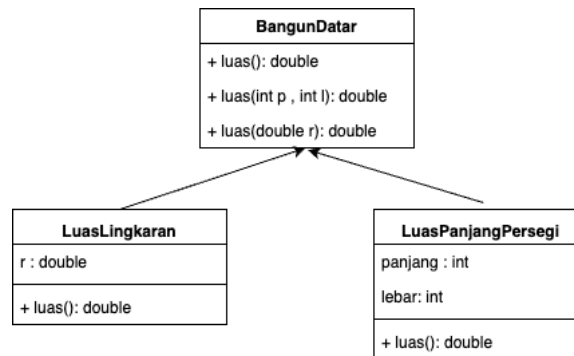




A. Latihan Overloading dan Overriding

Class Diagram



1. Buat Package baru bernama “BAB5.Override_Overload” pada project “PraktikumOOP_NIM”, dengan cara klik kanan pada Source Package → New → Java Package → masukkan nama package (BAB5.Override_Overload) → klik Finish.
2. Buatlah Class baru dengan nama “BangunDatar” dalam package BAB5.Override_Overload, lalu tambahkan source code seperti berikut:

Source Code:

```
public class BangunDatar {
    public double Luas() {
        return 0;
    }

    public int Luas(int p, int l){
        return ( p * l );
    }

    public double Luas(double r){
        return (( r * r ) * 3.14);
    }
}
```

3. Buatlah 2 Class baru dengan nama “LuasPanjangPersegi” dan “LuasLingkaran” dalam package BAB5.Override_Overload, lalu tambahkan source code seperti berikut:

Source Code LuasPanjangPersegi :

```
public class LuasPersegiPanjang extends BangunDatar{
    int panjang;
    int lebar;

    @Override
    public double Luas() {
        return Luas( p: panjang, l: lebar);
    }
}
```



Source Code LuasLingkaran:

```
public class LuasLingkaran extends BangunDatar{  
    double r;  
  
    @Override  
    public double Luas() {  
        return Luas(r);  
    }  
}
```

4. Buatlah Desain GUI dengan nama “GUI_BangunDatar” seperti pada halaman di bawah ini.

LUAS BANGUN DATAR

L. Persegi Panjang ▼

Panjang

Lebar

Proses

Hasil

5. Terapkan method overloading kedalam GUI_BangunDatar ketika memilih L.Persegi Panjang maka akan memproses method “Luas(int p, int l)” dan jika memilih L.Lingkaran maka akan menjalankan method “Luas(Int r)”.
6. Aturan aktifasi label dan text field:
 - a. Ketika memilih L.Lingkaran
 - Label “Panjang” berubah menjadi “Jari-jari” , label “Lebar” non-aktif
 - Text field lebar non-aktif
 - b. Ketika memilih L.Persegi Panjang
 - Label “Panjang” aktif , label “Lebar” aktif
 - Text field lebar aktif
7. Jalankan Program hingga hasilnya seperti gambar di bawah ini.



B. Project Praktikum (Mendesain ulang GUI_Mahasiswa + method Batal)

1. Pastikan desain GUI_Mahasiswa tampak seperti gambar di bawah ini.

2. Lalu rubah desain menjadi seperti gambar di bawah ini.

Tabel Properti:

No	Komponen	Properti	Nilai
1	jLabel1	Text	DATA MAHASISWA



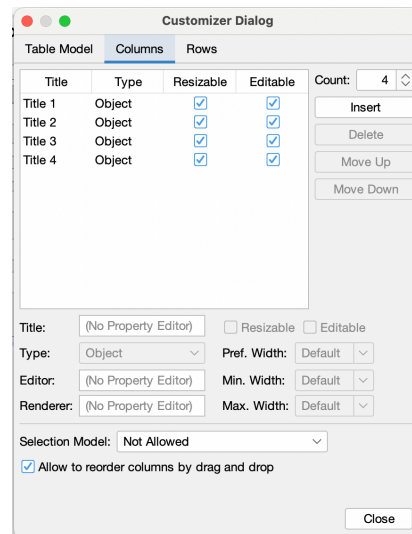
2	jLabel2	Text	NIM
3	jLabel3	Text	Nama
4	jLabel4	Text	Jenis Kelamin
5	jLabel5	Text	Prodi
6	jLabel6	Text	Angkatan
7	jLabel7	Text	Alamat
8	jLabel8	Text	Search
9	jTextField1	Name	txtNim
		Text	
10	jTextField2	Name	txtNama
		Text	
11	jTextField3	Name	txtProdi
		Text	
12	jTextField4	Name	radiobtnLaki
		Text	Laki-laki
13	jTextField5	Name	radiobtnPerempuan
		Text	Perempuan
14	jTextField6	Name	txtAngkatan
		Text	
15	jTextField7	Name	txtAlamat
		Text	
16	jTextField8	Name	txtSearch
		Text	
17	jButton1	Name	btnSimpan
		Text	Simpan
18	jButton2	Name	btnHapus
		Text	Hapus
19	jButton3	Name	btnClose
		Text	Close
20	jButton4	Name	btnBatal
		Text	Batal



21	jButton5	Name	btnPenilaian
		Text	Form Penilaian
22	jTable	Name	table_data_mahasiswa
		Text	



3. Untuk merubah *field* pada *table* yaitu klik kanan pada table , kemudian tab Table Contents lalu klik Columns dan ubah title dari field table.



4. Masukkan Script berikut diantara nama package dan project praktikum

```
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableColumnModel;
import javax.swing.table.DefaultTableModel;
```

5. Masukkan Source code berikut pada constructor GUI_Mahasiswa

```
public GUI_Mahasiswa() {
    initComponents();
    // Mengambil model data dari tabel dan menyimpannya dalam objek DefaultTableModel dataModel
    DefaultTableModel dataModel = (DefaultTableModel) table_data_mahasiswa.getModel();
    // Mendapatkan jumlah baris yang ada dalam model data saat ini
    int rowCount = dataModel.getRowCount();
    while (rowCount > 0) {
        // Menghapus baris terakhir dari model data
        dataModel.removeRow(rowCount - 1);
        // Memperbarui nilai rowCount setelah penghapusan baris terakhir
        rowCount = dataModel.getRowCount();
    }
}
```

6. Tambahkan script berikut di bawah method konstruktor dari GUI_Mahasiswa.

Source Code:

```
public void clear() {
    txtNim.setText("");
    txtNama.setText("");
    txtProdi.setText("");
    txtAngkatan.setText("");
    txtAlamat.setText("");
    btnGroupJk.clearSelection();
}
```



7. Ubah source code btnSimpan , seperti berikut

```
// Menampilkan pesan dialog bahwa data telah ditambahkan ke
tabel
JOptionPane.showMessageDialog(null, "Data anda
Ditambahkan Ke tabel");
// Mengambil model data dari tabel
DefaultTableModel dataModel = (DefaultTableModel)
table_data_mahasiswa.getModel();
// Inisialisasi sebuah ArrayList bernama 'list'
List list = new ArrayList<>();
// Mengatur tabel untuk membuat kolom dari model secara
otomatis

table_data_mahasiswa.setAutoCreateColumnsFromModel(true);
// Membuat instance dari kelas Mahasiswa
Mahasiswa mhs = new Mahasiswa();
// Mengatur data NIM menggunakan nilai dari komponen
txtNim
mhs.dataNIM(txtNim.getText());
// Mengatur data Nama menggunakan nilai dari komponen
txtNama
mhs.dataNama(txtNama.getText());
// Mengatur jenis kelamin sesuai dengan radio button
yang dipilih
String JenKel = "";
if (radiobtnLaki.isSelected()) {
    mhs.dataJenisKelamin(radiobtnLaki.getText());
} else {
    mhs.dataJenisKelamin(radiobtnPerempuan.getText());
}
// Mengatur data Prodi menggunakan nilai dari komponen
txtProdi
mhs.dataProdi(txtProdi.getText());
// Mengatur data Angkatan menggunakan nilai dari
komponen txtAngkatan
mhs.dataAngkatan(txtAngkatan.getText());
// Mengatur data Alamat menggunakan nilai dari komponen
txtAlamat
mhs.dataAlamat(txtAlamat.getText());
// Menambahkan data-data dari objek Mahasiswa ke dalam
ArrayList 'list'
list.add(mhs.cetakNIM());
list.add(mhs.cetakNama());
list.add(mhs.cetakJenisKelamin());
list.add(mhs.cetakProdi());
list.add(mhs.cetakAngkatan());
list.add(mhs.cetakAlamat());
// Menambahkan baris baru ke model tabel menggunakan
data dari ArrayList 'list'
dataModel.addRow(list.toArray());
// Memanggil fungsi 'clear' untuk membersihkan nilai
dari komponen
clear();
```



8. Pilih Tab Desain, klik 2 kali pada button Close, tambahkan script berikut

Source Code:

```
private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    //exit  
    dispose();  
}
```

9. Pilih Tab Desain, klik 2 kali pada button Batal, tambahkan script berikut

Source Code:

```
private void btnBatalActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    clear();  
}
```

10. Pilih Tab Desain, klik 2 kali pada button Hapus, tambahkan script berikut

Source Code:

```
private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    DefaultTableModel dataModel = (DefaultTableModel) table_data_mahasiswa.getModel();  
    int rowCount = dataModel.getRowCount();  
    while (rowCount > 0) {  
        dataModel.removeRow(rowCount - 1);  
        rowCount = dataModel.getRowCount(); // Update  
    }  
}
```

11. Running program hingga hasilnya ketika semua textfield di isi dan jika button “Simpan” ditekan.

NIM	Nama	JK	Prodi	Angkatan	Alamat
2118112	Firman	Laki-laki	Teknik Info...	2021	Malang

Tugas 5

1. Implementasikan Overloading dan overriding dari class yang sudah dibuat sebelumnya, Sesuai Dengan Tema masing-masing.
2. Ubah textarea dari GUI Sesuai Tema masing-masing, menjadi tampilan table.
3. Tambahkan juga Button (Simpan, Hapus, Batal, Close), jika ada button yang tidak diperlukan tidak usah di tampilkan.



NB: Memberikan dokumentasi pada tiap source code program akan menambahkan poin plus, dokumentasi dapat berupa komentar pada tiap source code.