**TUGAS PERTEMUAN KE – 3**

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

**(PRAKTIKUM OOP 2022-2023)**

## Tugas Rumah 1 :

Judul : Sitem Informasi Wisata

Tema : Sistem Informasi

Class diagram (Class List.java):

|  |
| --- |
| List  **Class** |
| * Kota   **Attribute**   * Nama * Deskripsi |
| * oid dataNama * Data Kota () * void dataDeskripsi   **Method**   * String cetakNama * cetakKota () * cetakDeskripsi () |
|  |

Class diagram (Nama\_Reservasi.java):

**Class**

|  |
| --- |
| Nama Reservasi |
| * Reservasi   **Attribute** |
| * void dataReservasi   **Method** |
| * String cetakReservasi()   **konstruktor** |

Class diagram (Biaya\_Masuk.java):

|  |
| --- |
| Biaya Masuk  **Class** |
| * Add HargaTiket * Add Jumlah   **Attribute**   * Add Total |
| * int HargaTiket * int Jumlah   **Method**   * int Total |
| * int HargaTiket() * int Jumlah()   **konstruktor**   * int Total() |

*Source code Object Class/Abstact*(List.java):

|  |
| --- |
| public class List {  String Nama, Kota, Deskripsi;    void dataNama(String Nama){  this.Nama = Nama;  }  void dataKota(String Kota){  this.Kota = Kota;  }  void dataDeskripsi(String Deskripsi){  this.Deskripsi = Deskripsi;  }  String cetakNama(){  return Nama;  }  String cetakKota(){ return Kota;  }  String cetakDeskripsi(){ return Deskripsi;  }    } |

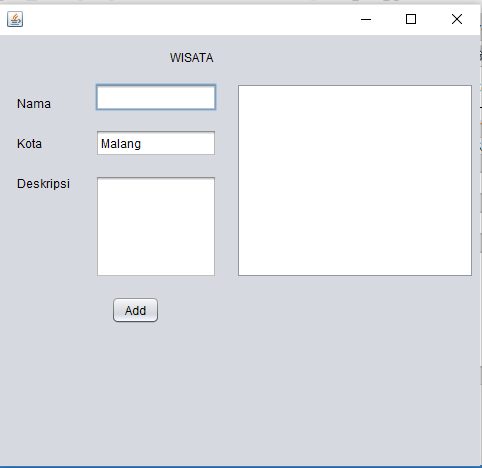
*Source code Object Class/Abstact*(Nama\_Reservasi.java):

|  |
| --- |
| public class Nama\_Reservasi extends List {  String Reservasi;    void dataReservasi(String Reservasi){  this.Reservasi = Reservasi;    }  String cetakReservasi(){  return Reservasi;  }    } |

*Source code Object Class/Abstact*(Biaya\_Masuk.java):

|  |
| --- |
| public class Biaya\_Masuk extends List{  int HargaTiket,Jumlah,Total;  int HargaTiket(){  return HargaTiket;  }  int Jumlah(){  return Jumlah;  }  int Total(){  return Total;  }  public int List(){  Total = (HargaTiket \* Jumlah);  return Total;  }    } |

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



Gambar 1.2 Desain GUI\_Wisata.java

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

Gambar 1.2 Desain GUI\_Wisata.java

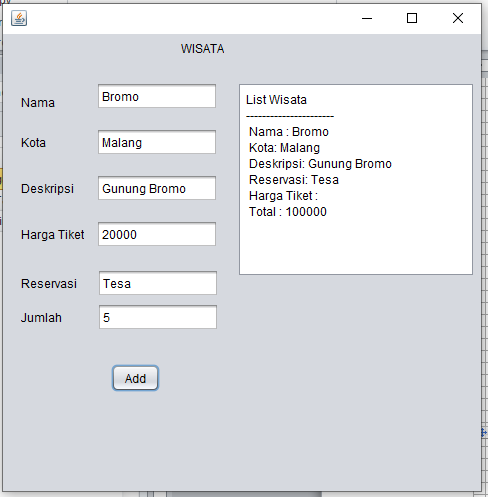
#### Tabel 1.1 Properti Desain GUI\_Wisata.java

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 | btnDta | |
| 9 | jTextArea | | Name | - | |
| Text | memoData | |
| 10 | jScrollPane1 | | - | - | |
| 11 | jLabel5 | | Text | Harga Tiket | |
| 12 | jLabel6 | | Text | Jumlah | |
| 13 | jTextField4 | | Name | HargaTiket | |
| Text | “ “ | |
| 14 | jTextField5 | | Name | jumlah | |
| Text | “ “ | |
| 15 | jLabel7 | Text | | | Reservasi |
| 16 | jTextField | Name | | | txtR |
| Text | | | “ ” |

*Source code Button/combobox* :

|  |
| --- |
| 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.dataKota(txtKota.getText());  wst.dataDeskripsi(txtDes.getText());    Nama\_Reservasi nr = new Nama\_Reservasi();  nr.dataReservasi(txtR.getText());    Biaya\_Masuk bm = new Biaya\_Masuk();  bm.HargaTiket = Integer.parseInt(HargaTiket.getText());  bm.Jumlah = Integer.parseInt(jumlah.getText());  memoData.append("List Wisata\n");  memoData.append("----------------------\n");  memoData.append(" Nama : " + wst.Nama + "\n");  memoData.append(" Kota: " + wst.Kota + "\n");  memoData.append(" Deskripsi: " + wst.Deskripsi + "\n");  memoData.append(" Reservasi: " + nr.Reservasi + "\n");  memoData.append(" Harga Tiket : " +"\n");  memoData.append(" Total : "+bm.List());  } |

Hasil Tampilan Gui Wisata:



Gambar 1.3 Tampilan Hasil GUI\_Wisata.java

Analisa:

Program akan meminta untuk menginputkan nama,lokasi wisata,deskripsi tempat tersebut bertipe string dan juga menginput harga tiket,jumlah, dan total yang bertipe int.Kemudian untuk memanggil text area diperlukan variable dari text area berupa memomabe. Karena yang sebelumnya system.out berubah menjadi variable dari text area.

## Kesimpulan

* + - 1. OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berbasis kepada objek.Class merupakan rancangan dari sebuah objek yang mendefenisikan atribut (ciri/variabel) dan method (perilaku) umum dari suatu objek yang dibuat.Atribut adalah karakteristik unik atau ciri dari sebuah objek. Karakteristik tersebut dapat berupa data/variabel yang akan dimiliki oleh objek dari kelas tersebut.
      2. Konstruktor (constructor) adalah metode khusus dalam pemrogramanberorientasi objek yang digunakan untuk menginisialisasi objek dari sebuah kelas. Konstruktor biasanya digunakan untuk mengatur nilai awal atau status awal dari objek tersebut. Konstruktor memiliki nama yang sama dengan nama kelasnya dan tidak mengembalikan nilai, biasanya digunakan untuk menginisialisasi atribut-atribut objek.
      3. Inheritance atau Pewarisan/Penurunan adalah konsep pemrograman dimana sebuah class dapat ‘menurunkan” properti dan method yang dimilikinya kepada class lain. Konsep inheritance ialah membuat sebuah struktur atau ‘hierarchy’ class dalam kode program, hal tersebut memungkinkan untuk melakukan pewarisan attribute atau method dari class yang umum ke class yang lebih spesifik.