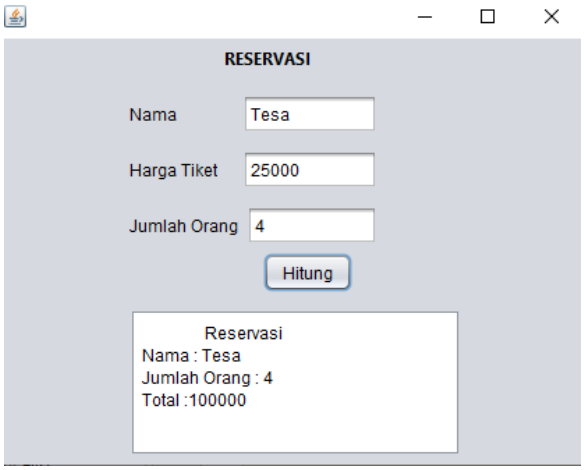
No	Nama Komponen	Properti	Value
1	jLabel1	Text	Reservasi
2	jLabel2	Text	Nama
3	jLabel3	Text	Harga Tiket
4	jLabel4	Text	Jumlah Orang
5	jTextField1	Name	txtR
		Text	“ “
6	jTextField2	Name	txtHarga
		Text	“ “
7	jTextField3	Name	txtHarga
		Text	“ “
8	jButton1	Name	btnHitung
		Text	Hitung
9	jScrollPane1	Name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

Source code Button bttnHitung :

```
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:



Gambar 6.6 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 Total dan Total di buat abstract. Pada kali ini juga ada class baru yaitu interface info yang memiliki void getInfo.Serta ada class baru yaitu data yang mengguanakan abstract.Pada gui wisata terjadi masalah yang membuat yang membuat memo tidak keluar.

5.2 Kesimpulan

1. Kata "polimorfisme" berasal dari bahasa Yunani yang berarti "banyak bentuk". Dalam konteks OOP, polimorfisme memungkinkan suatu entitas (seperti method atau function) untuk berperilaku dengan cara yang berbeda tergantung pada konteks di mana entitas(seperti method atau function) tersebut digunakan . Abstract Method adalah sebuah ‘method dasar’ yang harus direpresentasikan ulang di dalam class anak (child class). Abstract method ditulis tanpa isi dari method, melainkan hanya ‘signature’-nya saja (ciri khas). Signature dari sebuah method adalah bagian method yang terdiri dari nama method dan parameternya (jika ada).
2. Polymorphisme pada java terdiri dari 2 jenis, yaitu Static Polymorphism dan Dynamic Polymorphism yaitu Compile-time: Tipe compile-time atau static terjadi ketika metode dijalankan pada waktu kompilasi (compile-time). Tipe ini terjadi saat menjalankan metode overloading, dan Run time: Tipe runtime atau dynamic merujuk pada metode dijalankan tepat saat kode yang dapat dieksekusi mulai dijalankan. Runtime berlangsung saat metode overriding.
3. Casting adalah proses mengubah tipe data dari satu tipe ke tipe(attribut maupun method) lainnya. Casting diperlukan untuk mengubah atau menyesuaikan tipe data antara tipe data yang berbeda. Terdapat 2 jens casting yaitu Upcasting dan downcasting adalah dua konsep yang terkait dengan pewarisan (inheritance) pada pemrograman berorientasi objek yang memanfaatkan hubungan antara kelas-kelas dalam hierarki pewarisan. Kedua konsep ini akan memberikan fleksibilitas dan kemampuan untuk mengelola hubungan antara kelas-kelas dalam hierarki pewarisan.