

A. Langkah Implementasi Abstract.

1. Buat Package baru bernama “BAB6.Abstract” pada project “PraktikumOOP_NIM”, dengan cara klik kanan pada Source Package → New → Java Package → masukkan nama package (BAB6.Abstract) → klik Finish.
2. Buatlah Class baru dengan nama “BangunDatarSegiEmpat” sebagai class class dalam package BAB6.Abstract, Lalu tambahkan perintah “abstract” di antara perintah public dan class seperti berikut :

```
12 public abstract class BangunDatar {
```

Hal ini dilakukan untuk mengubah yang semulanya Class “BangunDatarSegiEmpat” adalah Class biasa menjadi Class Abstract.

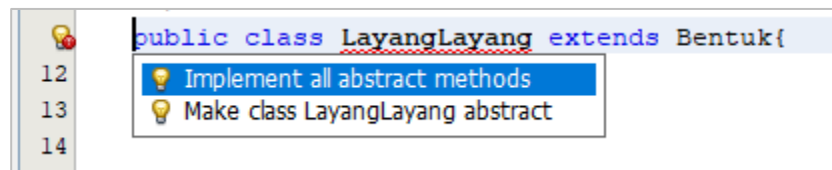
3. Buatlah abstract method pada abstract class BangunDatarSegiEmpat seperti berikut :

```
abstract double luas();  
abstract double keliling();
```

4. Lalu buat Class baru dengan nama “Jajargenjang” dan “LayangLayang” sebagai turunan dari Class “BangunDatarSegiEmpat”.

```
11 public class JajajarGenjang extends BangunDatar{  
11 public class LayangLayang extends BangunDatar {
```

Dalam gambar diatas terlihat peringatan kesalahan dikarenakan syarat utama Abstract Class adalah semua method di dalamnya **harus** di implementasikan maka klik tanda error lalu laukan seperti gambar di bawah.



Maka secara Otomatis Class “LayangLayang” Akan meng-Override semua method dari Class “BangunDatarSegiEmpat”.

```
11 public class LayangLayang extends BangunDatar {  
12  
13     @Override  
14     double luas() {  
15         throw new UnsupportedOperationException( message: "Not suppo  
16     }  
17  
18     @Override  
19     double keliling() {  
20         throw new UnsupportedOperationException( message: "Not suppo  
21     }  
22  
23 }
```

5. Kemudian, pada class LayangLayang, tambahkan source code seperti berikut:

Source code:

```
11 public class LayangLayang extends BangunDatar {
12
13     int d1, d2, a, b;
14
15     public LayangLayang() {
16         this.d1 = 5;
17         this.d2 = 6;
18         this.a = 12;
19         this.b = 35;
20     }
21
22     @Override
23     public double luas() {
24         return (0.5 * d1 * d2);
25     }
26
27     @Override
28     public double keliling() {
29         return (2 * (a + b));
30     }
31 }
```

6. Pada Class Jajargenjang, tambahkan source code seperti berikut:

Source code:

```
11 public class JajajarGenjang extends BangunDatar{
12
13     double a,t,b;
14     public JajajarGenjang(){
15         this.a = 8;
16         this.t = 3;
17         this.b = 9;
18     }
19     @Override
20     public double luas() {
21         return (a*t);
22     }
23
24     @Override
25     public double keliling() {
26         return (2*(a+b));
27     }
28 }
```

7. Buatlah class baru dengan nama “Main”, lalu tambahkan source code seperti berikut:

Source code:

```
11 public class Main {
12     public static void main(String[] args) {
13         System.out.println(x: "LUAS dan KELILING");
14         LayangLayang x = new LayangLayang();
15         System.out.println(x: "-----");
16         System.out.println("Luas Layang-layang : " + x.luas());
17         System.out.println("Keliling Layang-layang : " + x.keliling());
18         System.out.println(x: "-----");
19         JajajarGenjang y = new JajajarGenjang();
20         System.out.println("Luas Jajargenjang : " + y.luas());
21         System.out.println("Keliling Jajargenjang : " + y.keliling());
22     }
23 }
```

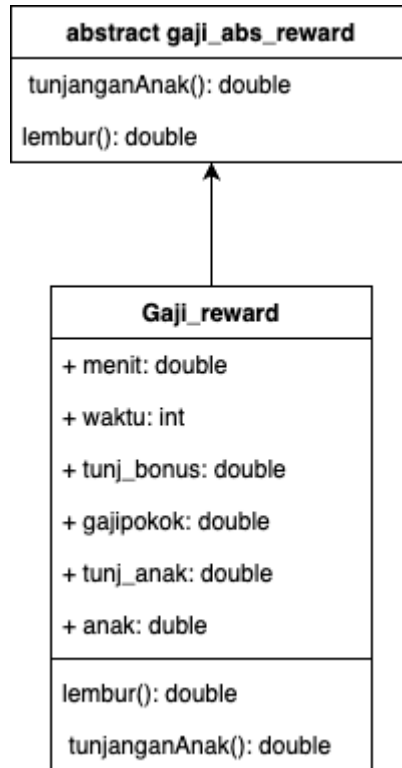
8. Jalankan program hingga hasilnya seperti gambar di bawah ini.

```
Output - PraktikumOOP_2118112 (run)

run:
LUAS dan KELILING
-----
Luas Layang-layang : 15.0
Keliling Layang-layang : 94.0
-----
Luas Jajargenjang : 24.0
Keliling Jajargenjang : 34.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

B. Latihan Abstract

1. Buatlah Class abstract dengan nama “Gaji_abs_reward”, lalu tambahkan atribut dan method seperti *class diagram* berikut:



2. Buat Class baru (JFrame Form) dengan nama “GUI_reward”. Desain GUI Seperti di bawah.

The screenshot shows a Java Swing window titled "Reward Pegawai". Inside the window, there are four text labels on the left and four corresponding text input fields on the right. The labels are "Gaji pokok", "Jumlah Anak", "Lembur(Menit)", and "Bonus pegawai". Below the "Lembur(Menit)" input field, there is a button labeled "Proses".

3. Pada method tunjangan di class Gaji_reward
 - a. Jika anak berjumlah 1 maka dia berhak mendapatkan 10% dari gaji pokok
 - b. Jika anak lebih dari 1 , maka dia berhak mendapatkan 20% dari gaji pokok.

4. Pada method lembur di class Gaji_reward
 - a. Buat konversi dari menit ke jam
 - b. Hasil konversi dari menit ke jam dikali Rp. 25.000
 - c. Bonus pegawai hasil dari tunj_anak + lembur
5. Outputkan ke dalam GUI

Reward Pegawai

Gaji pokok	4500000
Jumlah Anak	1
Lembur(Menit)	60
<input type="button" value="Proses"/>	
Bonus pegawai	475000.0

C. Project Praktikum (Mendesain ulang GUI_Penilaian)

1. Pada package “ProjectPraktikum_Nim”, buka class Penilaian(Bab4), kemudian ubah menjadi class abstract.
2. Ubah method nilaikeaktifan di dalam penilaian menjadi method abstract seperti berikut:

Source code :

```
//method abstract  
abstract double nilaiKeaktifan();
```

3. Kemudian Override method abstract yang sudah dibuat di class penilaian ke class KeaktifanMahasiswa.

Source Code :

```
public class KeaktifanMahasiswa extends Penilaian{  
    int nilai_keaktifan;  
    public KeaktifanMahasiswa(){  
        this.nilai_keaktifan = 0;  
    }  
    @Override  
    public double nilaiKeaktifan(){
```

```

        return ((nilai_keaktifan* 0.1) + nilaiAkhir());
    }
}

```

4. Ubah Tampilan GUI_penilaian(Bab 4)

Sebelum :

Sesudah :

Tabel komponen :

No	Objek	Properti	Nilai
1	jLabel1	Text	PROGRAM PENILAIAN
2	jLabel2	Text	NIM
3	jLabel3	Text	Nama
4	jLabel4	Text	Kode Matakuliah

5	jLabel5	Text	NP1
6	jLabel6	Text	NP2
7	jLabel7	Text	NP3
8	jLabel8	Text	UTS
9	jLabel9	Text	UAS
10	jLabel10	Text	Praktikum
11	jLabel11	Text	Nilai Keaktifan
12	jTextField1	Name	txtNIM
		Text	
13	jTextField2	Name	txtNama
		Text	
14	jTextField3	Name	txtKodeMK
		Text	
15	jTextField4	Name	txtNP1
		Text	
16	jTextField5	Name	txtNP2
		Text	
17	jTextField6	Name	txtNP3
		Text	
18	jTextField7	Name	txtUts
		Text	
19	jTextField8	Name	txtPraktikum
		Text	
20	jTextField9	Name	txtUas
		Text	
21	jTextField10	Name	txtKeaktifan
		Text	
22	jButton1	Name	btnNA
		Text	Hasil Nilai Akhir

23	jTable	Name	table_penilaian_matakuliah
		Text	
24	JCheckBox	Text	Tambahkan Nilai Keaktifan
		Name	checkBox

5. Kemudian buka GUI_Penilaian , Masukkan library berikut

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

6. Masukkan juga source code berikut ke dalam constructor GUI

```
// Mengambil model data dari tabel dan menyimpannya dalam objek
DefaultTableModel dataModel

        DefaultTableModel dataModel = (DefaultTableModel)
table_penilaian_matakuliah.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(); // Update rowCount
after removal

        }

        KeaktifanMahasiswa nilai = new KeaktifanMahasiswa();

        txtKeaktifan.setText(Integer.toString(nilai.nilai_keaktifan));

        txtKeaktifan.setEnabled(false);
```

7. Buat method dibawah constructor GUI, masukkan source code berikut :

```
public void clear() {
```



```

txtNim.setText("");
txtNama.setText("");
txtKodeMK.setText("");
txtNP1.setText("");
txtNP2.setText("");
txtNP3.setText("");
txtPraktikum.setText("");
txtUts.setText("");
txtUas.setText("");
txtKeaktifan.setText("0");
}

```

8. Ubah source code pada button “Hasil nilai akhir” menjadi seperti berikut.

Source code :

```

double keaktifan;

JOptionPane.showMessageDialog(null, "Data anda Ditambahkan
Ke tabel");

// Mengambil model data dari tabel

DefaultTableModel dataModel = (DefaultTableModel)
table_penilaian_matakuliah.getModel();

// Inisialisasi sebuah ArrayList bernama 'list'
List list = new ArrayList<>();

KeaktifanMahasiswa nilai = new KeaktifanMahasiswa();

nilai.NIM = txtNim.getText();

nilai.nama = txtNama.getText();

nilai.kode_mk = txtKodeMK.getText();

nilai.setNP1(Integer.parseInt(txtNP1.getText()));
nilai.setNP2(Integer.parseInt(txtNP2.getText()));
nilai.setNP3(Integer.parseInt(txtNP3.getText()));

nilai.setNilaiPrak(Integer.parseInt(txtPraktikum.getText()));

nilai.setUTS(Integer.parseInt(txtUts.getText()));

nilai.setUAS(Integer.parseInt(txtUas.getText()));

```

```
        nilai.nilai_keaktifan =
Integer.parseInt(txtKeaktifan.getText());

        if (checkBox.isSelected()) {

            keaktifan = nilai.nilaiKeaktifan();

        } else {

            keaktifan = nilai.tampilNA();

        }

        // Mengatur tabel untuk membuat kolom dari model secara
otomatis

table_penilaian_matakuliah.setAutoCreateColumnsFromModel(true);

        list.add(nilai.NIM);

        list.add(nilai.kode_mk);

        list.add(nilai.getNP1());

        list.add(nilai.getNP2());

        list.add(nilai.getNP3());

        list.add(nilai.getNilaiPrak());

        list.add(nilai.getUTS());

        list.add(nilai.getUAS());

        list.add(keaktifan);

        // Menambahkan baris baru ke model tabel menggunakan data
dari ArrayList 'list'

        dataModel.addRow(list.toArray());

        // Memanggil fungsi 'clear' untuk membersihkan nilai dari
komponen

        clear();
```

Hasil :

Nim	KD MK	NP 1	NP 2	NP 3	UTS	PRAK	UAS	NA
2118112	IF2118	80	80	80	80	80	80	81.0

Tugas 6

1. Implementasikan Abstract di class yang sudah dibuat sebelumnya (Boleh tambah 1 class abstract, boleh menggunakan class yang sudah dibuat sebelumnya dan di buat abstract kemudian menambahkan method abstract), Sesuai Dengan Tema masing-masing. Outputnya ditampilkan dengan table. (Jangan gunakan GUI yang dikerjakan di pertemuan 5).

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