

LAPORAN RESMI

Praktikum 11 Overloading dan Overriding

Mata Kuliah: Praktek Pemrograman Berbasis Objek



Dosen Pengampu: Andhik Ampuh Yunanto S.Kom., M.Kom.

Disusun oleh:

M. Ainur Ramadhan (3122500047)

2 D3 Teknik Informatika B

**PROGRAM STUDI D3 TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2023/2024**

A. TUGAS PENDAHULUAN

1. Jelaskan perbedaan overloading method dan overloading constructor!

Jawab:

Overloading Method dapat terjadi pada satu class atau antara super class dan subclass sedangkan overloading constructor terjadi pada satu kelas saja. Selanjutnya pada overloading method terdapat Return type sedangkan pada overloading constructor tidak terdapat return type.

2. Apakah overloading bisa terjadi antara superclass dan subclass? Jika bisa, berikan contohnya!

Jawab:

Ya, bisa. Contoh:

```
public class Bentuk {
    public void gambar(int t1) {
        // 1 parameter titik, untuk menggambar titik
        System.out.println("Menggambar titik pada titik ke-" + t1);
    }

    public void gambar(int t1, int t2) {
        // 2 parameter titik, untuk menggambar garis
        System.out.println("Menggambar garis dari titik ke-" + t1 + "
ke titik ke-" + t2);
    }

    public void gambar(int t1, int t2, int t3) {
        // 3 parameter titik, untuk menggambar segitiga
        System.out.println("Menggambar segitiga dengan titik " + t1 +
", " + t2 + ", " + t3);
    }

    public void gambar(int t1, int t2, int t3, int t4) {
        // 4 parameter titik, untuk menggambar persegi empat
        System.out.println("Menggambar persegi empat dengan titik " +
t1 + ", " + t2 + ", " + t3 + ", " + t4);
    }
}

public class WarnaiBentuk extends Bentuk {
    public void gambar(String warna, int t1, int t2) {
        // Menggambar garis dengan warna
        System.out.println("Menggambar garis dengan warna " + warna +
" dari titik ke-" + t1 + " ke titik ke-" + t2);
    }

    public void gambar(String warna, int t1, int t2, int t3) {
        // Menggambar segitiga dengan warna
    }
}
```

```

        System.out.println("Menggambar segitiga dengan warna " + warna
+ " dan titik " + t1 + ", " + t2 + ", " + t3);
    }

    public void gambar(String warna, int t1, int t2, int t3, int t4) {
        // Menggambar persegi empat dengan warna
        System.out.println("Menggambar persegi empat dengan warna " +
warna + " dan titik " + t1 + ", " + t2 + ", " + t3 + ", " + t4);
    }
}

```

3. Apakah overloading konstruktor bisa terjadi antara superclass dan subclass? Jika bisa berikan contohnya!

Tidak bisa, karena Overloading constructor hanya bisa terjadi didalam satu kelas saja.
Contoh:

```

public class Employee {
    public Employee(String name) {
        this.name = name;
    }

    public Employee(String name, String address) {
        this.name = name;
        this.address = address;
    }

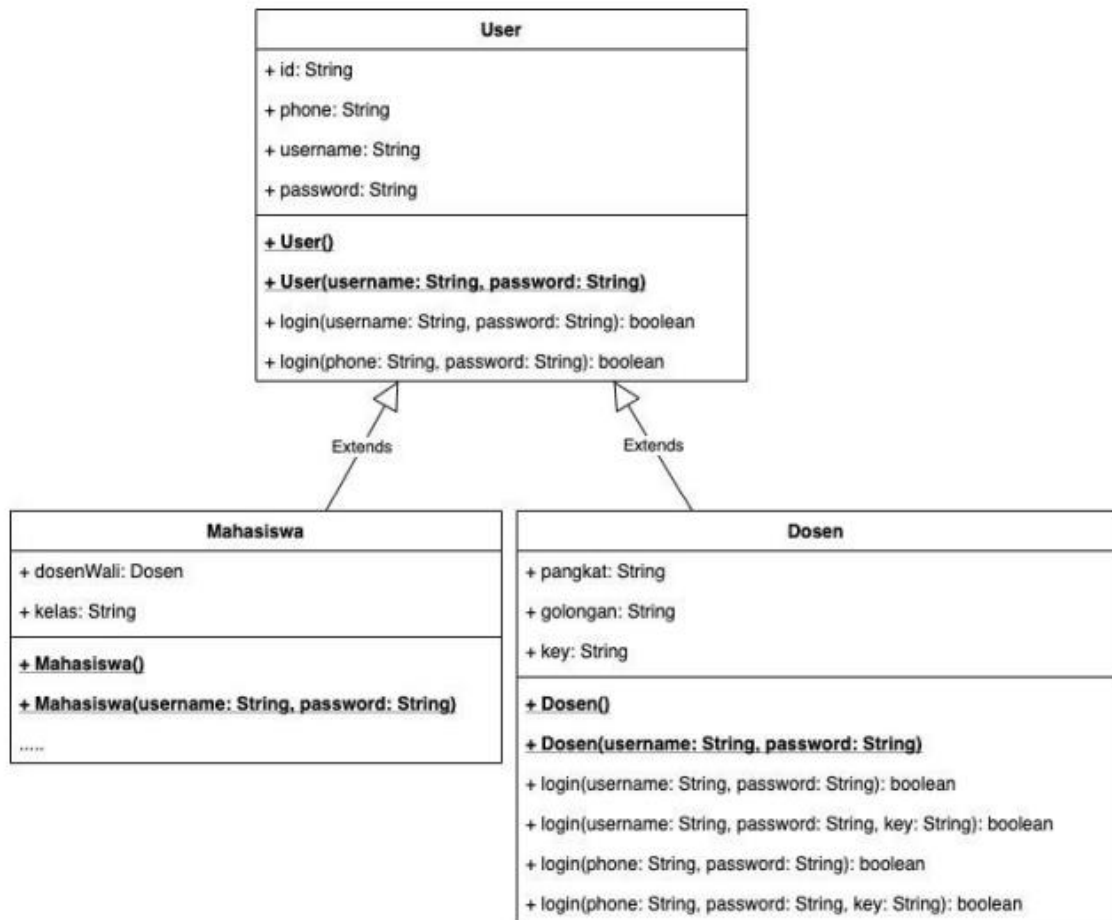
    public Employee(String name, String address, double salary) {
        this.name = name;
        this.address = address;
        this.salary = salary;
    }
}

```

B. LATIHAN

Melakukan overriding, overloading, dan overloading constructor

Perbaiki class diagram berikut dengan mengimplementasikan enkapsulasi. Kemudian implementasikan menjadi kode program! (*enkapsulasi menggunakan private)



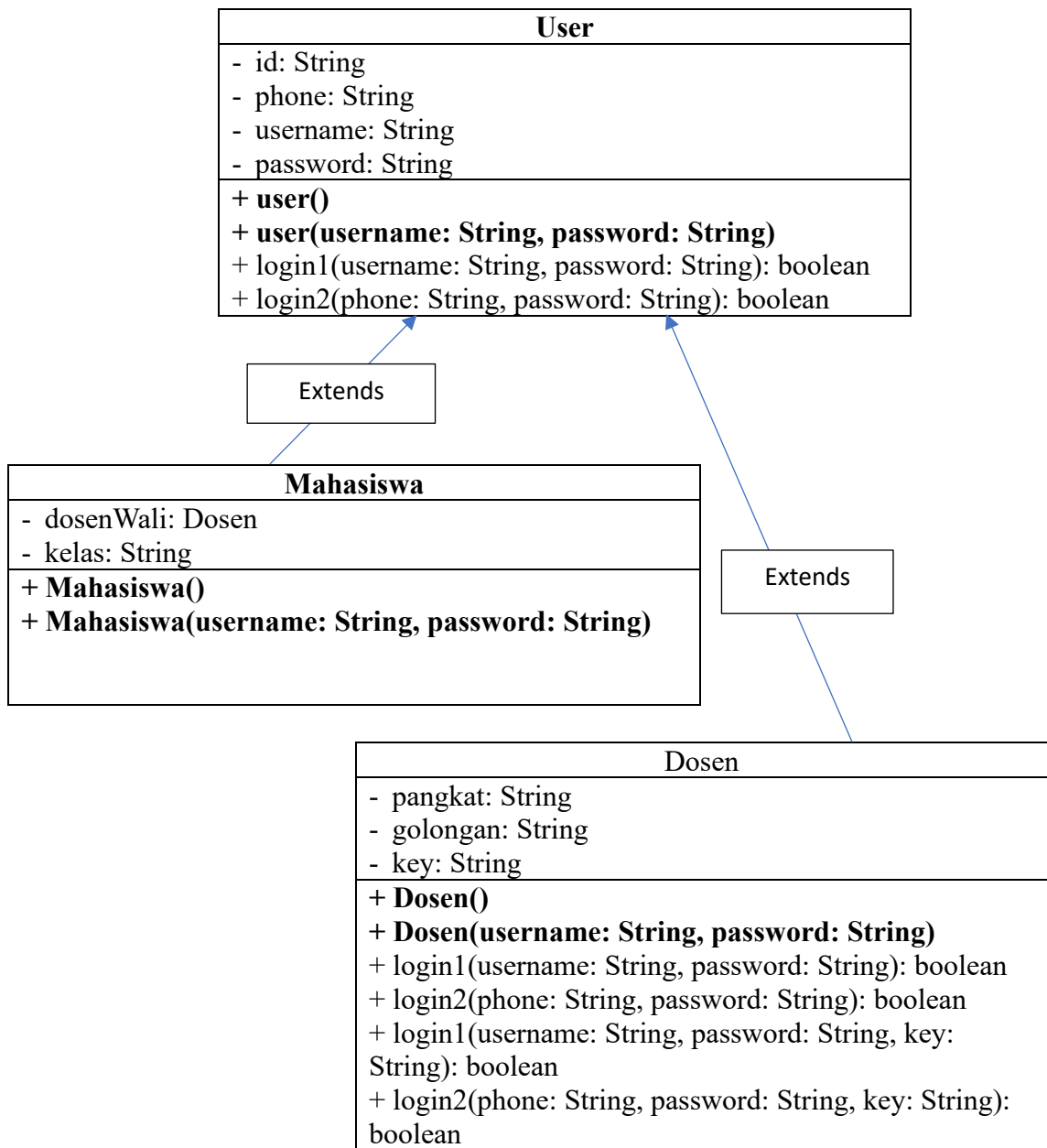
```

public class TestLogin {
    public static void main(String[] arguments) {
        Mahasiswa fadilah = new Mahasiswa();
        fadilah.setUsername("fadilah");
        fadilah.setPassword("123456");
        fadilah.setPhone("083853501388");
        boolean result = fadilah.login("083853501388",
"123456");
        System.out.println(result ? "Login berhasil" :
"Login gagal");

        Dosen fahrul = new Dosen("fahrul", "123456");
        fahrul.setKey("19990129");
        result = fahrul.login("fahrul", "123456",
"19990129");
        System.out.println(result ? "Login berhasil" :
"Login gagal");
    }
}
  
```

Jawab:

Berikut Class Diagram dengan mengimplementasikan enkapsulasi



Source Code

User.java

```

package Praktikum11.Latihan;

public class User {
    private String id;
    private String phone;
    private String username;
    private String password;

    public User() {
    }
  
```

```
public User(String username, String password) {
    this.username = username;
    this.password = password;
}

public void setId(String id) {
    this.id = id;
}

public String getId() {
    return id;
}

public void setPhone(String phone) {
    this.phone = phone;
}

public String getPhone() {
    return phone;
}

public void setUsername(String username) {
    this.username = username;
}

public String getUsername() {
    return username;
}

public void setPassword(String password) {
    this.password = password;
}

public String getPassword() {
    return password;
}

public boolean login1(String username, String password) {
    if (username == getUsername() && password == getPassword()) {
        return true;
    } else
        return false;
}

public boolean login2(String phone, String password) {
    if (phone == getPhone() && password == getPassword()) {
        return true;
    } else
```

```
        return false;
    }
}
```

Dosen.java

```
package Praktikum11.Latihan;

public class Dosen extends User {
    private String pangkat;
    private String golongan;
    private String key;

    public Dosen() {
    }

    public Dosen(String username, String password) {
        setUsername(username);
        setPassword(password);
    }

    public void setPangkat(String pangkat) {
        this.pangkat = pangkat;
    }

    public String getPangkat() {
        return pangkat;
    }

    public void setGolongan(String golongan) {
        this.golongan = golongan;
    }

    public String getGolongan() {
        return golongan;
    }

    public void setKey(String key) {
        this.key = key;
    }

    public String getKey() {
        return key;
    }

    public boolean login1(String username, String password) {
        if (username == getUsername() && password == getPassword()) {
            return true;
        } else
    }
}
```

```

        return false;
    }

    public boolean login1(String username, String password, String key) {
        if (username == getUsername() && password == getPassword()
            && key == getKey()) {
            return true;
        } else
            return false;
    }

    public boolean login2(String phone, String password) {
        if (phone == getPhone() && password == getPassword()) {
            return true;
        } else
            return false;
    }

    public boolean login2(String phone, String password, String key) {
        if (phone == getPhone() && password == getPassword() && key ==
getKey()) {
            return true;
        }
        else
            return false;
    }
}

```

Mahasiswa.java

```

package Praktikum11.Latihan;

public class Mahasiswa extends User {
    private Dosen dosenWali;
    private String kelas;

    public Mahasiswa() {
    }

    public Mahasiswa(String username, String password) {
        setUsername(username);
        setPassword(password);
    }

    public Dosen getDosenWali() {
        return dosenWali;
    }

    public void setDosenWali(Dosen dosenWali) {

```



```

        this.dosenWali = dosenWali;
    }

    public void setKelas(String kelas) {
        this.kelas = kelas;
    }

    public String getKelas() {
        return kelas;
    }
}

```

TestLogin.java

```

package Praktikum11.Latihan;

public class TestLogin {
    public static void main(String[] arg) {
        Mahasiswa fadilah = new Mahasiswa();
        fadilah.setUsername("fadilah");
        fadilah.setPassword("123456");
        fadilah.setPhone("083853501388");
        boolean result = fadilah.login2("083853501388", "123456");
        System.out.println(result ? "Login berhasil" : "Login gagal");

        Dosen fahrul = new Dosen("fahrul", "123456");
        fahrul.setKey("19990129");
        result = fahrul.login1("fahrul", "123456", "19990129");
        System.out.println(result ? "Login berhasil" : "Login gagal");
    }
}

```

Output:

```

PS C:\Users\USER\OOP\Praktikum> javac -cp . Praktikum11/Latihan/TestLogin.java
PS C:\Users\USER\OOP\Praktikum> java Praktikum11.Latihan.TestLogin
Login berhasil
Login berhasil

```

Penjelasan:

Untuk class user dua method login tersebut sebelumnya sama, namun disini saya ubah menjadi method login1 dan login2 hal ini agar tidak terjadi error pada dua method yang sama namun parameter yang berbeda. Setelah itu pemanggilan method pada class TestLogin yang awalnya login saja diubah sesuai dengan isi parameter yang digunakan.

C. TUGAS

Modifikasi program pada soal latihan dengan menghilangkan method setter & getter pada semua class tapi setiap variable tetap harus menggunakan modifier private.

Jawab:

User.java

```
package Praktikum11.Tugas;

public class User {
    private String id;
    private String phone;
    private String username;
    private String password;

    public User() {
    }

    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }

    public User(String username, String password, String phone) {
        this.username = username;
        this.password = password;
        this.phone = phone;
    }

    public boolean login1(String Username, String Password) {
        if (username == Username && password == Password) {
            return true;
        } else {
            return false;
        }
    }

    public boolean login2(String Phone, String Password) {
        if (phone == Phone && password == Password) {
            return true;
        } else {
            return false;
        }
    }
}
```

Analisa:

Menghilangkan semua method set dan get, namun untuk constructor user dilakukan overloading yang memiliki parameter phone, hal ini berguna untuk mendapatkan nilai phone yang akan diinputkan nantinya.

Dosen.java

```
package Praktikum11.Tugas;

public class Dosen extends User {
    private String pangkat;
    private String golongan;
    private String key;

    public Dosen() {
    }

    public Dosen(String username, String password, String key) {
        super(username, password);
        this.key = key;
    }

    public boolean login1(String username, String password) {
        return super.login1(username, password);
    }

    public boolean login1(String username, String password, String key) {
        return super.login1(username, password) && this.key == key;
    }

    public boolean login2(String phone, String password) {
        return super.login2(phone, password);
    }

    public boolean login2(String phone, String password, String key) {
        return super.login2(phone, password) && this.key == key;
    }
}
```

Analisa:

Menghilangkan semua method set dan get, namun untuk constructor dosen ditambah parameternya dengan variable key hal ini digunakan untuk mengeset nilai key nantinya. Setelah itu di dalam constructor terdapat pemanggilan variable dari super class yaitu username dan password. Namun untuk key hanya terdapat di dalam kelas dosen saja, sehingga tidak diperlukan pemanggilan super class.

Mahasiswa.java

```
package Praktikum11.Tugas;

public class Mahasiswa extends User {
    private Dosen dosenWali;
```

```

private String kelas;

public Mahasiswa(String username, String password, String phone) {
    super(username, password, phone);
}
}

```

Analisa:

Menghilangkan method set dan get, namun untuk constructor mahasiswa parameternya ditambah variable phone hal ini digunakan untuk mengeset nilai phone nantinya dan didalam constructor terdapat pemanggilan variable username, password, phone yang berasal dari superclass.

TestLogin.java

```

package Praktikum11.Tugas;

public class TestLogin {
    public static void main(String[] arguments) {
        Mahasiswa fadilah = new Mahasiswa("fadilah", "123456",
"083853501388");
        boolean result = fadilah.login2("083853501388", "123456");
        System.out.println(result ? "Login berhasil" : "Login gagal");

        Dosen fahrul = new Dosen("fahrul", "123456", "19990129");
        result = fahrul.login1("fahrul", "123456", "19990129");
        System.out.println(result ? "Login berhasil" : "Login gagal");
    }
}

```

Analisa:

Menghilangkan metode setKey pada saat mengeset nilai key, sehingga nilai key diset dengan menggunakan constructor kelas Dosen yang mana parameter berisikan username, password, dan key.

Output

```

PS C:\Users\USER\OOP\Praktikum> javac -cp . Praktikum11/Tugas/*.java
PS C:\Users\USER\OOP\Praktikum> java Praktikum11.Tugas.TestLogin
Login berhasil
Login berhasil

```

D. KESIMPULAN

Overloading dan overriding adalah dua konsep penting dalam pemrograman berorientasi objek yang memungkinkan fleksibilitas dalam mendefinisikan dan menggunakan metode dalam kelas-kelas yang berbeda. Overloading memungkinkan untuk memiliki metode dengan nama yang sama tetapi parameter yang berbeda, sedangkan overriding memungkinkan subclass untuk mengubah perilaku metode yang diwarisi dari superclassnya.