**Jelaskan Perbedaan Overloading Dan Overriding**

* Metode overloading adalah polimorfisme waktu kompilasi. Metode overriding adalah polimorfisme run time.
* Overloading Ini membantu meningkatkan keterbacaan program. Sementara itu overriding digunakan untuk memberikan implementasi spesifik pada metode yang sudah tersedia oleh class induknya atau kelas super.
* Overloading ini terjadi di dalam kelas. Sementara itu overriding dilakukan dalam dua kelas dengan hubungan warisan.
* Metode overloading mungkin atau mungkin tidak memerlukan pewarisan. Sementara metode override selalu membutuhkan pewarisan.
* Dalam hal ini, metode overloading harus memiliki nama  sama dan class yang berbeda. Sementara dalam hal ini, metode overriding harus memiliki nama dan class yang sama.
* Dalam metode overloading, tipe pengembalian bisa atau tidak bisa sama, tetapi kita harus mengubah parameter. Sementara dalam overriding, tipe pengembalian harus sama atau ko-varian.(Bagaas)

**Overloading**

* Nama method sama.
* Parameter Berbeda.
* Biasanya dalam satu class.

Contoh Sederhana Overloading :

Class RoundingValue{

private double defaultGrade = 100.3;

public RoundingValue(){}

/\*

\* First Method

\*/

public double roubdingUp () {

System.out.println("Excecuting first method");

Return (double) Math.ceil(defaultGrade);

}

/\*

\* Second Method - Its have different parameter with first method

\*/

public double roubdingUp (double numberGrade) {

System.out.println("Excecuting second method");

Return (double) Math.ceil(numberGrade);

}

/\*

\* Main Method - First Call

\*/

public static void main String[] args() {

RoundingValue roundingValue = new RoundingValue();

System.out.print("Value from first method : ");

System.out.println(""+ roundingValue.roubdingUp() );

System.out.print("Value from first method : ");

System.out.print(""+ roundingValue.roubdingUp(222.2));

}

}

**Overriding**

* Nama method sama.
* Isi / Body method berbeda.
* Biasanya berbeda class. interface dan implement.

Contoh Sederhana Overreading :

Class OverRiding {

public OverRiding(){}

/\*

\* First Method

\*/

public void firstMethod ( ) {

System.out.println("Method on class OverRiding is calling ");

}

}

Second Class

Class SubOverRiding extends OverRiding {

/\*

\* Override First Method on class OverRiding

\*/

public void firstMethod () {

System.out.println("This method override ");

System.out.println("Method on class SubOverRiding in calling ");

}

/\*

\* Main Method

\*/

Public static void main (String [ ] args) {

new OverRiding().firstMethod();

new SubOverRiding().firstMethod();;

}

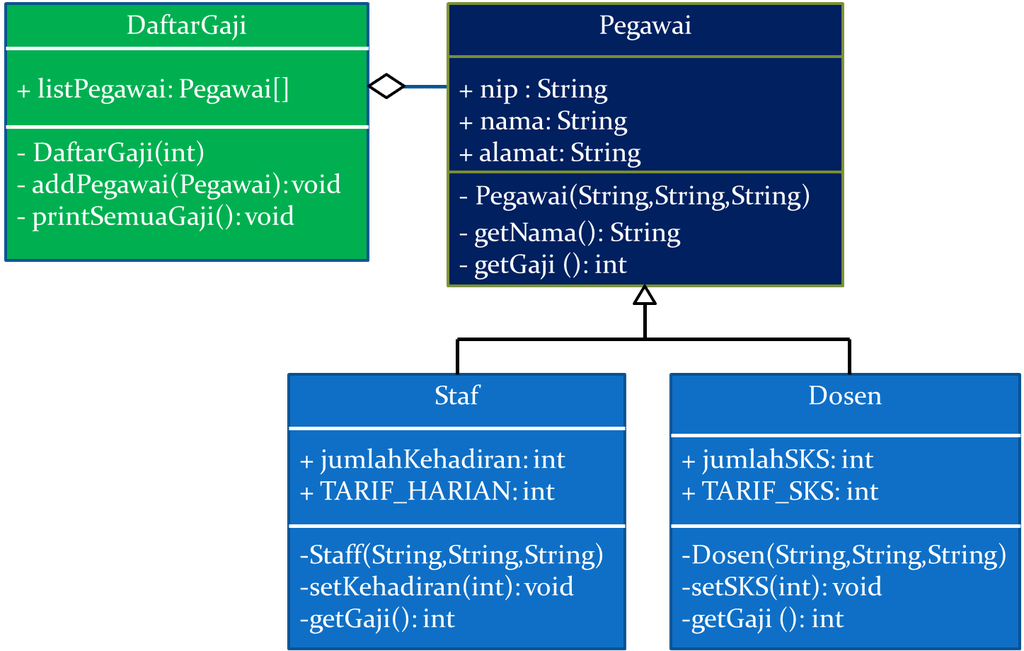
}

**STUDI KASUS: Hitung Gaji di Sebuah Universitas**

* ?Terdapat dua golongan pegawai di universitas, yaitu Dosen dan Staf biasa. Semua pegawai mendapatkan gaji awal yang sama, yaitu 1.500.000
* ?Gaji tambahan dosen dihitung dari jumlah sks yang diampu
* ?Gaji tambahan staf dihitung dari jumlah kehadiran per bulan
* ?Buatlah program untuk mencetak semua gaji pegawai

**JAWAB:**

Dari requirement di atas, kita bisa mendesain diagram class sebagai berikut:



**Diagram Class**

**IMPLEMENTASI DALAM BAHASA JAVA**

package penggajian;

/\*\*

\*

\* @author AryaDharmaadi

\*/

public class Pegawai {

    private String nip;

    private String nama;

    private String alamat;

    public Pegawai(){

    }

    public Pegawai(String nip, String nama, String alamat) {

        this.nip = nip;

        this.nama = nama;

        this.alamat = alamat;

    }

    public void setNama(String nama){

        this.nama = nama;

    }

    public String getNama(){

        return this.nama;

    }

    public int getGaji(){

        return 1500000;

    }

}

Class Staf:

package penggajian;

/\*\*

\*

\* @author AryaDharmaadi

\*/

public class Staf extends Pegawai{

    private int jumlah\_kehadiran;

    private final int TARIF\_HARIAN = 50000;

    public Staf(){

    }

    public Staf(String nip, String nama, String alamat) {

        super(nip, nama, alamat);

    }

    public void setKehadiran(int jumlah\_kehadiran){

        this.jumlah\_kehadiran = jumlah\_kehadiran;

    }

    @Override

    public int getGaji(){

        int total\_gaji = jumlah\_kehadiran\*TARIF\_HARIAN;

        return total\_gaji+super.getGaji();

    }

}

Class Dosen:

package penggajian;

/\*\*

\*

\* @author AryaDharmaadi

\*/

public class Dosen extends Pegawai{

    private int jumlahSKS;

    private final int TARIF\_SKS = 120000;

    public Dosen(){

    }

    public Dosen(String nip, String nama, String alamat) {

        super(nip, nama, alamat);

    }

    public void setSKS(int SKS){

        this.jumlahSKS = SKS;

    }

    @Override

    public int getGaji(){

        int total\_gaji = jumlahSKS\*TARIF\_SKS;

        return total\_gaji+super.getGaji();

    }

}

Class DaftarGaji:

package penggajian;

/\*\*

\*

\* @author AryaDharmaadi

\*/

public class DaftarGaji {

    private Pegawai[] listPegawai;

    private int jumlahPegawaiSekarang = 0;

    public DaftarGaji(int jumlah\_pegawai){

        listPegawai = new Pegawai[jumlah\_pegawai];

    }

    public void addPegawai(Pegawai p){

        listPegawai[jumlahPegawaiSekarang] = p;

        jumlahPegawaiSekarang++;

    }

    public void printSemuaGaji(){

        for (int i=0;i<jumlahPegawaiSekarang;i++){

            System.out.print(listPegawai[i].getNama()+" mendapatkan gaji ");

            System.out.println(listPegawai[i].getGaji());

        }

    }

}

Class ProgramUtama:

package penggajian;

/\*\*

\*

\* @author AryaDharmaadi

\*/

public class ProgramUtama {

    /\*\*

     \* @param args the command line arguments

     \*/

    public static void main(String[] args) {

        Staf s1 = new Staf();

        s1.setNama("Putu");

        s1.setKehadiran(20);

        Dosen d1 = new Dosen();

        d1.setNama("Dharmadi");

        d1.setSKS(12);

        Pegawai p1 = new Pegawai();

        p1.setNama("Kadek");

        DaftarGaji daftar\_gaji = new DaftarGaji(3);

        daftar\_gaji.addPegawai(s1);

        daftar\_gaji.addPegawai(d1);

        daftar\_gaji.addPegawai(p1);

        daftar\_gaji.printSemuaGaji();

    }

}