



A. Latihan Interface 1

1. Buka Latihan pembayaran di bab 7 (pembayaran).
2. Buat 2 class interface baru yaitu dapatDiscan dan dapatKembalian

```
public interface dapatDiscan {  
    void scanQris();  
}
```

```
public interface dapatKembalian {  
    abstract double kembalian(double total ,double jmlUang);  
}
```

3. Buat 3 class baru yaitu Cash (implements dapatKembalian) , Emoney(implements dapatDiscan) , dan Kredit

```
public class Cash implements dapatKembalian{  
    @Override  
    public double kembalian(double total , double jmlUang) {  
        double jmlKembalian;  
        jmlKembalian = jmlUang - total;  
        return jmlKembalian;  
    }  
}
```

```
public class Emoney implements dapatDiscan{  
    @Override  
    public void scanQris() {  
        System.out.println("Pembayaran Lewat QRcode :  
91257237523752375");  
        System.out.println("Pembayaran Berhasil!!!!!!!!!!!!!!");  
    }  
}
```

```
public class Kredit {  
    void cekKartuKredit(String norek,String input){  
        if(norek.equals(input)){  
            System.out.println("Pembayaran Kredit succes");  
        }else{  
            System.out.println("Pembayaran Gagal, kata sandi  
salah");  
        }  
    }  
}
```

4. Tambahkan attribute baru pada tiap kode member “S0001” ,”P0001” , “G0001”

```
public String InputPinKredit;
```

5. Kemudian tambahkan source code berikut pada tiap kode member “S0001” , “P0001” , “G0001” di bawah method **tampilkanMember()**

```
void jenisPembayaran(Cash cash) {  
    if (bayar >= total) {  
        System.out.println("Kembalian : " +  
cash.kembalian(total, bayar));  
    } else if (bayar <= total) {
```



```
        System.out.println("uang anda kurang");
    } else {
        System.out.println("Pembayaran Berhasil");
    }
}

void jenisPembayaran(Emoney emoney) {
    emoney.scanQris();
}

void jenisPembayaran(Kredit kredit) {
    kredit.cekKartuKredit(kode, InputPinKredit);
}
```

6. Ubah main pembayaran seperti berikut :

```
11 public class main_pembayaran {
12     public static void main(String[] args) {
13         // Upcasting: Objek dari kelas turunan diubah menjadi objek dari kelas induk
14         Pembayaran payment;
15         payment = new P0001();
16         payment.tampilkanMember();
17
18         //buat object tiap pembayaran
19         Cash cash = new Cash();
20         Kredit kredit = new Kredit();
21         Emoney emoney = new Emoney();
22
23         if (payment instanceof P0001) {
24             P0001 p0001 = (P0001) payment;
25             System.out.println("Jenis Mmber: " + p0001.member(member: p0001));
26             System.out.println("Total Pembelian: " + p0001.hitPembayaran(bayar: 500000));
27
28             // pembayaran dengan cash
29             p0001.bayar = 600000;
30             p0001.jenisPembayaran(cash);
31
32             // pembayaran dengan kartu kredit
33             // p0001.InputPinKredit = "P0001";
34             // p0001.jenisPembayaran(kredit);
35
36             // pembayaran dengan Qris
37             // p0001.jenisPembayaran(emoney);
38
39             // pembayaran dengan menggunakan saldo
40             // System.out.println("sisa saldo: " + p0001.potSaldo());
41         }
42     }
43 }
44 }
```

7. Hasil Runing :

```
Output - PraktikumOOP_2118112 (run) x
run:
Member P0001 dengan diskon 30%
Jenis Mmber: Platinum
Total Pembelian: 500000.0
Kembalian : 100000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Analisa :

Pada source code diatas terdapat 2 class interface , yang mana method pada class interface akan diturunkan pada class yang mengimplements-nya. Pada source code diatas dapatDiscan impelements pada class Emoney karena pada emoney pembayaran dapat dilakukan lewat qrcode , pada class Cash mengimplements class kembalian dimana pada pembayaran cash pastinya kemungkinan besar akan mendapatkan kembalian, sedangkan Emoney dan Kredit



pastinya kemungkinan kecil tidak ada kembalian, kemudian untuk Kredit tidak mengimplements class interface karena untuk kredit menggunakan pin untuk keamanannya, kemudian tiap kode member akan memiliki method tiap pembayaran disini digunakan sebuah polymorfisme untuk memanfaatkan jenisPembayaran kemudian di dalam method jenis pembayaran kita panggil class yang berkaitan dengan pembayaran seperti Cah , kredit , Emoney. Pada class main_pembayaran buat ke 3 object pembayaran kemudian panggil method jenispembayaran masukkan pembayaran yang kita inginkan pada parameter jenis pembayaran.

B. Latihan Interface 2

1. Buatlah 2 class Interface dengan nama “Keliling” dan “SimetriBangunDatar” .

interface Keliling
+ hitungKeliling(): double

interface SimetriBangunDatar
+ hitungSimetriputar():double

2. Buat abstract class dengan nama “BangunDatar”.

Abstract BangunDatar
+ abstract hitungluas():double
+ abstract tampilHasil() : double

3. Buat class dengan nama “PersegiPanjang”. Extends “BangunRuang”
Implementasikan “Keliling,SimetriBangunDatar”.

```
public class PersegiPanjang extends BangunDatar implements Keliling,SimetriBangunDatar{  
    ...  
}
```

4. Buat class dengan nama “SegiTigaSamaKaki”. Extends “BangunRuang”
Implementasikan “Keliling”.

```
public class SegiTigaSamaKaki extends BangunDatar implements Keliling{  
    ...  
}
```

5. Override semua method, dan masukkan source Pada class “PersegiPanjang”.

```
12 public class PersegiPanjang extends BangunDatar implements Keliling,SimetriBangunDatar{  
13     public double panjang, lebar;  
14  
15     @Override  
16     public double hitungluas() {  
17         return (panjang * lebar);  
18     }  
19     @Override  
20     public void tampilHasil() {  
21         System.out.println("Luas Persegi panjang = " + Double.toString(a.hitungluas()));  
22         System.out.println("Keliling Persegi Panjang = " + Double.toString(a.hitungKeliling()));  
23         System.out.println("Banyak simetri Putar = "+ hitungSimetriputar());  
24     }  
25     @Override  
26     public double hitungKeliling() {  
27         return (2*(panjang * lebar));  
28     }  
29     @Override  
30     public double hitungSimetriputar() {  
31         return 2.0;  
32     }  
33 }
```



6. Override semua method, dan masukkan source Pada class “SegiTigaSamaKaki”.

```
12 public class SegiTigaSamaKaki extends BangunDatar implements Keliling{
13     public double alas, tinggi;
14     public double a,b,c;
15     @Override
16     public double hitungluas() {
17         return ((alas * tinggi)/2);
18     }
19     @Override
20     public void tampilHasil() {
21         System.out.println("Luas Segitiga = "+ hitungluas());
22         System.out.println("Kelilng Segitiga = "+ hitungKeliling());
23     }
24     @Override
25     public double hitungKeliling() {
26         return (a+b+c);
27     }
28 }
```

7. Buat class Driver class dengan nama “mainpersegi”.

Masukkan *source code* berikut :

```
11 public class mainpersegi {
12     public static void main(String[] args) {
13         PersegiPanjang persegipanjang = new PersegiPanjang();
14         persegipanjang.panjang = 5;
15         persegipanjang.lebar = 6;
16         persegipanjang.tampilHasil();
17     }
18 }
```

8. Buat class Driver class dengan nama “mainsegitiga”.

Masukkan *source code* berikut :

```
11 public class mainSegitiga {
12     public static void main(String[] args) {
13         SegiTigaSamaKaki segiTigaSamaKaki = new SegiTigaSamaKaki();
14         segiTigaSamaKaki.alas = 6;
15         segiTigaSamaKaki.tinggi = 8;
16         segiTigaSamaKaki.a = 5;
17         segiTigaSamaKaki.b = 5;
18         segiTigaSamaKaki.c = 3;
19         segiTigaSamaKaki.tampilHasil();
20     }
21 }
```

9. Kemudian Jalankan program

Output - PraktikumOOP_2118112 (run)

```
run:
Luas Segitiga = 24.0
Kelilng Segitiga = 13.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output - PraktikumOOP_2118112 (run)

```
run:
Luas Persegi panjang = 30.0
Keliling Persegi Panjang = 60.0
Banyak simetri Putar = 2.0
BUILD SUCCESSFUL (total time: 0 seconds)
```



10. Tambahkan GUI

Ketentuan GUI:

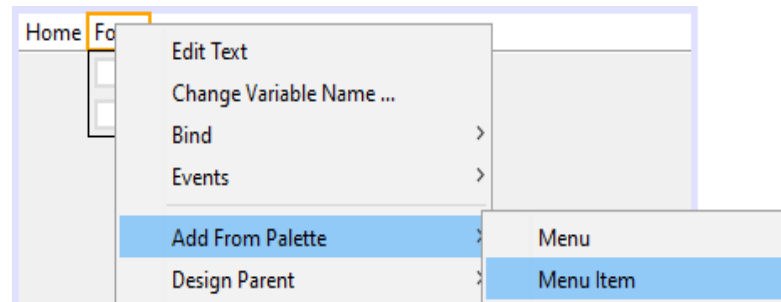
- Ketika combobox memilih “Persegi Panjang ” maka label sisi A hingga C beserta textfield otomatis akan disable(tidak bisa diisi)

- Ketika combobox memilih “Segitiga sama kaki” maka label dari Panjang dan lebar akan otomatis berganti dengan alas dan tinggi, kemudian untuk sisi A hingga C beserta textfield otomatis enable(bisa diisi)

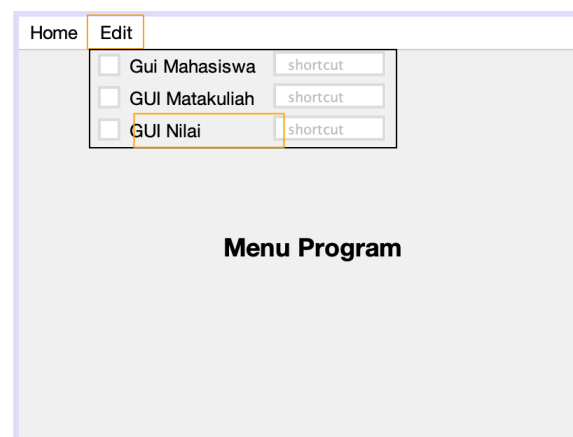


C. Project Praktikum (Membuat GUI Menu Utama).

1. Buat JFrame Form dengan nama “GUI_MenuUtama” dalam package “ProjectPraktikum”
2. Kemudian tambahkan properties Menu Bar, untuk menambah menu item, klik kanan pada tab menu → Add From Palette → Menu Item, seperti gambar berikut:



3. Buat Desain Gui seperti gambar berikut :



Tabel Properti:

No	Objek	Properti	Nilai
1	jLabel1	Text	PROGRAM SEDERHANA PENILAIAN MATA KULIAH
2	jMenuBar	jMenu	[1] = Home [2] = Form
		jMenuItem	[1] = GUI Mahasiswa [2] = GUI Mata Kuliah [3] = GUI Nilai



4. Kemudian klik 2 kali pada MenuItem “GUI Mahasiswa”, tambahkan Script berikut:

Source Code:

```
GUI_Mahasiswa mhs = new GUI_Mahasiswa();  
mhs.show();
```

5. Kemudian klik 2 kali pada MenuItem “GUI Mata Kuliah”, tambahkan Script berikut:

Source Code:

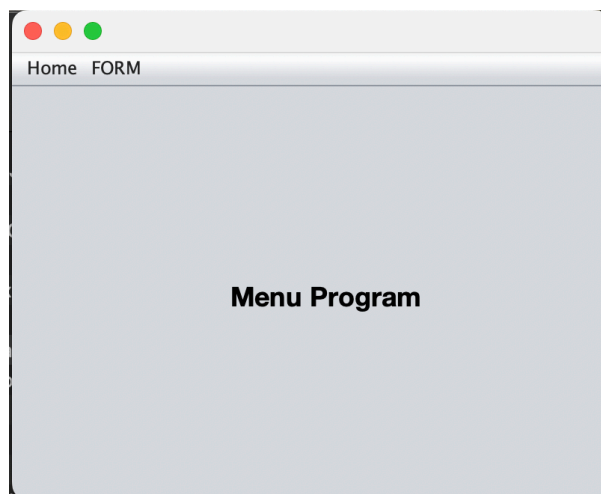
```
GUI_Matkul mk = new GUI_Matkul();  
mk.show();
```

6. Kemudian klik 2 kali pada MenuItem “GUI Nilai”, tambahkan Script berikut:

Source Code:

```
GUI_Nilai nilai = new GUI_Nilai();  
nilai.show();
```

7. Ketika di jalankan maka seperti gambar di bawah ini. Kemudian coba tab pada form dan pilih tampilan gui yang ingin diakses.





Tugas 8:

1. Implementasikan Interface pada project masing-masing.
2. Buat sebuah menu seperti contoh diatas sesuai project masing-masing.
3. Selesaikan desain GUI_Nilai, GUI_Mahasiswa dan GUI_Matkul pada project praktikum (**Bagi yang belum tuntas mengerjakan saat praktikum**)
4. Install aplikasi xampp pada link di bawah (sesuaikan dengan spesifikasi laptop masing-masing), Nb : Boleh menggunakan aplikasi untuk server local selain Xampp, seperti Laragon , namun di praktikum akan menggunakan xampp.
<https://www.apachefriends.org/download.html>
5. Silahkan masing-masing mahasiswa memiliki akun GitHub , dan atur konfigurasi secara global (cara mendaftar dan konfigurasi liat di youtube). Jangan lupa simpan email dan password akun GitHub, karena akun GitHub akan dipakai di beberapa semester depan.
6. **Silahkan Bertanya terkait nilai pada asisten masing-masing dari bab 1-6 apakah nilai sudah terinput atau belum, jika belum silahkan kirim file bab yang masih kosong pada asisten masing-masing berbentuk softfile(tidak di print). Pengiriman paling lambat tanggal (senin, 4 desember 2023) . Jika bab 1-6 masih belum mengkonfirmasi nilai yang kosong maka nilai akan diberikan 0 pada bab tersebut, dan di absensi juga.**