



**TUGAS PERTEMUAN KE – 6**  
**(PRAKTIKUM OOP 2022-202)**

<b>NAMA</b>	Tesalonika Dua Nurak
<b>NIM</b>	2218015
<b>KELAS</b>	A
<b>PEMBERI TUGAS</b>	Muhammad Harifin (2118131)

**5.1 Tugas Rumah 1 : Menerapkan Konsep Abstract**

Judul : Informasi Wisata

Tema : Sistem Informasi

List	} Class
<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;

    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 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	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 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) :

The image shows a simple Java Swing window titled "LOGIN". Inside the window, there are two labels: "Nama" and "Password". Each label is positioned to the left of a rectangular text input field. Below these two input fields, centered horizontally, is a button labeled "Login". The entire window has a light gray background and a thin blue border.

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	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 :

LOGIN

Nama tesalonika

Password 12345

Login

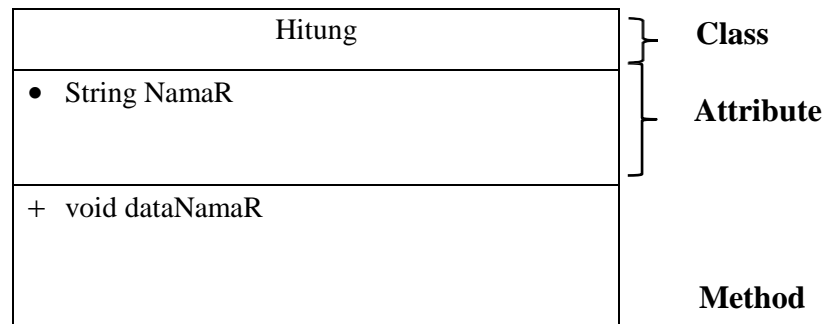
Gambar 6.4 Hasil Tampilan Login.java



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):



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();  
}
```

*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 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 6.6 Desain Gui Reservasi.javaTabel

Analisa:

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. Abstract class adalah sebuah konsep dalam pemrograman berorientasi objek (OOP) yang digunakan untuk membuat kelas dasar yang tidak dapat diinstansiasi sendiri. Sebagai contoh class Hitung dan class Data diatas.





## 5.2 Kesimpulan

1. Abstract Class adalah sebuah class yang tidak bisa di-instansiasi (tidak bisa dibuat menjadi objek) , berperan sebagai ‘kerangka dasar’ bagi class turunannya. Di dalam sebuah abstract class setidaknya memiliki satu atau lebih method abstrak. Fungsi dari class abstract ini adalah untuk mempertahankan hirarky dari parent class ke kelas turunan dari induknya. Abstract class digunakan di dalam inheritance (pewarisan class) untuk ‘memaksakan’ implementasi method yang sama bagi seluruh class yang diturunkan dari abstract class.
2. 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).
3. Keuntungan Penggunaan Abstract Class yaitu Struktur Hierarki Membantu dalam membangun struktur hierarki kelas, di mana kelas abstrak dapat berperan sebagai titik pusat untuk kelas-kelas yang lebih khusus. Pewarisan (Inheritance): Memfasilitasi pewarisan kelas. Kelas turunan dapat mewarisi sifat dan perilaku dari kelas abstrak

