

LAPORAN 7

Pemrograman Berorientasi Objek

“Pewarisan (Inheritance)”

Dosen Pengampu : Paulina Heruningsih Prima Rosa



DIBUAT OLEH :

Nama : Yohanis Celvin D.P.U.Pati
NIM : 245314033

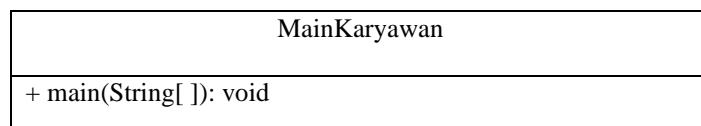
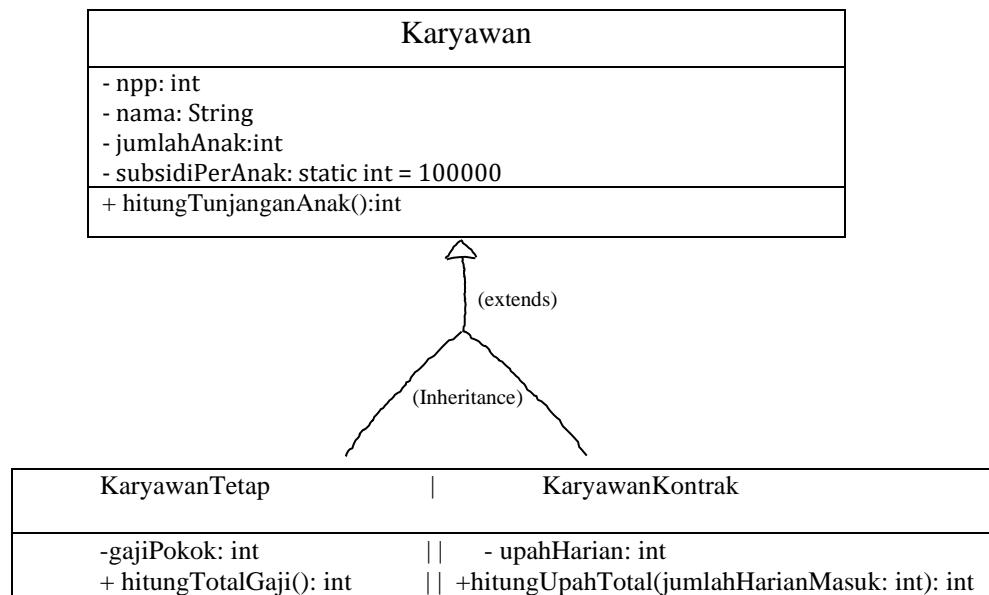
KELAS : BP

**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 × 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 × 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 inisialisasi 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
47     // Atribut private untuk gaji pokok
48     private int gajiPokok;
49
50     // Konstruktor untuk karyawan tetap
51     public KaryawanTetap(int npp, String nama, int jumlahAnak, int gajiPokok) {
52         // Memanggil konstruktor parent class
53         super(npp, nama, jumlahAnak);
54         this.gajiPokok = gajiPokok; // Mengisi atribut gajiPokok
55     }
56
57     // Method untuk menghitung total gaji
58     public int hitungTotalGaji() {
59         // Gaji pokok ditambah tunjangan anak dari parent class
60         return gajiPokok + super.hitungTunjanganAnak();
61     }
62
63     // Getter method untuk gajiPokok
64     public int getGajiPokok() { return gajiPokok; }
65 }
66
67 // Kelas turunan untuk karyawan kontrak
68 class KaryawanKontrak extends Karyawan {
69     // Atribut private untuk upah harian
70     private int upahHarian;
71
72     // Konstruktor untuk karyawan kontrak
73     public KaryawanKontrak(int npp, String nama, int jumlahAnak, int upahHarian) {
74         // Memanggil konstruktor parent class
75         super(npp, nama, jumlahAnak);
76         this.upahHarian = upahHarian; // Mengisi atribut upahHarian
77     }
78
79     // Method untuk menghitung upah total
80     public int hitungUpahTotal(int jumlahHariMasuk) {
81         // (Upah harian kali jumlah hari) plus tunjangan anak dari parent class
82         return (upahHarian * jumlahHariMasuk) + super.hitungTunjanganAnak();
83     }
84
85     // Getter method untuk upahHarian
86     public int getUpahHarian() { return upahHarian; }
87 }
```

```

11 * @author Colvin 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
30         // Membuat objek karyawan tetap
31         KaryawanTetap tetrap = new KaryawanTetap(nppTetap, namaTetap, anakTetap, gajiPokok);
32
33         // Input data karyawan kontrak
34         System.out.println("DATA KARYAWAN KONTRAK");
35         System.out.print("Masukkan NPP: ");
36         int nppKontrak = scanner.nextInt(); // Membaca NPP
37         scanner.nextLine(); // Membersihkan newline
38         System.out.print("Masukkan Nama: ");
39         String namaKontrak = scanner.nextLine(); // Membaca nama
40         System.out.print("Masukkan Jumlah Anak: ");
41         int anakKontrak = scanner.nextInt(); // Membaca jumlah anak
42         System.out.print("Masukkan Upah Harian: ");
43         int upahHarian = scanner.nextInt(); // Membaca upah harian
44         System.out.print("Masukkan Jumlah Hari Masuk: ");
45
46         int hariKerja = scanner.nextInt(); // Membaca jumlah hari masuk
47         // Membuat objek karyawan kontrak
48         KaryawanKontrak kontrak = new KaryawanKontrak(nppKontrak, namaKontrak, anakKontrak, upahHarian);
49
50         // Menampilkan hasil perhitungan
51         System.out.println("TOTAL PERHITUNGAN");
52         System.out.println("Karyawan Tetap");
53         System.out.println("NPP: " + tetrap.getNpp()); // Menampilkan NPP
54         System.out.println("Nama: " + tetrap.getNama()); // Menampilkan nama
55         System.out.println("Total Gaji Rp: " + tetrap.hitungTotalGaji()); // Menampilkan total gaji
56         System.out.println("Karyawan Kontrak");
57         System.out.println("NPP: " + kontrak.getNpp()); // Menampilkan NPP
58         System.out.println("Nama: " + kontrak.getNama()); // Menampilkan nama
59         System.out.println("Total Upah Rp: " + kontrak.hitungUpahTotal(hariKerja));
60
61         // Kelutup scanner
62         scanner.close();
63     }
64 }
65 
```

● Output

```

cd C:\Users\Windows\OneDrive\Documents\NetBeansProjects\Modul7PBO_FIX; "JAVA_HOME=C:\Program Files\Java\\">
Scanning for projects...
-----< com.mycompany:Modul7PBO_FIX >-----
Building Modul7PBO_FIX 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----
--- resources:3.3:resources (default-resources) @ Modul7PBO_FIX ---
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\NetBeansProjects\Modul7PBO_FIX\src\main\resources
--- 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 NPF: 203
Masukkan Nama: cia
Masukkan Jumlah Anak: 1
Masukkan Gaji Pokok: 5000000

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

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

Karyawan Kontrak:
NPF: 422
Nama: delta
Total Upah: Rp2030000

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

b. Sesudah perbaikan

<pre> 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 inisialisasi 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 47 // Atribut private untuk gaji pokok 48 private int gajiPokok; 49 50 // Konstruktor untuk karyawan tetap 51 public KaryawanTetap(int npp, String nama, int jumlahAnak, int gajiPokok) { 52 // Memanggil konstruktur parent class 53 super(npp, nama, jumlahAnak); 54 this.gajiPokok = gajiPokok; // Mengisi atribut gajiPokok 55 } 56 57 // Method untuk menghitung total gaji 58 public int hitungTotalGaji() { 59 // Gaji pokok ditambah tunjangan anak dari parent class 60 return gajiPokok + super.hitungTunjanganAnak(); 61 } 62 63 // Getter method untuk gajiPokok 64 public int getGajiPokok() { return gajiPokok; } 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 konstruktur 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 }</pre>	<pre> 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 NPF: "); 21 int nppTetap = scanner.nextInt(); // Membaca NPF 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 30 // Membuat Objek Karyawan tetap 31 KaryawanTetap tetap = new KaryawanTetap(nppTetap, namaTetap, anakTetap, gajiPokok); 32 33 // Input data karyawan kontrak 34 System.out.println("\nDATA KARYAWAN KONTRAK"); 35 System.out.print("Masukkan NPF: "); 36 int nppKontrak = scanner.nextInt(); // Membaca NPF 37 scanner.nextLine(); // Membersihkan newline 38 System.out.print("Masukkan Nama: "); 39 String namaKontrak = scanner.nextLine(); // Membaca nama 40 System.out.print("Masukkan Jumlah Anak: "); 41 int anakKontrak = scanner.nextInt(); // Membaca jumlah anak 42 System.out.print("Masukkan Upah Harian: "); 43 int upahHarian = scanner.nextInt(); // Membaca upah harian 44 System.out.print("Masukkan Jumlah Hari Masuk: "); 45 int jumlahHariMasuk = scanner.nextInt(); // Membaca jumlah hari masuk 46 // Komposit objek karyawan kontrak 47 KaryawanKontrak kontrak = new KaryawanKontrak(nppKontrak, namaKontrak, anakKontrak, upahHarian); 48 49 // Menampilkan hasil perhitungan 50 System.out.println("Masukkan NPF: " + kontrak.getNpp()); // Menampilkan NPF 51 System.out.println("NPF: " + tetap.getNpp()); // Menampilkan NPF 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("NPF: " + kontrak.getNpp()); // Menampilkan NPF 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(jumlahHariMasuk)); 60 61 // Menutup scanner 62 scanner.close(); 63 } 64 }</pre>
--	--

- Output:

```

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
Nothing to compile - all classes are up to date.

--- compiler:3.13.0:compile (default-compile) @ Modul7PBO_FIX ---
DATA KARYAWAN TETAP
Masukkan NPF: 203
Masukkan Nama: cia
Masukkan Jumlah Anak: 1
Masukkan Gaji Pokok: 5000000

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

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

Karyawan Kontrak:
NPF: 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.

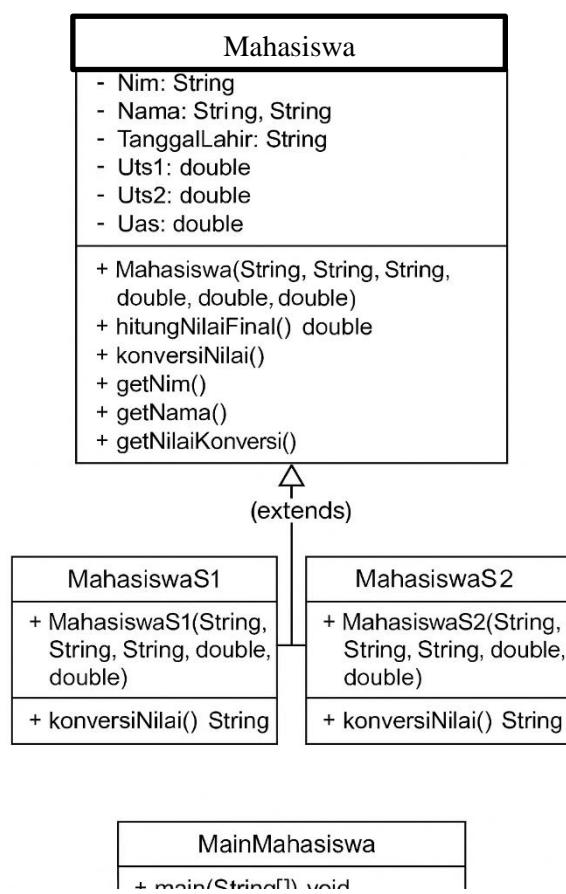
Perbaikan kode harus tidak hanya pada komentar dan kerapian, tapi juga pada struktur desain program (misal: buat class Menu, method tampilData(), dll) agar OOP-nya benar-benar maksimal.

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 ≥ 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 Help File Edit View Insert Tools Window Help
  + Author Calvin Pati
  +
  public class Mahasiswa {
    // atribut
    protected String nim;
    protected String Nama;
    protected String TanggalLahir;
    protected double Uts1;
    protected double Uts2;
    protected double Uas;
    protected double nilai;
    protected String nilaiKonversi;
    // konstruktor
    public Mahasiswa(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) {
        this.Nim = Nim;
        this>Nama = Nama;
        this.TanggalLahir = TanggalLahir;
        this.Uts1 = Uts1;
        this.Uts2 = Uts2;
        this.Uas = Uas;
        this.nilai = hitungNilaiFinal();
        this.nilaiKonversi = konversiNilai();
    }
    // metode untuk hitung nilai final
    public double hitungNilaiFinal() {
        return (0.3 * Uts1) + (0.3 * Uts2) + (0.4 * Uas);
    }
    public String konversiNilai() {
        if (nilai >= 80) {
            return "A";
        } else if (nilai >= 70) {
            return "A-";
        } else if (nilai >= 60) {
            return "B";
        } else if (nilai >= 50) {
            return "C";
        } else if (nilai >= 35) {
            return "D";
        } else {
            return "E";
        }
    }
    // metode getter
    public String getNim() {
        return Nim;
    }
    public String getNama() {
        return Nama;
    }
    public String getTglLahir() {
        return TanggalLahir;
    }
    public double getUts1() {
        return Uts1;
    }
    public double getUts2() {
        return Uts2;
    }
    public double getUas() {
        return Uas;
    }
    public String getNilaiKonversi() {
        return nilaiKonversi;
    }
}
  +
  /**
   * 
   * @author Calvin Pati
   */
  public class MahasiswaS1 extends Mahasiswa {
    public MahasiswaS1(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) {
        super(Nim, Nama, TanggalLahir, Uts1, Uts2, Uas);
    }
    @Override
    public String konversiNilai() {
        if (Nilai >= 80) {
            return "A";
        } else if (Nilai >= 70) {
            return "A-";
        } else if (Nilai >= 60) {
            return "B";
        } else if (Nilai >= 45) {
            return "C";
        } else {
            return "E";
        }
    }
}
  +
  /**
   * 
   * @author Calvin Pati
   */
  import java.lang.String;
  public class MahasiswaS2 extends Mahasiswa {
    public MahasiswaS2(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) {
        super(Nim, Nama, TanggalLahir, Uts1, Uts2, Uas);
    }
    @Override
    public String konversiNilai() {
        if (Nilai >= 85) {
            return "A";
        } else if (Nilai >= 70) {
            return "A-";
        } else if (Nilai >= 60) {
            return "B";
        } else if (Nilai >= 45) {
            return "C";
        } else {
            return "E";
        }
    }
}
  +
  /**
   * 
   * @author Calvin Pati
   */
  import java.util.Scanner;
  public class MainMahasiswa {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // untuk input data mahasiswa S1
        System.out.println("Masukkan data Mahasiswa S1:");
        System.out.print("NIM: ");
        String nimS1 = input.nextLine();
        System.out.print("Nama: ");
        String namaS1 = input.nextLine();
        System.out.print("Tanggal Lahir (DD-MM-YYYY): ");
        String tglS1 = input.nextLine();
        System.out.print("Nilai UTS 1: ");
        double uts1S1 = input.nextDouble();
        System.out.print("Nilai UTS 2: ");
        double uts2S1 = input.nextDouble();
        System.out.print("Nilai UAS: ");
        double uasS1 = input.nextDouble();
        input.nextLine(); // untuk pindah di baris baru

        // untuk input data mahasiswa S2
        System.out.println("\nMasukkan data Mahasiswa S2:");
        System.out.print("NIM: ");
        String nimS2 = input.nextLine();
        System.out.print("Nama: ");
        String namaS2 = input.nextLine();
        System.out.print("Tanggal Lahir (DD-MM-YYYY): ");
        String tglS2 = input.nextLine();
        System.out.print("Nilai UTS 1: ");
        double uts1S2 = input.nextDouble();
        System.out.print("Nilai UTS 2: ");
        double uts2S2 = input.nextDouble();
        System.out.print("Nilai UAS: ");
        double uasS2 = input.nextDouble();

        // objek untuk mahasiswa S1 dan S2
        MahasiswaS1 mhsS1 = new MahasiswaS1(nimS1, namaS1, tglS1, uts1S1, uts2S1, uasS1);
        MahasiswaS2 mhsS2 = new MahasiswaS2(nimS2, namaS2, tglS2, uts1S2, uts2S2, uasS2);

        // Untuk cetak hasil
        System.out.println("Mahasiswa S1 hitung Nilai:");
        System.out.println("Mahasiswa S1:");
        System.out.print("NIM: " + mhsS1.getNim());
        System.out.print("Nama: " + mhsS1.getNama());
        System.out.print("Nilai Final: " + mhsS1.getNilai());
        System.out.print("Nilai Huruf: " + mhsS1.getNilaiKonversi());

        System.out.println("\nMahasiswa S2:");
        System.out.print("NIM: " + mhsS2.getNim());
        System.out.print("Nama: " + mhsS2.getNama());
        System.out.print("Nilai Final: " + mhsS2.getNilai());
        System.out.print("Nilai Huruf: " + mhsS2.getNilaiKonversi());
    }
}

```

- 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: cia
Tanggal Lahir (DD-MM-YYYY): 21-03-2006
Nilai UTS 1: 99
Nilai UTS 2: 79
Nilai UAS: 80

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

Mahasiswa S2:
NIM: 245314020
Nama: cia
Nilai Final: 85.4
Nilai Huruf: A
-----
BUILD SUCCESS
-----
Total time: 56.029 s

```

b. Listing Program Perbaika (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
22     // Konstruktor kelas Mahasiswa
23     public Mahasiswa(String Nim, String Nama, String TanggalLahir,
24             double Uts1, double Uts2, double Uas) {
25         // Inisialisasi atribut Nim dengan parameter Nim
26         this.Nim = Nim;
27         // Inisialisasi atribut Nama dengan parameter Nama
28         this.Nama = Nama;
29         // Inisialisasi atribut TanggalLahir dengan parameter TanggalLahir
30         this.TanggalLahir = TanggalLahir;
31         // Inisialisasi atribut Uts1 dengan parameter Uts1
32         this.Uts1 = Uts1;
33         // Inisialisasi atribut Uts2 dengan parameter Uts2
34         this.Uts2 = Uts2;
35         // Inisialisasi atribut Uas dengan parameter Uas
36         this.Uas = Uas;
37         // Hitung nilai Final menggunakan method hitungNilaiFinal()
38         this.Final = hitungNilaiFinal();
39         // Konversi nilai ke huruf menggunakan method konversiNilai()
40         this.NilaiKonversi = konversiNilai();
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     // Getter untuk atribut Nim
73     public String getNim() {
74         return Nim;
75     }
76 }

```

```

7 /**
8 * @author Celvin Pati
9 */
10 public class Mahasiswa1 extends Mahasiswa {
11     public Mahasiswa1(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 * @author Celvin Pati
28 */
29 // Deklarasi kelas Mahasiswa2 yang merupakan turunan dari kelas Mahasiswa
30 public class Mahasiswa2 extends Mahasiswa {
31
32     // Konstruktor untuk kelas Mahasiswa2
33     public Mahasiswa2(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) {
34         // Memanggil konstruktor dari kelas induk (Mahasiswa) dengan parameter yang diberikan
35         super(Nim, Nama, TanggalLahir, Uts1, Uts2, Uas);
36     }
37
38     // Override method konversiNilai() dari kelas induk
39     @Override
40     public String konversiNilai() {
41         // Jika nilai Final >= 80, return "A"
42         if (Final >= 80) return "A";
43         // Jika nilai Final >= 70 (dan <80), return "B"
44         else if (Final >= 70) return "B";
45         // Jika nilai Final >= 56 (dan <70), return "C"
46         else if (Final >= 56) return "C";
47         // Jika nilai Final >= 45 (dan <56), return "D"
48         else if (Final >= 45) return "D";
49         // Jika nilai Final < 45, return "E"
50         else return "E";
51     }
52 }
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
22
23
24
25
26
27
28
29
2
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
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
4
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```

• Output:

The screenshot shows the NetBeans IDE interface. On the left is the Java code editor with the following content:

```

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         // Memerlukan newline di buffer setelah input angka
45         System.out.print("Masukkan Nilai UAS: ");
46         double uasS1 = input.nextDouble();
47
48         // Membersihkan buffer setelah input angka
49         input.nextLine();
50
51         // Membuat objek MahasiswaS1 dengan data yang diinput
52         MahasiswaS1 mhsS1 = new MahasiswaS1(nimS1, namaS1, tglS1, uts1S1, uts2S1, uasS1);
53
54         // =====
55         // INPUT DATA MAHASISWA S2
56         // =====
57         System.out.println("=>==> 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         // Simbul objek MahasiswaS2 dengan data yang diinput
84         MahasiswaS2 mhsS2 = new MahasiswaS2(nimS2, namaS2, tglS2, uts1S2, uts2S2, uasS2);
85         System.out.println("==>==> MULAI PENGIRIMAN NILAI ==>==> ");
86
87         // Menampilkan data mahasiswa S1
88         System.out.println("MAHASISWA S1");
89         System.out.println("NIM : " + mhsS1.getNim()); // Menampilkan NIM
90         System.out.println("Nama : " + mhsS1.getNama()); // Menampilkan nama
91         System.out.println("Tanggal Lahir: " + mhsS1.getTanggalLahir()); // Menampilkan tanggal lahir
92         System.out.println("Nilai Final : " + mhsS1.getFinal()); // Menampilkan nilai final
93         System.out.println("Grade : " + mhsS1.getNilaiKonservasi()); // Menampilkan grade
94
95         // Menampilkan data mahasiswa S2
96         System.out.println("MAHASISWA S2");
97         System.out.println("NIM : " + mhsS2.getNim()); // Menampilkan NIM
98         System.out.println("Nama : " + mhsS2.getNama()); // Menampilkan nama
99         System.out.println("Tanggal Lahir: " + mhsS2.getTanggalLahir()); // Menampilkan tanggal lahir
100        System.out.println("Nilai Final : " + mhsS2.getFinal()); // Menampilkan nilai final
101        System.out.println("Grade : " + mhsS2.getNilaiKonservasi()); // Menampilkan grade
102
103        // Menutup Scanner untuk membebaskan resource java
104        input.close();
105    }

```

On the right, the build output window shows the following logs:

```

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
-----[ ! ]-----

```

At the bottom center, there is a message: "Activate Window".

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 informal	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 maupun yang akan dibahas minggu berikutnya dalam matakuliah PBO? Rencana saya yaitu saya mau belajar mandiri atau membuat satu program berkaitan dengan (Pewarisan).