

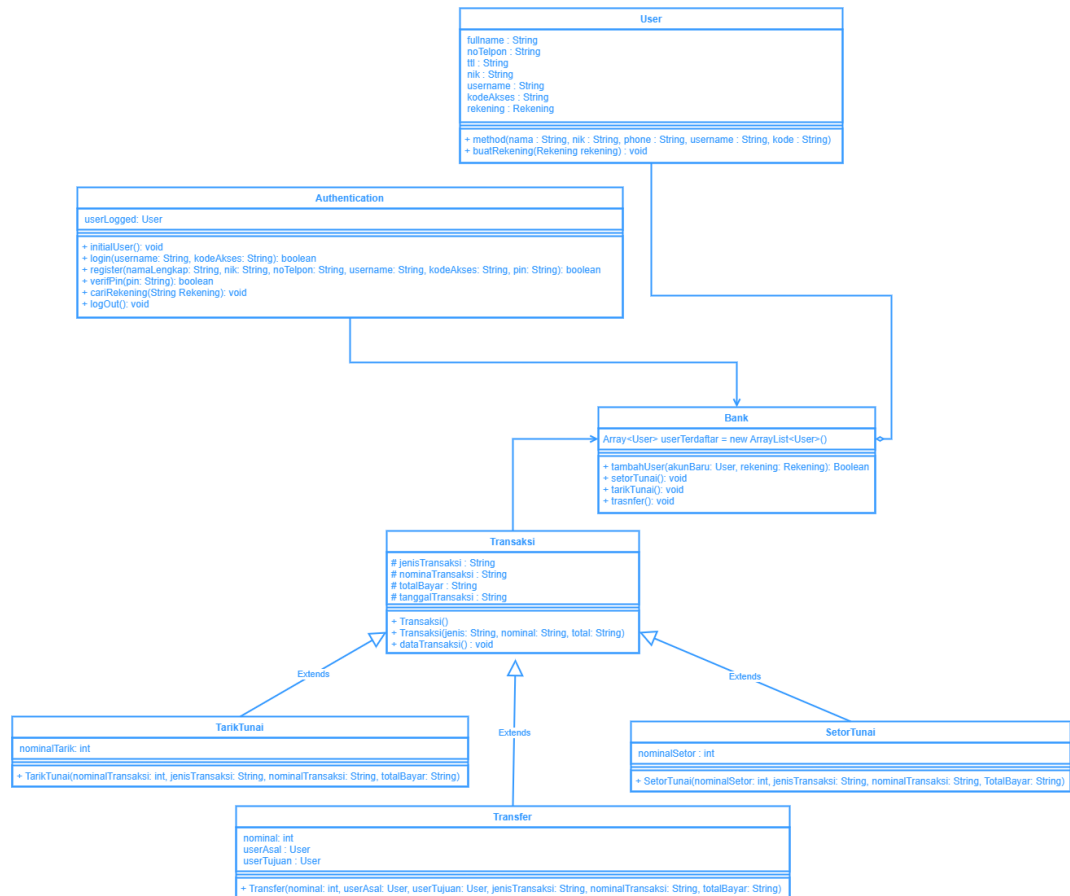


Laporan Asistensi

Tugas Pertama

Buatlah Desain Class Diagram yang Merepresentasikan Class-Class yang ada pada Studi Kasus Aplikasi Banking.

Diagram Program





Laporan Asistensi

Tugas Kedua

Buatlah Kode Program yang menerapkan Konsep Inheritance supaya pada saat mencetak data pada setiap Class akan memiliki keluaran (output) yang berbeda-beda.

Source Code

Class Member.java

```
public class Member {  
    protected String nama, alamat, password;  
    protected int noTelepon;  
  
    public String getPassword() {  
        return password;  
    }  
}
```

Class DataMahasiswa.java

```
public class DataMahasiswa extends Member {  
    protected int npm;  
    public int getNpm() {  
        return npm;  
    }  
    DataMahasiswa(String nama, String alamat, int noTelepon,  
int npm, String password) {  
        super.nama = nama;  
        super.alamat = alamat;  
        super.noTelepon = noTelepon;  
        this.npm = npm;  
        super.password = password;  
    }  
  
    public void cetak() {  
        System.out.println("Nama          : "+super.nama);  
        System.out.println("Alamat        : "+super.alamat);  
        System.out.println("No. Telepon: "+super.noTelepon);  
        System.out.println("NPM          : "+this.npm);  
        System.out.println("Pass         : "+super.password);  
        System.out.println("-----");  
    }  
}
```



Laporan Asistensi

Class Mahasiswa.java

```
import java.util.ArrayList;

public class Mahasiswa{
    private ArrayList<DataMahasiswa> mahasiswa;

    public Mahasiswa(){
        mahasiswa = new ArrayList<>();
    }
    private int search(int npm)
    {
        for (int baris = 0; baris < this.mahasiswa.size();
        baris++)
        {
            if (npm == this.mahasiswa.get(baris).getNpm())
            {
                return baris;
            }
        }
        return -1;
    }

    public void read()
    {
        System.out.println("==== DATA Mahasiswa =====");
        for (DataMahasiswa value : this.mahasiswa) {
            value.cetak();
        }
    }

    public void create(String nama, String alamat, int
    noTelepon, int npm, String password)
    {
        this.mahasiswa.add(new DataMahasiswa(nama, alamat,
        noTelepon, npm, password));
    }

    public int cekLogin(int npm, String pass)
    {

```



Laporan Asistensi

```
        for (DataMahasiswa value : this.mahasiswa) {
            if (npm == value.getNpm()) {
                if (pass.equals(value.getPassword())) {
                    return value.getNpm();
                }
            }
        }
        return -1;
    }

    public int login(int npm, String pass)
    {
        int npmMahasiswa = this.cekLogin(npm,pass);
        if (npmMahasiswa == -1)
        {
            System.out.println("NPM "+npm +" login gagal");
        }
        else
        {
            System.out.println("NPM "+npm +" login berhasil");
        }
        System.out.println("-----");
        return npmMahasiswa;
    }

    public void delete(int npm)
    {
        int index = this.search(npm);
        if (index == -1)
        {
            System.out.println("data tidak ditemukan");
        }
        else
        {
            this.mahasiswa.remove(index);
            System.out.println("data dengan NPM "+index +"
berhasil dihapus");
        }
        System.out.println("-----");
    }
}
```

Class DataTutor.java

```
public class DataTutor extends Member {
    protected int nip;
```



Laporan Asistensi

```
public int getNip() {
    return nip;
}
DataTutor(String nama, String alamat, int noTelepon, int
nip, String password) {
    super.nama = nama;
    super.alamat = alamat;
    super.noTelepon = noTelepon;
    this.nip = nip;
    super.password = password;
}

public void cetak() {
    System.out.println("Nama          : "+super.nama);
    System.out.println("Alamat          : "+super.alamat);
    System.out.println("No. Telepon: "+super.noTelepon);
    System.out.println("NIP          : "+this.nip);
    System.out.println("Pass          : "+super.password);
    System.out.println("-----");
}
}
```

Class Tutor.java

```
import java.util.ArrayList;

public class Tutor{
    private ArrayList<DataTutor> tutor;

    public Tutor(){
        tutor = new ArrayList<>();
    }
    private int search(int npm)
    {
        for (int baris = 0; baris < this.tutor.size();
baris++)
        {
            if (npm == this.tutor.get(baris).getNip())
            {
                return baris;
            }
        }
        return -1;
    }

    public void read()
```



Laporan Asistensi

```
{
    System.out.println("==== DATA Tutor Pendidik ====");
    for (DataTutor value : this.tutor) {
        value.cetak();
    }
}

public void create(String nama, String alamat, int
noTelepon, int nip, String password)
{
    this.tutor.add(new DataTutor(nama, alamat, noTelepon,
nip, password));
}

public int cekLogin(int npm, String pass)
{
    for (DataTutor value : this.tutor) {
        if (npm == value.getNip()) {
            if (pass.equals(value.getPassword())) {
                return value.getNip();
            }
        }
    }
    return -1;
}

public int login(int npm, String pass)
{
    int npmMahasiswa = this.cekLogin(npm,pass);
    if (npmMahasiswa == -1)
    {
        System.out.println("NIP "+npm +" login gagal");
    }
    else
    {
        System.out.println("NIP "+npm +" login berhasil");
    }
    System.out.println("-----");
    return npmMahasiswa;
}

public void delete(int npm)
{
}
```



Laporan Asistensi

```
int index = this.search(npm);
if (index == -1)
{
    System.out.println("data tidak ditemukan");
}
else
{
    this.tutor.remove(index);
    System.out.println("data dengan NIP "+index +
    berhasil dihapus");
}
System.out.println("-----");
}
```

Class Kinerja.java

```
public class Kinerja {
    public void kinerja() {
        System.out.println("Tugas Asistensi Pertemuan Ke-7");

        //input Data Mahasiswa
        Mahasiswa mahasiswa = new Mahasiswa();
        Tutor tutor = new Tutor();
        mahasiswa.create("Ryan", "Raya mastrip 177",
419382948, 7467, "ryanga");
        mahasiswa.create("Mutiarra", "Pondok Maritim",
936301749, 9869, "mdsq");
        tutor.create("Arif", "Griya taman Asri", 946292837,
1543, "rif");
        tutor.create("Ferdinand", "Royal Residence",
937219275, 1653, "frdnd");
        mahasiswa.read();
        tutor.read();

        //cek & method Delete
        mahasiswa.delete(5939);
        mahasiswa.delete(9869);
        tutor.delete(1653);
        tutor.delete(1837);

        //tes login
        int npm = mahasiswa.login(9869, "mdsq");
        int npm1 = mahasiswa.login(7467,"ryanga");
    }
}
```



Laporan Asistensi

```
int nip = tutor.login(1653, "frdnd");
int nip1 = tutor.login(1543, "rif");

    }
}
```

Class Main.java

```
public class Main {
    public static void main(String[] args) {
        Kinerja kerja = new Kinerja();
        kerja.kinerja();
    }
}
```

Output Program

```
"C:\Program Files\Java\jdk-16.0.1\bin\java.exe"
Tugas Asistensi Pertemuan Ke-7
==== DATA Mahasiswa ====
Nama      : Ryan
Alamat    : Raya mastrip 177
No. Telepon: 419382948
NPM       : 7467
Pass      : ryanga
-----
Nama      : Mutiara
Alamat    : Pondok Maritim
No. Telepon: 936301749
NPM       : 9869
Pass      : mdsq
-----
==== DATA Tutor Pendidik ====
Nama      : Arif
Alamat    : Griya taman Asri
No. Telepon: 946292837
NIP       : 1543
Pass      : rif
-----
Nama      : Ferdinand
Alamat    : Royal Residence
No. Telepon: 937219275
NIP       : 1653
Pass      : frdnd

data tidak ditemukan
-----
data dengan NPM 1 berhasil dihapus
-----
data dengan NIP 1 berhasil dihapus
-----
data tidak ditemukan
-----
NPM 9869 login gagal
-----
NPM 7467 login berhasil
-----
NIP 1653 login gagal
-----
NIP 1543 login berhasil
-----

Process finished with exit code 0
```