

## A. Langkah Implementasi Polimorfisme

1. Buat Package baru bernama “BAB7.Polimorfisme” pada project “PraktikumOOP\_NIM”, dengan cara klik kanan pada Source Package → New → Java Package → masukkan nama package (BAB7.Polimorfisme) → klik Finish.
2. Buatlah Class baru dengan nama “Buah” dalam package BAB7.Polimorfisme, seperti berikut :

Source code:

```
12 public class Buah {
13     public String nama;
14
15     public void setNamaBuah(){
16         this.nama = "Apel,mangga,pisang,jambu,dll";
17     }
18
19     void makanBuah() {
20         System.out.println( x:"Makan buah secara umum.");
21     }
22 }
```

3. Buat class dengan nama “Apel” dan “Pisang” sebagai Sub Class dari “Buah”, lalu masukan *source code* berikut :

```
11 public class Apel extends Buah {
12     public Apel() {
13         this.nama = "apel";
14     }
15     // implementasi makanBuah() khusus untuk Apel
16     void makanBuah() {
17         System.out.println(nama + " dimakan dengan di kupas kulitnya atau dimakan langsung (digigit).");
18     }
19 }
20
21 public class Pisang extends Buah {
22     public Pisang() {
23         this.nama = "Pisang";
24     }
25     // implementasi makanBuah() khusus untuk Pisang
26     void makanBuah() {
27         System.out.println(nama + " dimakan dengan di Kupas kulit pisang sebelum dimakan.");
28     }
29 }
```

4. Buat class dengan nama “Manusia”, lalu masukan *source code* berikut :

```
11 // Kelas Manusia dengan metode makanBuah() yang menerima objek buah sebagai parameter
12 public class Manusia {
13     // Metode makanBuah() yang memanfaatkan polimorfisme
14     void makanBuah(Buah buah) {
15         buah.makanBuah(); // Polimorfisme: memanggil metode makanBuah() sesuai dengan tipe objek
16     }
17 }
```

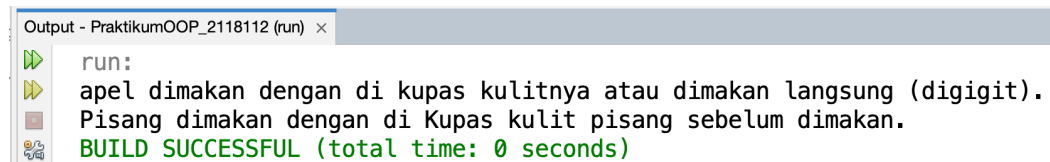
-----

-----

5. Buat class dengan nama “Main” sebagai *driver class*, lalu masukan *source code* Berikut:

```
11 public class main {
12     public static void main(String[] args) {
13         // Membuat objek Manusia
14         Manusia manusia = new Manusia();
15
16         // Membuat objek Apel dan Pisang
17         Buah apel = new Apel();
18         Buah pisang = new Pisang();
19
20         // Memanggil metode makanBuah() pada Manusia untuk objek Apel dan Pisang
21         manusia.makanBuah(buah:apel); // Output: Makan apel dengan di gigit.
22         manusia.makanBuah(buah:pisang); // Output: Kupas pisang sebelum dimakan.
23     }
24 }
```

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



```
Output - PraktikumOOP_2118112 (run) x
run:
apel dimakan dengan di kupas kulitnya atau dimakan langsung (digigit).
Pisang dimakan dengan di Kupas kulit pisang sebelum dimakan.
BUILD SUCCESSFUL (total time: 0 seconds)
```

Analisa:

Pada program tersebut Membuat poliorfisme static karena dia masih dalam satu class yang memiliki nama yang sama. Bentuk polimorfisme dari program yaitu ,manusia memiliki method makanBuah() dengan parameter object buah. parameter ini dapat menerima object yang merupakan sub-class dari buah Apel, Pisang. Barulah pada method makanBuah dideteksi, apakah buah tersebut Apel atau Pisang atau yang lain. Kemudian diperlakukan dengan berbeda (Apel bisa langsung dimakan, Pisang dikupas dahulu,dls).

## B. Latihan implementasi casting

1. Buatlah Class baru dengan nama “Pembayaran” bertipe abstract dalam package BAB7.Polimorfisme, masukkan source code seperti berikut :

```
15 public abstract class Pembayaran {
16     abstract double hitPembayaran(double bayar);
17     abstract double cekKode(String input);
18     abstract void tampilkanMember();
19
20     String member(G0001 member) {
21         return "Gold";
22     }
23
24     String member(P0001 member) {
25         return "Platinum";
26     }
27
28     String member(S0001 member) {
29         return "Silver";
30     }
31 }
```

2. Buatlah class baru dengan nama “S0001”, extends ke Pembayaran, masukkan sourcecode berikut :

```
11 public class S0001 extends Pembayaran {
12     public String name, alamat;
13     public int saldo;
14     public String kode;
15     double diskon, bayar, total;
16     public S0001() {
17         this.name = "Mohammad Harifin";
18         this.kode = "S0001";
19         this.saldo = 500000;
20         this.alamat = "Malang, jl.singosari no 23";
21     }
22     public double hasil() {
23         return diskon;
24     }
25     @Override
26     double cekKode(String input) {
27         if (input.compareTo( anotherString:kode) == 0) {
28             diskon = 0.1;
29         } else {
30             diskon = 0;
31         }
32         return diskon;
33     }
34     @Override
35     double hitPembayaran(double bayar) {
36         diskon = bayar * diskon;
37         total = bayar - diskon;
38         return total;
39     }
40     double potSaldo() {
41         return this.saldo - total;
42     }
43     @Override
44     void tampilkanMember() {
45         System.out.println( x:"Member S0001 dengan diskon 10%");
46     }
47 }
```

3. Buatlah class baru dengan nama “G0001”, extends ke Pembayaran, masukkan sourcecode berikut :

```
11 public class G0001 extends Pembayaran {
12     public String name, alamat;
13     public int saldo;
14     public String kode;
15     double diskon, bayar, total;
16     public G0001() {
17         this.name = "Muhammad ridho";
18         this.kode = "G0001";
19         this.saldo = 5000000;
20         this.alamat = "Malang, jl.lowakwaru no 50";
21     }
22     @Override
23     double cekKode(String input) {
24         if (input.compareTo( anotherString:kode) == 0) {
25             diskon = 0.2;
26         } else {
27             diskon = 0;
28         }
29         return diskon;
30     }
31     @Override
32     double hitPembayaran(double bayar) {
33         diskon = bayar * diskon;
34         total = bayar - diskon;
35         return total;
36     }
37     double potSaldo() {
38         return this.saldo - total;
39     }
40     @Override
41     void tampilkanMember() {
42         System.out.println( x: "Member G0001 dengan diskon 20%");
43     }
44 }
```

4. Buatlah class baru dengan nama “P0001”, extends ke Pembayaran, masukkan sourcecode berikut :

```
11 class P0001 extends Pembayaran {
12     public String name, alamat;
13     public int saldo;
14     public String kode;
15     public double diskon, bayar, total;
16
17     public P0001() {
18         this.name = "Firman Frezy Pradana";
19         this.kode = "P0001";
20         this.saldo = 3000000;
21         this.alamat = "Malang, jl.singosari gang singa no 3";
22     }
23
24     @Override
25     double cekKode(String input) {
26         if (input.compareTo( anotherString:kode) == 0) {
27             diskon = 0.3;
28         } else {
29             diskon = 0;
30         }
31         return diskon;
32     }
33
34     @Override
35     double hitPembayaran(double bayar) {
36         diskon = bayar * diskon;
37         total = bayar - diskon;
38         return total;
39     }
40
41     double potSaldo() {
42         return this.saldo - total;
43     }
44
45     @Override
46     void tampilkanMember() {
47         System.out.println( x: "Member P0001 dengan diskon 30%");
48     }
49 }
```

5. Buatlah class baru dengan nama “Main\_pembayaran”, masukkan sourcecode 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         //tampilan output dari Upcasting
17         payment.tampilkanMember();
18
19         // Downcasting : Objek dari kelas induk diubah kembali menjadi objek dari kelas turunan
20         //Cek apakah P0001 merupakan turunan dari payment.
21         if (payment instanceof P0001) {
22             P0001 p0001 = (P0001) payment; //Downcasting dengan menggunakan (P0001)
23
24             // Mengakses metode yang hanya ada di kelas P0001
25             System.out.println("Jenis Mmber: " + p0001.member( member:p0001));
26             System.out.println("Total Pembelian: " + p0001.hitPembayaran( bayar:500000));
27             System.out.println("sisa saldo: " + p0001.potSaldo());
28         }
29     }
30 }
```

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

```
Output - PraktikumOOP_2118112 (run) x
run:
Member P0001 dengan diskon 30%
Jenis Mmber: Platinum
Total Pembelian: 500000.0
sisa saldo: 2500000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Analisa:

Dalam sourcecode tersebut, konsep polimorfisme diterapkan melalui upcasting dan downcasting. Pada bagian upcasting, objek dari kelas turunan “P0001” dianggap sebagai objek kelas induk Pembayaran, dan metode tampilkanMember() dipanggil. Kemudian, dengan menggunakan downcasting dan operator instanceof, dilakukan pengecekan apakah objek tersebut merupakan turunan dari P0001. Jika ya, objek diubah kembali menjadi objek P0001, memungkinkan untuk mengakses ke metode khusus seperti member(), hitPembayaran(), dan potSaldo().

### C. Project Praktikum (GUI\_Matkul)

1. Pastikan desain GUI\_Matkul(Bab 2) tampak seperti gambar di bawah ini.

**Data Matakuliah**

Kode Matakuliah	<input type="text"/>	<div></div>
Mata Kuliah	<input type="text"/>	
Dose Pengajar	<input type="text"/>	
Jumlah SKS	<input type="text"/>	
<div>Cetak Dosen Matakuliah</div>		

2. Lalu rubah desain menjadi seperti gambar di halaman berikut ini.

Tabel Properti:

No	Objek	Properti	Nilai
1	jLabel1	Text	DATA MATA KULIAH
2	jLabel2	Text	Kode Mata Kuliah
3	jLabel3	Text	Mata Kuliah
4	jLabel4	Text	Dosen Pengajar
5	jLabel5	Text	Jumlah SKS
6	jTextField1	Name	txtKdMatakuliah
		Text	
7	jTextField2	Name	txtMatakuliah
		Text	
8	jTextField3	Name	txtDosenPengajar
		Text	
9	jTextField4	Name	txtJmlSks
		Text	
10	jTextField5	Name	txtCari
		Text	
11	jButton1	Name	btnSimpan
		Text	Simpan
12	jButton2	Name	btnUbah
		Text	Ubah
13	jButton3	Name	btnHapus
		Text	Hapus
14	jButton4	Name	btnBatal
		Text	Batal
15	jButton5	Name	btnCari
		Text	Cari
16	jButton6	Name	btnNilai

		Text	Form Nilai
17	JTable	Name	table_data_matakuliah

3. Masukkan script berikut diantara nama package dan nama praktikum

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

4. Pilih Tab Source, kemudian tambahkan script berikut di bawah method konstruktor Gui\_matkul.

*Source Code:*

```
DefaultTableModel dataModel = (DefaultTableModel)
table_data_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
}
Data_kuliah dtMatkul = new Data_kuliah();
txtKdMatakuliah.setText(dtMatkul.kode_mk);
txtMatakuliah.setText(dtMatkul.nama_mk);

txtDosenPengajar.setText(dtMatkul.dosen_pengampu);
txtJmlSks.setText(Integer.toString(dtMatkul.jml_sks));
```

5. Pilih Tab Source, kemudian tambahkan script berikut di bawah method konstruktor dari GUI\_Matkul.

*Source Code:*

```
public void clear(){
txtKdMatakuliah.setText("");
txtMatakuliah.setText("");
txtDosenPengajar.setText("");
txtJmlSks.setText("");}
```

6. Ubah source code yang ada di button simpan, ubah script berikut

*Source Code:*

```

OptionPane.showMessageDialog(null, "Data anda Ditambahkan Ke
tabel");
// Mengambil model data dari tabel
DefaultTableModel dataModel = (DefaultTableModel)
table_data_matakuliah.getModel();
// Inisialisasi sebuah ArrayList bernama 'list'
List list = new ArrayList<>();
// Mengatur tabel untuk membuat kolom dari model
secara otomatis

table_data_matakuliah.setAutoCreateColumnsFromModel(true)
;
    Data_kuliah dtMatkul = new Data_kuliah();
    list.add(dtMatkul.kode_mk);
    list.add(dtMatkul.nama_mk);
    list.add(dtMatkul.dosen_pengampu);
    list.add(dtMatkul.jml_sks);
    // Menambahkan baris baru ke model tabel menggunakan
data dari ArrayList 'list'
    dataModel.addRow(list.toArray());
    // Memanggil fungsi 'clear' untuk membersihkan nilai
dari komponen
clear();

```

7. Pilih Tab Desain , Klik 2 kali pada button Hapus, Tambahkan script berikut

*Source Code:*

```

DefaultTableModel dataModel = (DefaultTableModel)
table_data_matakuliah.getModel();
    int rowCount = dataModel.getRowCount();
    while (rowCount > 0) {
        dataModel.removeRow(rowCount - 1);
        rowCount = dataModel.getRowCount(); // Update
    }

```

8. 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();
}

```

9. Running program hingga hasilnya ketika semua textfield di isi dan jika button “Batal” di tekan maka semua value yang ada di textfield GUI\_Matkul akan hilang.



**Data Matakuliah**

Kode Matakuliah

Mata Kuliah

Dose Pengajar

Jumlah SKS

Cari

Kode MK	Mata Kuliah	Nama dosen	Jml SKS
IF2112	OOP	YOSEP AGUS P...	4

Simpan

Ubah Hapus Batal

Form Nilai

## Tugas 7

1. Implementasikan Polimorfisme pada project masing- masing.
2. Ubah textArea menjadi table jika gui anda masih meggunakan textarea

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