



**TUGAS PERTEMUAN KE – 5**  
**(PRAKTIKUM OOP 2022-202)**

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

**4.1 Tugas Rumah 1 : Menerapkan Konsep Overloading Dan Overriding**

Judul : Informasi Wisata

Tema : Sistem Informasi

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

*Source code Object Class (List):*

```
public class List {  
    String Nama, Deskripsi;  
    private String Kota;  
    int Jumlah, Total;  
  
    public String getkota()  
    {  
        return Kota;  
    }  
    public void setKota(String Kota){  
        this.Kota = Kota;  
    }  
}
```



```
void dataNama(String Nama){
    this.Nama = Nama;
}
void dataDeskripsi(String Deskripsi){
    this.Deskripsi = Deskripsi;
}
}
```

Desain *form* (Wisata.java):

Gambar 4.1 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	Add
		Text	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("");
    //
    Wisata wst = new Wisata();
    List wst = new List();
    wst.dataNama(txtNama.getText());
    wst.setKota(txtKota.getText());
    wst.dataDeskripsi(txtDes.getText());

    memoData.append("List Wisata\n");
    memoData.append("-----\n");
    memoData.append(" Nama : " + wst.Nama + "\n");
    memoData.append(" Kota: " + wst.getkota() +
"\n");
    memoData.append(" Deskripsi: " + wst.Deskripsi +
"\n");

    Reservasi r = new Reservasi();
    r.setVisible(true);
}
```

Hasil Tampilan:



Gambar 4.2 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) :

The image shows a graphical user interface for a login system. It has a light gray background with a thin blue border. At the top center, the word "LOGIN" is written in bold black text. Below this, there are two labels: "Nama" and "Password", both in bold black text. To the right of each label is a white rectangular text input field with a thin gray border. Below these two input fields, centered, is a white rectangular button with a thin gray border and the word "Login" written in bold black text.

Gambar 4.3 Desain Gui Login .java



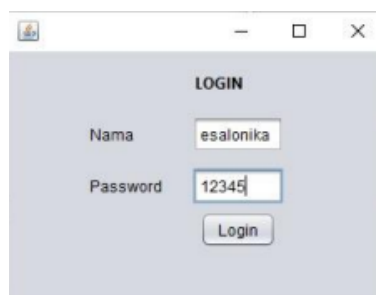
Tabel 3.2 Properti Desain Login.java

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	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 :



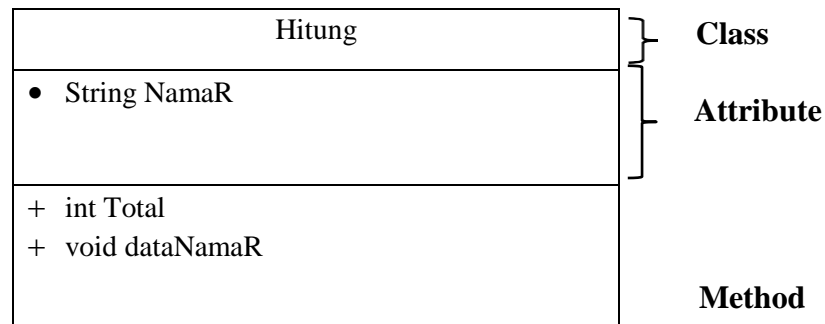
Gambar 4.4 Hasil Tampilan Login.java



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):



Source code Object Class (Hitung.java):

```
public class Hitung extends List{
    String NamaR;
    private int Harga;

    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;
    }
}
```



Desain *form* (GUI\_Reservasi.java):

Gambar 4.5 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 btnHitung :*

```
private void
btnHitungActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    memo.setText("");
    Hitung h = new Hitung();
    h.dataNamaR(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 4.6 Desain Gui Reservasi.javaTabel

Analisa:

Program diatas merupakan implementasi dari class hitung yang di hubungkan gui reservasi.Tampil yang amuncul 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 class overloading terdapat pada class hitung yang berfungsi untuk berfungsi kemampuan dari subclass untuk memodifikasi methoddarisuperclass.

## 4.2 Kesimpulan

1. Method Overloading adalah sebuah kemampuan yang membolehkan sebuah class mempunyai 2 atau lebih method dengan nama yang sama. Dengan menggunakan method overloading , kita dapat membuat beberapa versi dari sebuah method dengan statemen yang berbeda berdasarkan parameter yang diberikan padanya
2. Overriding method adalah kemampuan dari subclass( child class ) untuk memodifikasi method dari superclass, dengan cara mendefinisikan kembali method superclass-nya.
3. Aturan Overriding terdapat dua yaitu Aturan Overriding Parameter yang terdapat pada method Overriding di subclass harus sama dengan parameter yang terdapat pada parent class, dan method pada subclass class yang akan di override, sebaiknya tidak menggunakan hak akses private, karena method pada subclass tidak boleh membatasi aksesibilitas yang lebih luas dari pada method pada parent class.



