

# **LAPORAN 7**

## **Pemrograman Berorientasi Objek**

**“Pewarisan (Inheritance)”**

**Dosen Pengampu : Paulina Heruningsih Prima Rosa**



**DIBUAT OLEH :**

**Nama : Yohanis Calvin D.P.U.Pati**

**NIM : 245314033**

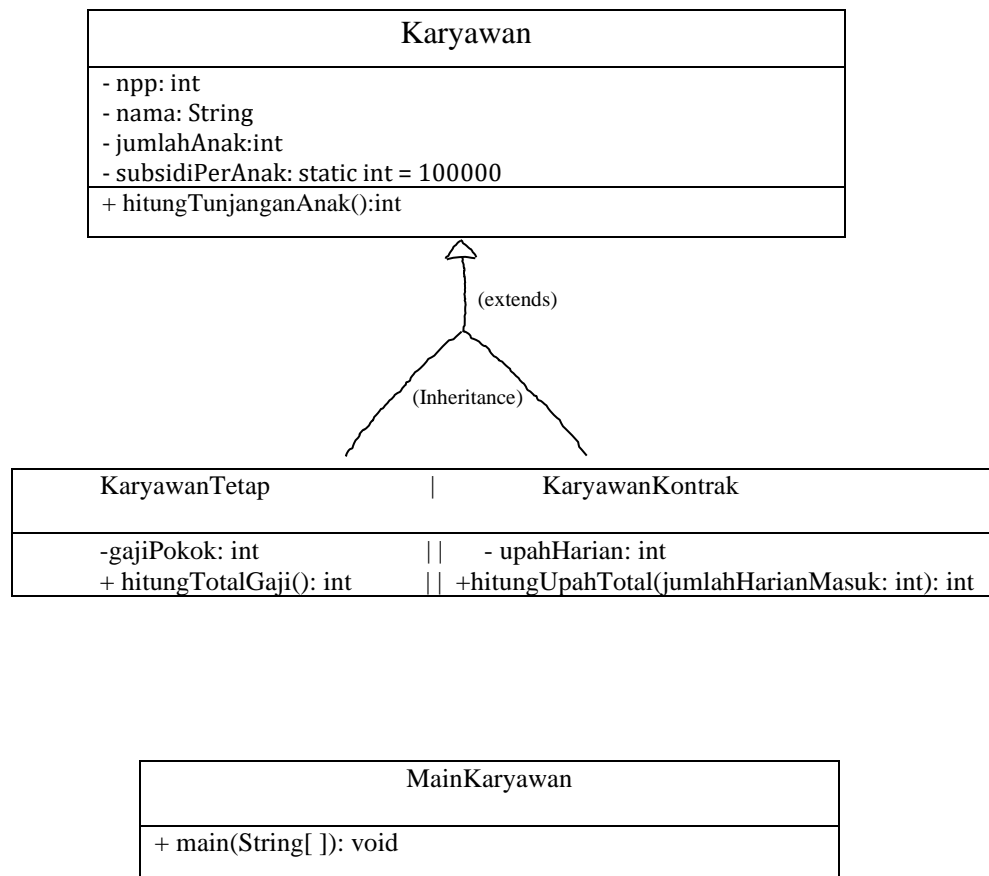
**KELAS : BP**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA  
2025**

# latihan kasus 1

## A. DIAGRAM CLASS

- Kasus yang dipecahkan: program ini memecahkan kasus system penggajian karyawan dengan dua jenis karyawan (tetap dan kontrak) yang memiliki perhitungan gaji berbeda. Dengan memakai konsep pewarisan (inheritance).
- Berikut Penggambaran diagram kelas:



## B. PENJELASAN DIAGRAM CLASS

### 1. Kelas Karyawan

- **kelas induk** (superclass) dari KaryawanTetap dan KaryawanKontrak.
- **Atribut:**
  - npp: Nomor Pokok Pegawai (tipe int)
  - nama: Nama karyawan (tipe String)
  - jumlahAnak: Jumlah anak dari karyawan (tipe int)
  - subsidiPerAnak: Konstanta subsidi per anak sebesar 100000 (static)
- **Metode:**
  - hitungTunjanganAnak(): Menghitung tunjangan berdasarkan jumlah anak  $\times$  subsidi per anak.

### 2. Kelas KaryawanTetap

- **Pewarisan:** extends Karyawan, artinya mewarisi semua atribut dan metode dari kelas Karyawan.
- **Atribut:**
  - gajiPokok: Gaji pokok tetap (tipe int)
- **metode:**
  - hitungTotalGaji(): Menghitung total gaji, biasanya gajiPokok + tunjangan anak.

### 3. Kelas KaryawanKontrak

- **Pewarisan:** extends karyawan.
- **Atribut:**
  - hitungTotalGaji(): Menghitung total gaji, biasanya gajiPokok + tunjangan anak.
- **Metode:**
  - hitungUpahTotal(jumlahHarianMasuk: int): Menghitung total upah berdasarkan jumlah hari kerja  $\times$  upah harian + tunjangan anak.

### 4. Kelas MainKaryawan

- **Kelas** ini berfungsi sebagai **kelas utama** (entry point).
- **Metode:**
  - main(String[]): Metode utama untuk menjalankan program. Di sinilah objek-objek dari kelas di atas akan dibuat dan dijalankan.
  -

## 5. Relasi Antara Kelas:

- **Inheritance (Pewarisan):**

- KaryawanTetap dan KaryawanKontrak adalah turunan dari Karyawan.
- Mereka mewarisi semua atribut dan metode dari Karyawan, lalu menambahkan atribut dan metode khusus sesuai jenisnya

## C. LISTING PROGRAM

### a. Sebelum Perbaikan

```
13 // Kelas dasar untuk semua karyawan
14 class Karyawan {
15     // Atribut private untuk nomor pegawai
16     private int npp;
17     // Atribut private untuk nama karyawan
18     private String nama;
19     // Atribut private untuk jumlah anak
20     private int jumlahAnak;
21     // Atribut static public untuk subsidi per anak (nilai tetap)
22     public static int subsidiPerAnak = 100000;
23
24     // Konstruktor untuk inialisasi objek Karyawan
25     public Karyawan(int npp, String nama, int jumlahAnak) {
26         this.npp = npp; // Mengisi atribut npp
27         this.nama = nama; // Mengisi atribut nama
28         this.jumlahAnak = jumlahAnak; // Mengisi atribut jumlahAnak
29     }
30
31     // Method untuk menghitung tunjangan anak
32     public int hitungTunjanganAnak() {
33         return jumlahAnak * subsidiPerAnak; // Jumlah anak dikali subsidi per anak
34     }
35
36     // Getter method untuk npp
37     public int getNpp() { return npp; }
38     // Getter method untuk nama
39     public String getName() { return nama; }
40     // Getter method untuk jumlahAnak
41     public int getJumlahAnak() { return jumlahAnak; }
42 }
43
44 // Kelas turunan untuk karyawan tetap
45 class KaryawanTetap extends Karyawan {
46     // Atribut private untuk gaji pokok
47     private int gajiPokok;
48
49     // Konstruktor untuk karyawan tetap
50     public KaryawanTetap(int npp, String nama, int jumlahAnak, int gajiPokok) {
51         // Memanggil konstruktor parent class
52         super(npp, nama, jumlahAnak);
53         this.gajiPokok = gajiPokok; // Mengisi atribut gajiPokok
54     }
55
56     // Method untuk menghitung total gaji
57     public int hitungTotalGaji() {
58         // Gaji pokok ditambah tunjangan anak dari parent class
59         return gajiPokok + super.hitungTunjanganAnak();
60     }
61
62     // Getter method untuk gajiPokok
63     public int getGajiPokok() { return gajiPokok; }
64 }
65
66 // Kelas turunan untuk karyawan kontrak
67 class KaryawanKontrak extends Karyawan {
68     // Atribut private untuk upah harian
69     private int upahHarian;
70
71     // Konstruktor untuk karyawan kontrak
72     public KaryawanKontrak(int npp, String nama, int jumlahAnak, int upahHarian) {
73         // Memanggil konstruktor parent class
74         super(npp, nama, jumlahAnak);
75         this.upahHarian = upahHarian; // Mengisi atribut upahHarian
76     }
77
78     // Method untuk menghitung upah total
79     public int hitungUpahTotal(int jumlahHariMasuk) {
80         // (Upah harian kali jumlah hari) plus tunjangan anak dari parent class
81         return (upahHarian * jumlahHariMasuk) + super.hitungTunjanganAnak();
82     }
83
84     // Getter method untuk upahHarian
85     public int getUpahHarian() { return upahHarian; }
86 }
87
```

```
11 // Author: Ceivin Pati
12 //
13 public class MainKaryawan {
14     public static void main(String[] args) {
15         // Membuat objek Scanner untuk input
16         Scanner scanner = new Scanner(System.in);
17
18         // Input data karyawan tetap
19         System.out.println("DATA KARYAWAN TETAP");
20         System.out.print("Masukkan NPP: ");
21         int nppTetap = scanner.nextInt(); // Membaca NPP
22         scanner.nextLine(); // Membersihkan newline
23         System.out.print("Masukkan Nama: ");
24         String namaTetap = scanner.nextLine(); // Membaca nama
25         System.out.print("Masukkan Jumlah Anak: ");
26         int anakTetap = scanner.nextInt(); // Membaca jumlah anak
27         System.out.print("Masukkan Gaji Pokok: ");
28         int gajiPokok = scanner.nextInt(); // Membaca gaji pokok
29         // Membuat objek Karyawan Tetap
30         KaryawanTetap tetap = new KaryawanTetap(nppTetap, namaTetap, anakTetap, gajiPokok);
31
32         // Input data karyawan kontrak
33         System.out.println("\nDATA KARYAWAN KONTRAK");
34         System.out.print("Masukkan NPP: ");
35         int nppKontrak = scanner.nextInt(); // Membaca NPP
36         scanner.nextLine(); // Membersihkan newline
37         System.out.print("Masukkan Nama: ");
38         String namaKontrak = scanner.nextLine(); // Membaca nama
39         System.out.print("Masukkan Jumlah Anak: ");
40         int anakKontrak = scanner.nextInt(); // Membaca jumlah anak
41         System.out.print("Masukkan Upah Harian: ");
42         int upahHarian = scanner.nextInt(); // Membaca upah harian
43         System.out.print("Masukkan Jumlah Hari Masuk: ");
44         int hariMasuk = scanner.nextInt(); // Membaca jumlah hari masuk
45         // Membuat objek Karyawan Kontrak
46         KaryawanKontrak kontrak = new KaryawanKontrak(nppKontrak, namaKontrak, anakKontrak, upahHarian);
47
48         // Menampilkan hasil perhitungan
49         System.out.println("\nHASIL PERHITUNGAN");
50         System.out.println("Karyawan Tetap:");
51         System.out.print("NPP: " + tetap.getNpp()); // Menampilkan NPP
52         System.out.print("Nama: " + tetap.getName()); // Menampilkan nama
53         System.out.print("Total Gaji: Rp " + tetap.hitungTotalGaji()); // Menampilkan total gaji
54
55         System.out.println("\nKaryawan Kontrak:");
56         System.out.print("NPP: " + kontrak.getNpp()); // Menampilkan NPP
57         System.out.print("Nama: " + kontrak.getName()); // Menampilkan nama
58         // Menampilkan total upah dengan parameter hari masuk
59         System.out.print("Total Upah: Rp " + kontrak.hitungUpahTotal(hariMasuk));
60
61         // Menutup scanner
62         scanner.close();
63     }
64 }
65
```

- Output

```

cd C:\Users\Windows\OneDrive\Documents\NetBeansProjects\Modul7PBO_FIX; "JAVA_HOME=C:\Program Files\Java\j\
Scanning for projects...

-----< com.mycompany:Modul7PBO_FIX >-----
Building Modul7PBO_FIX 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Modul7PBO_FIX ---
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\NetBeansProjects\Modul7PBO_FIX\src\s
--- compiler:3.13.0:compile (default-compile) @ Modul7PBO_FIX ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ Modul7PBO_FIX ---

DATA KARYAWAN TETAP
Masukkan NPP: 203
Masukkan Nama: cia
Masukkan Jumlah Anak: 1
Masukkan Gaji Pokok: 5000000

DATA KARYAWAN KONTRAK
Masukkan NPP: 422
Masukkan Nama: delta
Masukkan Jumlah Anak: 2
Masukkan Upah Harian: 500
Masukkan Jumlah Hari Masuk: 6

HASIL PERHITUNGAN
Karyawan Tetap:
NPP: 203
Nama: cia
Total Gaji: Rp5100000

Karyawan Kontrak:
NPP: 422
Nama: delta
Total Upah: Rp203000

BUILD SUCCESS
-----
Total time: 53.990 s
Finished at: 2025-04-25T02:49:33+07:00
-----

```

## b. Sesudah perbaikan

```

13 // Kelas dasar untuk semua karyawan
14 class Karyawan {
15     // Atribut private untuk nomor pegawai
16     private int npp;
17     // Atribut private untuk nama karyawan
18     private String nama;
19     // Atribut private untuk jumlah anak
20     private int jumlahAnak;
21     // Atribut static public untuk subsidi per anak (nilai tetap)
22     public static int subsidiPerAnak = 100000;
23
24     // Konstruktor untuk inialisasi objek Karyawan
25     public Karyawan(int npp, String nama, int jumlahAnak) {
26         this.npp = npp; // Mengisi atribut npp
27         this.nama = nama; // Mengisi atribut nama
28         this.jumlahAnak = jumlahAnak; // Mengisi atribut jumlahAnak
29     }
30
31     // Method untuk menghitung tunjangan anak
32     public int hitungTunjanganAnak() {
33         return jumlahAnak * subsidiPerAnak; // Jumlah anak dikali subsidi per anak
34     }
35
36     // Getter method untuk npp
37     public int getNpp() { return npp; }
38     // Getter method untuk nama
39     public String getNama() { return nama; }
40     // Getter method untuk jumlahAnak
41     public int getJumlahAnak() { return jumlahAnak; }
42 }
43
44 // Kelas turunan untuk karyawan tetap
45 class KaryawanTetap extends Karyawan {
46     // Atribut private untuk gaji pokok
47     private int gajiPokok;
48
49     // Konstruktor untuk karyawan tetap
50     public KaryawanTetap(int npp, String nama, int jumlahAnak, int gajiPokok) {
51         // Memanggil konstruktor parent class
52         super(npp, nama, jumlahAnak);
53         this.gajiPokok = gajiPokok; // Mengisi atribut gajiPokok
54     }
55
56     // Method untuk menghitung total gaji
57     public int hitungTotalGaji() {
58         // Gaji pokok ditambah tunjangan anak dari parent class
59         return gajiPokok + super.hitungTunjanganAnak();
60     }
61
62     // Getter method untuk gajiPokok
63     public int getGajiPokok() { return gajiPokok; }
64 }
65
66 // Kelas turunan untuk karyawan kontrak
67 class KaryawanKontrak extends Karyawan {
68     // Atribut private untuk upah harian
69     private int upahHarian;
70
71     // Konstruktor untuk karyawan kontrak
72     public KaryawanKontrak(int npp, String nama, int jumlahAnak, int upahHarian) {
73         // Memanggil konstruktor parent class
74         super(npp, nama, jumlahAnak);
75         this.upahHarian = upahHarian; // Mengisi atribut upahHarian
76     }
77
78     // Method untuk menghitung upah total
79     public int hitungUpahTotal(int jumlahHariMasuk) {
80         // (Upah harian kali jumlah hari) plus tunjangan anak dari parent class
81         return (upahHarian * jumlahHariMasuk) + super.hitungTunjanganAnak();
82     }
83
84     // Getter method untuk upahHarian
85     public int getUpahHarian() { return upahHarian; }
86 }
87

```

```

11 // Author: Calvin Pati
12
13 public class MainKaryawan {
14     public static void main(String[] args) {
15         // Membuat objek Scanner untuk input
16         Scanner scanner = new Scanner(System.in);
17
18         // Input data karyawan tetap
19         System.out.println("DATA KARYAWAN TETAP");
20         System.out.print("Masukkan NPP: ");
21         int nppTetap = scanner.nextInt(); // Membaca NPP
22         scanner.nextLine(); // Membersihkan newline
23         System.out.print("Masukkan Nama: ");
24         String namaTetap = scanner.nextLine(); // Membaca nama
25         System.out.print("Masukkan Jumlah Anak: ");
26         int anakTetap = scanner.nextInt(); // Membaca jumlah anak
27         System.out.print("Masukkan Gaji Pokok: ");
28         int gajiPokok = scanner.nextInt(); // Membaca gaji pokok
29         // Membuat objek karyawan tetap
30         KaryawanTetap tetap = new KaryawanTetap(nppTetap, namaTetap, anakTetap, gajiPokok);
31
32         // Input data karyawan kontrak
33         System.out.println("\nDATA KARYAWAN KONTRAK");
34         System.out.print("Masukkan NPP: ");
35         int nppKontrak = scanner.nextInt(); // Membaca NPP
36         scanner.nextLine(); // Membersihkan newline
37         System.out.print("Masukkan Nama: ");
38         String namaKontrak = scanner.nextLine(); // Membaca nama
39         System.out.print("Masukkan Jumlah Anak: ");
40         int anakKontrak = scanner.nextInt(); // Membaca jumlah anak
41         System.out.print("Masukkan Upah Harian: ");
42         int upahHarian = scanner.nextInt(); // Membaca upah harian
43         System.out.print("Masukkan Jumlah Hari Masuk: ");
44         int hariMasuk = scanner.nextInt(); // Membaca jumlah hari masuk
45         // Membuat objek karyawan kontrak
46         KaryawanKontrak kontrak = new KaryawanKontrak(nppKontrak, namaKontrak, anakKontrak, upahHarian);
47
48         // Menampilkan hasil perhitungan
49         System.out.println("\nHASIL PERHITUNGAN");
50         System.out.println("Karyawan Tetap:");
51         System.out.println("NPP: " + tetap.getNpp()); // Menampilkan NPP
52         System.out.println("Nama: " + tetap.getNama()); // Menampilkan nama
53         System.out.println("Total Gaji: Rp" + tetap.hitungTotalGaji()); // Menampilkan total gaji
54
55         System.out.println("\nKaryawan Kontrak:");
56         System.out.println("NPP: " + kontrak.getNpp()); // Menampilkan NPP
57         System.out.println("Nama: " + kontrak.getNama()); // Menampilkan nama
58         // Menampilkan total upah dengan parameter hari masuk
59         System.out.println("Total Upah: Rp" + kontrak.hitungUpahTotal(hariMasuk));
60
61         // Menutup scanner
62         scanner.close();
63     }
64 }

```

- Output:

```

-----< com.mycompany:Modul7PBO_FIX >-----
Building Modul7PBO_FIX 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Modul7PBO_FIX ---
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\NetBeansProjects\Moc

--- compiler:3.13.0:compile (default-compile) @ Modul7PBO_FIX ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ Modul7PBO_FIX ---
DATA KARYAWAN TETAP
Masukkan NPP: 203
Masukkan Nama: cia
Masukkan Jumlah Anak: 1
Masukkan Gaji Pokok: 5000000

DATA KARYAWAN KONTRAK
Masukkan NPP: 422
Masukkan Nama: delta
Masukkan Jumlah Anak: 2
Masukkan Upah Harian: 500
Masukkan Jumlah Hari Masuk: 6

HASIL PERHITUNGAN
Karyawan Tetap:
NPP: 203
Nama: cia
Total Gaji: Rp5100000

Karyawan Kontrak:
NPP: 422
Nama: delta
Total Upah: Rp203000
-----
BUILD SUCCESS
-----
Total time: 53.990 s
Finished at: 2025-04-25T02:49:33+07:00
-----

```

## D. ANALISA PERBANDINGA PROGRAM DIBUAT SAAT PRKTIKUM DAN DIRUMAH:

### 1. Perbedaan kedua program:

Aspek	Program paraktikum	Di rumah
Struktur File	Semua kelas ditulis dalam satu file (kecuali MainKaryawan)	Struktur tetap serupa, tetapi belum ada pemisahan class ke file terpisah
Modularitas Output	Menampilkan data di MainKaryawan langsung	Belum ada modularisasi method tampilkan data
Pemrosesan Data	Input manual di MainKaryawan	Sama: input tetap dilakukan di MainKaryawan
Tujuan Program	Menghitung total gaji karyawan tetap dan kontrak	Masih sama: menghitung total gaji dan upah
Pemanfaatan OOP	Menggunakan inheritance, constructor, encapsulation, method	Masih sama, tidak ada peningkatan OOP secara signifikan
Kemampuan Skalabilitas	Belum modular, sulit dikembangkan lebih lanjut	Masih sama, tidak ada pengembangan signifikan

## 2. Kekurangan Program Saat Praktikum

- Struktur Masih Kaku: Semua proses input dan output dilakukan langsung di main, belum menggunakan fungsi atau modularisasi.
- Tidak Ada Validasi Input: Jika user memasukkan nilai tidak wajar (misalnya negatif), program tetap berjalan.

## 3. Kelebihan Program Hasil Perbaikan

- Banyak penambahan komentar untuk setiap langkah, memudahkan pemahaman.
- Penempatan atribut dan method sesuai standar OOP, meskipun belum terlalu modular.
- Struktur class siap untuk ditambah method seperti tampilkanData(), editData(), atau penyimpanan ke file.

## 4. Kesimpulan dari Perbandingan

Program hasil perbaikan lebih baik dan terstruktur, tapi secara logika masih sama karena belum ada penambahan fitur modular seperti polymorphism atau override method.

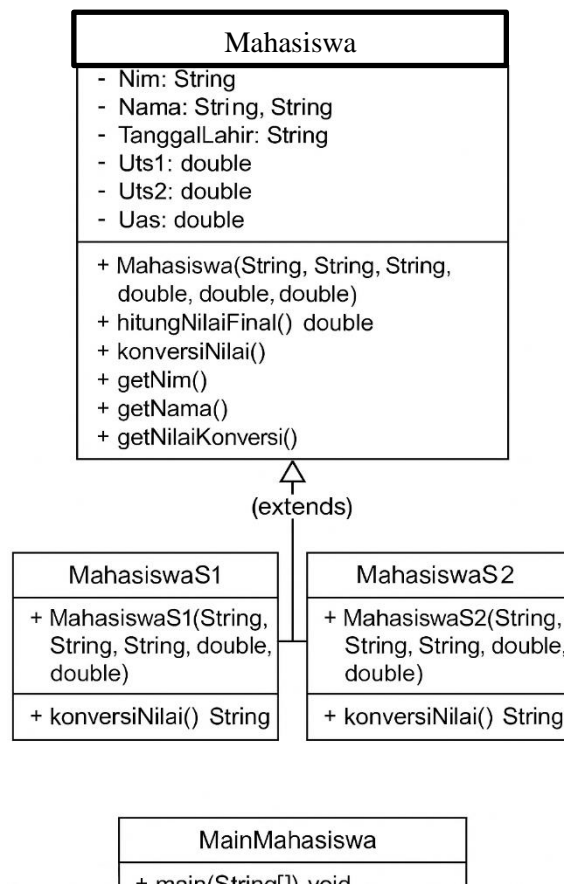
Perbaikan kode harus tidak hanya pada komentar dan kerapian, tapi juga pada struktur desain program (misal: buat class Menu, method tampilData(), dll) agar OOP-nya benar-benar maksimal.

## Latihan Kasus 2

### A. DIAGRAM CLASS

- Kasus yang dipecahkan program:

Program tersebut memecahkan kasus pengolahan data akademik mahasiswa dengan menghitung nilai akhir berdasarkan UTS dan UAS, serta mengonversi nilai tersebut ke dalam bentuk huruf mutu (A–E) yang berbeda tergantung jenjang pendidikan (S1 atau S2).





## B. PENJELASAN DIAGRAM CLASS

### 1. Kelas Mahasiswa

- **Atribut** (protected)

Nim, Nama, TanggalLahir, Uts1, Uts2, Uas, Final, NilaiKonversi

- **Method:**

- hitungNilaiFinal(): Menghitung nilai akhir berdasarkan bobot 30% UTS1, 30% UTS2, dan 40% UAS.
- konversiNilai(): Mengubah nilai angka menjadi huruf mutu berdasarkan rentang nilai.
- Getter (getNim(), getNama(), getFinal(), getNilaiKonversi())

### 2. Kelas mahasiswaS1

- **Method Override:**

- konversiNilai(): Mengubah nilai akhir ke huruf mutu khusus untuk jenjang S1.

### 3. Kelas MahasiswaS2 (turunan dari Mahasiswa)

- **Method Override:**

- konversiNilai(): Konversi nilai dengan standar yang lebih tinggi (A jika  $\geq 85$ ).

### 4. Kelas MainMahasiswa

- **Tugas:**

- Menyediakan logika utama program: input data mahasiswa, membuat objek MahasiswaS1 dan MahasiswaS2, serta menampilkan hasil.

### 5. Relasi antar kelas:

- **Inheritance** (Pewarisan):

- MahasiswaS1 dan MahasiswaS2 *mewarisi* (extends) dari Mahasiswa.

## C. LISTING PROGRAM

### a. Listing Program saat Praktikum

```
Source History View Tools Window Help
* Author: Calvin Dati
*/

public class Mahasiswa {
    // atribut
    protected String nim;
    protected String nama;
    protected String tanggalahir;
    protected double uts1;
    protected double uts2;
    protected double uas;
    protected double final;
    protected String nilaikonversi;
    // constructor
    public Mahasiswa(String Nim, String Nama, String Tanggalahir, double Uts1, double Uts2, double Uas) {
        this.nim = Nim;
        this.nama = Nama;
        this.tanggalahir = Tanggalahir;
        this.uts1 = Uts1;
        this.uts2 = Uts2;
        this.uas = Uas;
        this.final = hitungNilaiFinal();
        this.nilaikonversi = konversiNilai();
    }

    // metode untuk hitung nilai final
    public double hitungNilaiFinal() {
        return (0.3 * Uas) + (0.3 * Uts2) + (0.4 * Uts1);
    }

    public String konversiNilai() {
        if (final >= 80) {
            return "A";
        } else if (final >= 70) {
            return "B";
        } else if (final >= 56) {
            return "C";
        } else if (final >= 45) {
            return "D";
        } else {
            return "E";
        }
    }

    // metode getter
    public String getnim() {
        return nim;
    }

    public String getnama() {
        return nama;
    }

    public String gettanggalahir() {
        return tanggalahir;
    }

    public double getuts1() {
        return uts1;
    }

    public double getuts2() {
        return uts2;
    }

    public double getuas() {
        return uas;
    }

    public String getfinal() {
        return final;
    }

    public String getnilaikonversi() {
        return nilaikonversi;
    }
}
```

Activate Windows  
Go to Settings to activate Windows

```
7 //
8 *
9 * Author: Calvin Dati
10 */
11
12 public class MahasiswaS1 extends Mahasiswa {
13
14     public MahasiswaS1(String Nim, String Nama, String Tanggalahir, double Uts1, double Uts2, double Uas) {
15         super(Nim, Nama, Tanggalahir, Uts1, Uts2, Uas);
16     }
17
18     @Override
19     public String konversiNilai() {
20         if (final >= 80) {
21             return "A";
22         } else if (final >= 70) {
23             return "B";
24         } else if (final >= 56) {
25             return "C";
26         } else if (final >= 45) {
27             return "D";
28         } else {
29             return "E";
30         }
31     }
32 }
```

```
7 //
8 *
9 * Author: Calvin Dati
10 */
11 import java.lang.String;
12
13 public class MahasiswaS2 extends Mahasiswa {
14
15     public MahasiswaS2(String Nim, String Nama, String Tanggalahir, double Uts1, double Uts2, double Uas) {
16         super(Nim, Nama, Tanggalahir, Uts1, Uts2, Uas);
17     }
18
19     @Override
20     public String konversiNilai() {
21         if (final >= 80) {
22             return "A";
23         } else if (final >= 70) {
24             return "B";
25         } else if (final >= 56) {
26             return "C";
27         } else if (final >= 45) {
28             return "D";
29         } else {
30             return "E";
31         }
32     }
33 }
```

```
9 //
10 * Author: Calvin Dati
11 */
12 import java.util.Scanner;
13
14 public class MainMahasiswa {
15
16     public static void main(String[] args) {
17         Scanner input = new Scanner(System.in);
18
19         // untuk input data mahasiswa S1
20         System.out.println("Masukkan data Mahasiswa S1:");
21         System.out.print("NIM: ");
22         String nimS1 = input.nextLine();
23         System.out.print("Nama: ");
24         String namaS1 = input.nextLine();
25         System.out.print("Tanggal Lahir (DD-MM-YYYY): ");
26         String tglS1 = input.nextLine();
27         System.out.print("Nilai UTS 1: ");
28         double uts1S1 = input.nextDouble();
29         System.out.print("Nilai UTS 2: ");
30         double uts2S1 = input.nextDouble();
31         System.out.print("Nilai UAS: ");
32         double uasS1 = input.nextDouble();
33         input.nextLine(); // untuk pinda di baris baru
34
35         // untuk input data mahasiswa S2
36         System.out.println("\nMasukkan data Mahasiswa S2:");
37         System.out.print("NIM: ");
38         String nimS2 = input.nextLine();
39         System.out.print("Nama: ");
40         String namaS2 = input.nextLine();
41         System.out.print("Tanggal Lahir (DD-MM-YYYY): ");
42         String tglS2 = input.nextLine();
43         System.out.print("Nilai UTS 1: ");
44         double uts1S2 = input.nextDouble();
45         System.out.print("Nilai UTS 2: ");
46         double uts2S2 = input.nextDouble();
47         System.out.print("Nilai UAS: ");
48         double uasS2 = input.nextDouble();
49
50         // obje untuk mahasiswa S1 dan S2
51         MahasiswaS1 mhsS1 = new MahasiswaS1(nimS1, namaS1, tglS1, uts1S1, uts2S1, uasS1);
52         MahasiswaS2 mhsS2 = new MahasiswaS2(nimS2, namaS2, tglS2, uts1S2, uts2S2, uasS2);
53
54         // Untuk cetak hasil
55         System.out.println("\nMasukkan hitung Nilai:");
56         System.out.println("Mahasiswa S1:");
57         System.out.print("NIM: " + mhsS1.getNim());
58         System.out.print("Nama: " + mhsS1.getNama());
59         System.out.print("Nilai Final: " + mhsS1.getFinal());
60         System.out.print("Nilai Huruf: " + mhsS1.getNilaiKonversi());
61
62         System.out.println("\nMahasiswa S2:");
63         System.out.print("NIM: " + mhsS2.getNim());
64         System.out.print("Nama: " + mhsS2.getNama());
65         System.out.print("Nilai Final: " + mhsS2.getFinal());
66         System.out.print("Nilai Huruf: " + mhsS2.getNilaiKonversi());
67
68     }
69 }
```

- Output Program:

```

--- compiler:3.13.0:compile (default-compile) @ Modul7PBO ---
Recompiling the module because of changed source code.
Compiling 4 source files with javac [debug release 23] to target/classes

--- exec:3.1.0:exec (default-cli) @ Modul7PBO ---
Masukkan data Mahasiswa S1:
NIM: 245314033
Nama: celvin
Tanggal Lahir (DD-MM-YYYY): 12-05-2006
Nilai UTS 1: 99
Nilai UTS 2: 89
Nilai UAS: 79

Masukkan data Mahasiswa S2:
NIM: 245314020
Nama: cila
Tanggal Lahir (DD-MM-YYYY): 21-03-2006
Nilai UTS 1: 99
Nilai UTS 2: 79
Nilai UAS: 80

Basil hitung Nilai:
Mahasiswa S1:
NIM: 245314033
Nama: celvin
Nilai Final: 88.0
Nilai Huruf: A

Mahasiswa S2:
NIM: 245314020
Nama: cila
Nilai Final: 85.4
Nilai Huruf: A

BUILD SUCCESS
Total time: 56.829 s

```

## b. Listing Program Perbaiki (Dirumah)

```

8  * @author Celvin Pati
9  */
10 // Deklarasi kelas utama Mahasiswa
11 public class Mahasiswa {
12     // Deklarasi atribut-atribut kelas dengan visibility protected
13     protected String Nim; // Nomor Induk Mahasiswa
14     protected String Nama; // Nama lengkap mahasiswa
15     protected String TanggalLahir; // Tanggal lahir format DD-MM-YYYY
16     protected double Uts1; // Nilai Ujian Tengah Semester 1
17     protected double Uts2; // Nilai Ujian Tengah Semester 2
18     protected double Uas; // Nilai Ujian Akhir Semester
19     protected double Final; // Nilai akhir yang dihitung
20     protected String NilaiKonversi; // Huruf mutu (A-E)
21     // Konstrutor kelas Mahasiswa
22     public Mahasiswa(String Nim, String Nama, String TanggalLahir,
23         double Uts1, double Uts2, double Uas) {
24         // Inisialisasi atribut Nim dengan parameter Nim
25         this.Nim = Nim;
26         // Inisialisasi atribut Nama dengan parameter Nama
27         this.Nama = Nama;
28         // Inisialisasi atribut TanggalLahir dengan parameter TanggalLahir
29         this.TanggalLahir = TanggalLahir;
30         // Inisialisasi atribut Uts1 dengan parameter Uts1
31         this.Uts1 = Uts1;
32         // Inisialisasi atribut Uts2 dengan parameter Uts2
33         this.Uts2 = Uts2;
34         // Inisialisasi atribut Uas dengan parameter Uas
35         this.Uas = Uas;
36         // Hitung nilai Final menggunakan method hitungNilaiFinal()
37         this.Final = hitungNilaiFinal();
38         // Konversi nilai ke huruf menggunakan method konversiNilai()
39         this.NilaiKonversi = konversiNilai();
40     }
41
42     // Method untuk menghitung nilai akhir
43     public double hitungNilaiFinal() {
44         // Rumus perhitungan nilai akhir:
45         // 30% UTS1 + 30% UTS2 + 40% UAS
46         return (0.3 * Uts1) + (0.3 * Uts2) + (0.4 * Uas);
47     }
48
49     // Method untuk mengkonversi nilai angka ke huruf
50     public String konversiNilai() {
51         // Jika nilai Final >= 80, return "A"
52         if (Final >= 80) {
53             return "A";
54         }
55         // Jika nilai Final >= 70, return "B"
56         else if (Final >= 70) {
57             return "B";
58         }
59         // Jika nilai Final >= 56, return "C"
60         else if (Final >= 56) {
61             return "C";
62         }
63         // Jika nilai Final >= 45, return "D"
64         else if (Final >= 45) {
65             return "D";
66         }
67         // Jika nilai Final < 45, return "E"
68         else {
69             return "E";
70         }
71     }
72
73     // Getter untuk atribut Nim
74     public String getNim() {
75         return Nim;
76     }
77 }

```

```

7  *
8  * @author Celvin Pati
9  */
10 public class MahasiswaS1 extends Mahasiswa {
11     public MahasiswaS1(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) {
12         super(Nim, Nama, TanggalLahir, Uts1, Uts2, Uas);
13     }
14
15     @Override
16     public String konversiNilai() {
17         if (Final >= 80) return "A";
18         else if (Final >= 70) return "B";
19         else if (Final >= 56) return "C";
20         else if (Final >= 45) return "D";
21         else return "E";
22     }
23 }
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73

```

## • Output:

```

11 import java.util.Scanner;
12
13 public class MainMahasiswa {
14     public static void main(String[] args) {
15         // Membuat objek Scanner untuk membaca input dari keyboard
16         Scanner input = new Scanner(System.in);
17
18         // =====
19         // INPUT DATA MAHASISWA S1
20         // =====
21         System.out.println("==== INPUT DATA MAHASISWA S1 ===");
22
23         // Meminta dan membaca NIM mahasiswa S1
24         System.out.print("Masukkan NIM: ");
25         String nimS1 = input.nextLine();
26
27         // Meminta dan membaca nama mahasiswa S1
28         System.out.print("Masukkan Nama: ");
29         String namaS1 = input.nextLine();
30
31         // Meminta dan membaca tanggal lahir mahasiswa S1
32         System.out.print("Masukkan Tanggal Lahir (DD-MM-YYYY): ");
33         String tglS1 = input.nextLine();
34
35         // Meminta dan membaca nilai UTS 1 mahasiswa S1
36         System.out.print("Masukkan Nilai UTS 1: ");
37         double uts1S1 = input.nextDouble();
38
39         // Meminta dan membaca nilai UTS 2 mahasiswa S1
40         System.out.print("Masukkan Nilai UTS 2: ");
41         double uts2S1 = input.nextDouble();
42
43         // Meminta dan membaca nilai UAS mahasiswa S1
44         // Meminta dan membaca nilai UAS mahasiswa S1
45         System.out.print("Masukkan Nilai UAS: ");
46         double uasS1 = input.nextDouble();
47
48         // Memberikan newline di buffer setelah input angka
49         input.nextLine();
50
51         // Membuat objek Mahasiswa1 dengan data yang diinput
52         Mahasiswa1 mhsS1 = new Mahasiswa1(nimS1, namaS1, tglS1, uts1S1, uts2S1, uasS1);
53
54         // =====
55         // INPUT DATA MAHASISWA S2
56         // =====
57         System.out.println("\n==== INPUT DATA MAHASISWA S2 ===");
58
59         // Meminta dan membaca NIM mahasiswa S2
60         System.out.print("Masukkan NIM: ");
61         String nimS2 = input.nextLine();
62
63         // Meminta dan membaca nama mahasiswa S2
64         System.out.print("Masukkan Nama: ");
65         String namaS2 = input.nextLine();
66
67         // Meminta dan membaca tanggal lahir mahasiswa S2
68         System.out.print("Masukkan Tanggal Lahir (DD-MM-YYYY): ");
69         String tglS2 = input.nextLine();
70
71         // Meminta dan membaca nilai UTS 1 mahasiswa S2
72         System.out.print("Masukkan Nilai UTS 1: ");
73         double uts1S2 = input.nextDouble();
74
75         // Meminta dan membaca nilai UTS 2 mahasiswa S2
76         System.out.print("Masukkan Nilai UTS 2: ");
77         double uts2S2 = input.nextDouble();
78
79         // Meminta dan membaca nilai UAS mahasiswa S2
80         System.out.print("Masukkan Nilai UAS: ");
81         double uasS2 = input.nextDouble();
82
83         // Membuat objek Mahasiswa2 dengan data yang diinput
84         Mahasiswa2 mhsS2 = new Mahasiswa2(nimS2, namaS2, tglS2, uts1S2, uts2S2, uasS2);
85
86         // =====
87         // HASIL PERHITUNGAN NILAI
88         // =====
89         System.out.println("\n==== HASIL PERHITUNGAN NILAI ===");
90
91         // Menampilkan data mahasiswa S1
92         System.out.println("Mahasiswa S1:");
93         System.out.println("NIM\t\t: " + mhsS1.getNim()); // Menampilkan NIM
94         System.out.println("Nama\t\t: " + mhsS1.getNama()); // Menampilkan nama
95         System.out.println("Tanggal Lahir\t: " + mhsS1.getTglLahir()); // Menampilkan tanggal lahir
96         System.out.println("Nilai Final\t: " + mhsS1.getFinal()); // Menampilkan nilai final
97         System.out.println("Grade\t\t: " + mhsS1.getNilaiKategori()); // Menampilkan grade
98
99         // Menampilkan data mahasiswa S2
100         System.out.println("Mahasiswa S2:");
101         System.out.println("NIM\t\t: " + mhsS2.getNim()); // Menampilkan NIM
102         System.out.println("Nama\t\t: " + mhsS2.getNama()); // Menampilkan nama
103         System.out.println("Tanggal Lahir\t: " + mhsS2.getTglLahir()); // Menampilkan tanggal lahir
104         System.out.println("Nilai Final\t: " + mhsS2.getFinal()); // Menampilkan nilai final
105         System.out.println("Grade\t\t: " + mhsS2.getNilaiKategori()); // Menampilkan grade
106
107         // Menutup Scanner untuk mencegah resource leak
108         input.close();
109     }
110 }

```

Activate Wink

```

-----< com.mycompany:Modul7PBO_FIX >-----
Building Modul7PBO_FIX 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----
--- resources:3.3.1:resources (default-resources) @ Modul7PBO_FIX ---
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\NetBeansProjects\Modul7PBO
--- compiler:3.13.0:compile (default-compile) @ Modul7PBO_FIX ---
Recompiling the module because of changed source code.
Compiling 6 source files with javac [debug release 23] to target\classes
--- exec:3.1.0:exec (default-cli) @ Modul7PBO_FIX ---
==== INPUT DATA MAHASISWA S1 ====
Masukkan NIM: 245314033
Masukkan Nama: delta
Masukkan Tanggal Lahir (DD-MM-YYYY): 24-03-2006
Masukkan Nilai UTS 1: 90
Masukkan Nilai UTS 2: 80
Masukkan Nilai UAS: 70

==== INPUT DATA MAHASISWA S2 ====
Masukkan NIM: 245314022
Masukkan Nama: cia
Masukkan Tanggal Lahir (DD-MM-YYYY): 21-04-2006
Masukkan Nilai UTS 1: 90
Masukkan Nilai UTS 2: 80
Masukkan Nilai UAS: 70

==== HASIL PERHITUNGAN NILAI ====
MAHASISWA S1:
NIM      : 245314033
Nama     : delta
Tanggal Lahir : 24-03-2006
Nilai Final : 79.0
Grade    : B

MAHASISWA S2:
NIM      : 245314022
Nama     : cia
Tanggal Lahir : 21-04-2006
Nilai Final : 79.0
Grade    : B

BUILD SUCCESS

Total time: 48.361 s
Finished at: 2025-04-25T01:44:08+07:00

```

#### D. ANALISA PERBANDINGA PROGRAM DIBUAT SAAT PRKTIKUM DAN DIRUMAH:

##### 1. Perbedaan kedua program:

Aspek	Program sebelum perbaikan	Program sesudah perbaikan
Komentar& Dokumentasi	Minim komentar, kurang penjelasan tiap bagian	Penuh komentar, setiap baris penting dijelaskan
Struktur Kode	Langsung ke logika utama	Lebih terstruktur, dibagi dengan penjelasan tiap bagian
Konsistensi Output	Output langsung cetak nilai	Output rapi dengan label dan format yang konsisten
Penutupan Scanner	Tidak ditutup	Scanner ditutup untuk mencegah resource leak
Method tampilanData()	Belum digunakan	Sudah disediakan meski belum diimplementasikan
Komentar Bahasa	Bahasa campuran, cenderung informa	Bahasa baku, deskriptif, dan sesuai konvensi OOP

##### 2. Kekurangan program saat praktikum

- Kurang komentar, membuat kode sulit dipahami orang lain.
- Format output belum rapi.
- Tidak menutup Scanner, yang bisa menyebabkan *resource leak*.
- Nama method dan atribut belum dijelaskan fungsinya.
- Tidak mengelompokkan bagian input dan output secara jelas.

3. Kelebihan program setelah perbaikan

- Penambahan komentar sangat membantu pemahaman.
- Output lebih informatif dan terformat dengan rapi.
- Penggunaan struktur program lebih tertata dan mudah dibaca.
- Penutupan Scanner menunjukkan *best practice* dalam pemrograman Java.
- Terdapat potensi pengembangan lewat tampilanData() di masa depan.

4. Kesimpulan:

Kesimpulannya Dengan memberikan komentar, penamaan yang jelas, dan pemisahan logika, program menjadi lebih rapi dan terstruktur. Praktikum sering fokus pada "fungsi", tetapi di rumah bisa disempurnakan menjadi lebih baik.

**A. Referensi yang diacu saat membuat laporan, termasuk gen AI jika Anda menggunakannya.**

- MODUL VII PRAKTIKUM PBO Pewarisan (Inheritance)
- POE AI

**B. Refleksi Proses Belajar:**

- Apakah Anda semakin paham ATAU semakin tidak paham? Yaa....(hanya mengalami sedikit kebingungan).
- Mengapa? Karena strur atau alur proses pembelajaran mendukung dan membantu saya untuk terus meningkatkan pemahaman saya.
- Apa rencana Anda selanjutnya terkait topik yang sudah dibahas maupunyaiangan dibahas minggu berikutnya dalam matakuliah PBO? Rencana saya yaitu saya mau belajar mandiri atu membuat satu program berkaitan dengan (Pewarisan).