

# **LAPORAN PEMROGRAMAN BERORIENTASI OBJEK**

## **MODUL KE – 4**

<<ARRAY OF OBJECTS>>



Oleh :

Nama : Yohanis Calvin D. P. U. Pati

NIM : 245314033

Kelas : BP

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS SANATA DHARMA  
SEMESTER GENAP 2025/2026**

## A. Tujuan

1. Mahasiswa mampu mendeklarasikan array of object.
2. Mahasiswa mampu mengalokasikan array of objects yang telah dideklarasikannya.
3. Mahasiswa mampu memasukkan data ke dalam array of objects.
4. Mahasiswa mampu memproses data dalam array of objects sesuai kepentingannya.

## B. Pelaksanaan Praktikum

### 1. Soal nomor 1 ( soal utama praktikum)

#### a. Soal

Buatlah class Mahasiswa dengan atribut nim, nama, tgl lahir bertipe String, sertauts1, uts2, uas masing-masing bertipe double. Class Mahasiswa mempunyai constructor, getter, setter serta metode untuk hitung nilai final dengan rumus :  $\text{final} = 30 \% \text{uts1} + 30 \% \text{uts2} + 40 \% \text{uas}$ .

Selain itu, ada metode untuk mengkonversi nilai final menjadi nilai huruf denganketentuan:

- A jika nilai final  $\geq 80$
- B jika nilai final  $\geq 70$  dan  $< 80$
- C jika nilai final  $\geq 56$  dan  $< 70$
- D jika nilai final  $\geq 45$  dan  $< 56$
- E jika nilai final  $< 45$ .

Buatlah main program yang dapat:

- mengisi data sejumlah N mahasiswa ke dalam array of Mahasiswa
- lalu menampilkan daftar nilai mahasiswa yang disertai dengan rata-rata nyabaikuntuk uts1, uts2, uas maupun nilai total dan nilai hurufnya dengan contohformatseperti di bawah ini
- menampilkan data mahasiswa yang memiliki nilai tertinggi

#### Daftar Mahasiswa Informatika Peserta PBO

No.	NIM	Nama	Tgl Lahir	Uts1	Uts2	Uas	Final	Nilai
1.	01251001	Edi	20/10/1995	67.40	70.60	85.00	...	B
2.	.....	.....	dst					
3.	.....	.....	dst					
.....								
-----								
Nilai tertinggi		Nim	: .....					
		Nama	: .....					
		Tgl lhr	: .....					
		Final	: .....					
		Nilai	: .....					

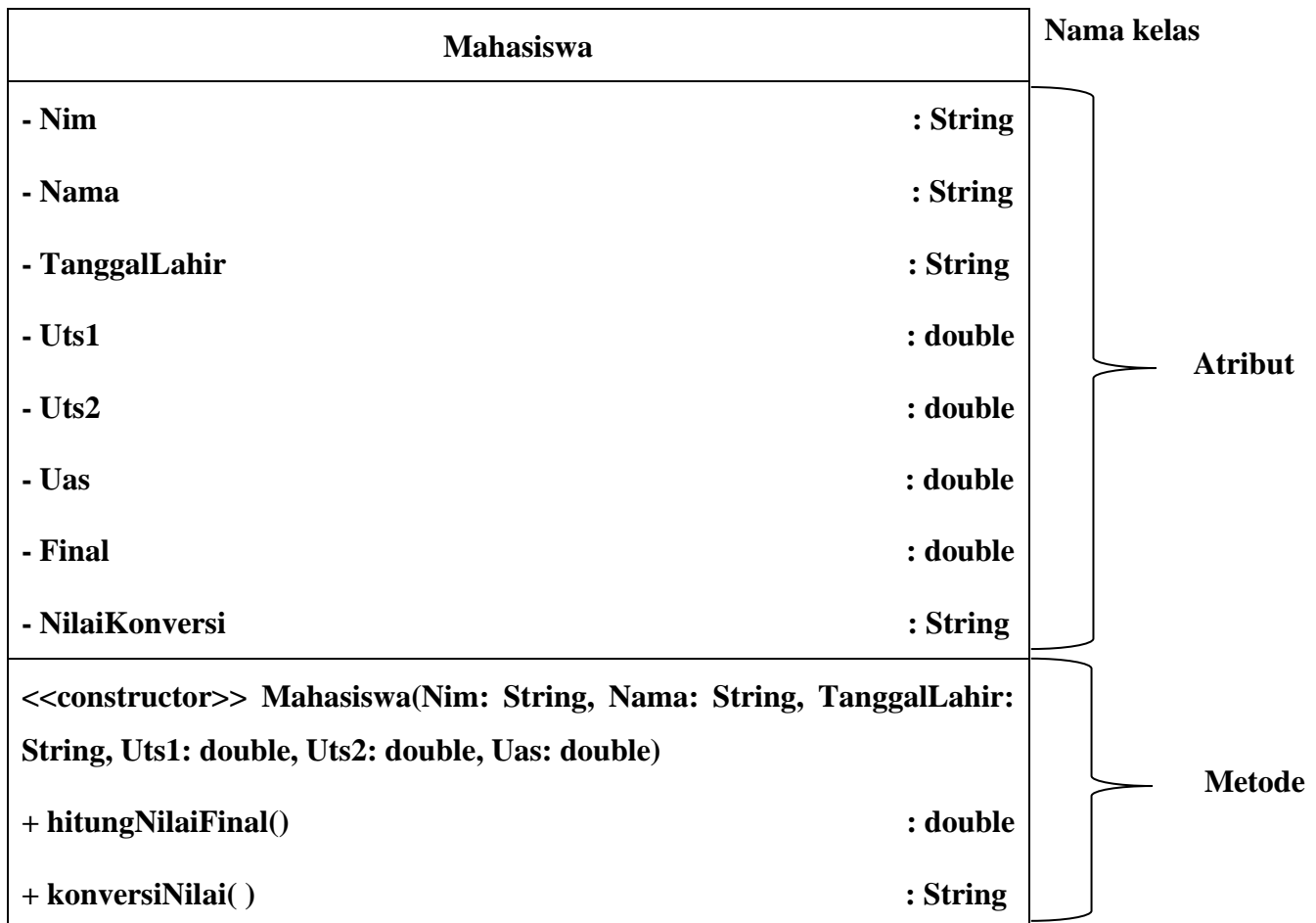
b. Gambaran umum:

- Penjelasan singkat kasus yang dipecahkan (1 – 2 kalimat)

Program ini adalah program untuk pengelolaan data mahasiswa yang dijalankan dengan mencatat NIM, nama, tanggal lahir, dan nilai uts1, uts2, dan uas. Jadi program ini dapat kita pakai untuk hitung nilai akhir mahasiswa sesuai bobot tertentu dan melakukan konversi menjadi nilai huruf, dan untuk menampilkan daftar mahasiswa dengan rata-rata nilai dan mahasiswa yang memiliki nilai tertinggi.

- Diagram kelas

Kelas atribut Mahasiswa dan class Main:



Assosiation



Main		Nama kelas
- mahasiswaList	: Mahasiswa[ ]	
- totalUTS1	: double	
- totalUTS2	: double	
- totalUAS	: double	
- totalFinal	: double	
- mahasiswaTerbaik	: Mahasiswa	
+ main(String[] args)	: void	Metode

- **Penjelasan diagram kelas yang dirancang (kelas-kelas yg ada dan artinya, relasi antar kelas).**

Program ini terdiri dari dua kelas atau program yaitu Mahasiswa dan Main.

### 1. Kelas Mahasiswa:

#### Fungsi:

Kelas ini berfungsi untuk representasi dan untuk mengelola nilai, dan data mahasiswa.

#### Atribut:

- **Nim, Nama, TanggalLahir** (atribut ini untuk menyimpan data identitas mahasiswa dan bersifat private dan bertipe string).
- **Uts1, Uts2, Uas** (atribut ini untuk menyimpan nilai ujian bertipe double dan bersifat private).
- **Final** (atribut ini berfungsi untuk menyimpan nilai akhir yang dihitung berdasarkan sesuai dengan bobot UTS1, UTS2, dan UAS dan bersifat private dan bertipe double).
- **NilaiKonversi** (atribut ini berfungsi untuk menyimpan nilai huruf: A, B, C, D, E dan bersifat private dan bertipe string).

#### Constructor:

- **<<constructor>> Mahasiswa(String, String, String, double, double, double)** konstruktor ini berfungsi untuk menginisialisasi semua atribut.

#### Metode:

- **hitungNilaiFinal():** metode ini berfungsi untuk menghitung nilai akhir dengan rumus matematika ini (30% UTS1 + 30% UTS2 + 40% UAS).
- **konversiNilai():** metode ini berfungsi untuk mengonversi nilai akhir (angka) menjadi huruf berdasarkan bobot nilai final sesuai dengan kriteria konversi (A–E).

## 2. Kelas Main:

### Fungsi:

kelas ini berfungsi sebagai tempat untuk menjalankan program.

### Atribut:

- **mahasiswaList: Mahasiswa[]** (array ini untuk menampung objek-objek Mahasiswa dan bersifat private).
- **totalUTS1, totalUTS2, totalUAS, totalFinal** (atribut ini berfungsi untuk menyimpan total nilai untuk keperluan perhitungan rata-rata dan bersifat private).
- **mahasiswaTerbaik: Mahasiswa** (atribut ini untuk menyimpan referensi ke mahasiswa yang memiliki nilai akhir tertinggi dan bersifat private).

### Metode:

- **Metode main(String[] args) : void** dalam metode ini berfungsi atau menjalankan fungsi untuk:
  - input data mahasiswa
  - menyimpan data dalam array **mahasiswaList**
  - menampilkan daftar mahasiswa
  - menghitung rata-rata nilai
  - menentukan mahasiswa dengan nilai tertinggi

## 3. Relasi antara kelas:

Relasi antara kelas Mahasiswa dan kelas Main bersifat association yang artinya bersifat:

- Main memakai kelas Mahasiswa sebagai model data, yang artinya kelas tersebut sebagai tempat untuk menyimpan informasi yang berkaitan dengan kasus atau data mahasiswa di kelas Mahasiswa.
- Didalam kelas Main, ada array atau koleksi objek-objek Mahasiswa yang dikelola dan diproses untuk menampilkan informasi, untuk menghitung nilai rata-rata dan mahasiswa nilai tertinggi.
- Kelas Main tidak mewarisi dari Mahasiswa, karena hanya bergantung dengan kelas Mahasiswa untuk struktur data dan metode, seperti perhitungan nilai akhir dan konversi nilai ke huruf.

Jadi pada dasarnya kelas main hanya berfungsi sebagai pengelola yang memanfaatkan atau menggunakan objek-objek dari kelas Mahasiswa untuk menjalankan program.

c. Program yang dibuat saat praktikum:

- Mahasiswa

```
9      * @author Celvin Pati
10     */
11     public class Mahasiswa {
12
13         String Nim;
14         String Nama;
15         String TanggalLahir;
16         double Uts1;
17         double Uts2;
18         double Uas;
19         double Final;
20         String NilaiKonversi;
21
22         public Mahasiswa(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) {
23             this.Nim = Nim;
24             this.Nama = Nama;
25             this.TanggalLahir = TanggalLahir;
26             this.Uts1 = Uts1;
27             this.Uts2 = Uts2;
28             this.Uas = Uas;
29         }
30
31         public String getNim() {
32             return Nim;
33         }
34
35         public String getNama() {
36             return Nama;
37         }
38
39         public String SetTanggalLahir() {
40             return TanggalLahir;
41         }
42
43         public double SetUts1() {
44             return Uts1;
45         }
46
47         public double SetUts2() {
48             return Uts2;
49         }
50
51         public double Uas() {
52             return Uas;
53         }
54
55         public double NilaiFinal() {
56             Final = (30 * Uts1) + (30 * Uts2) + (40 * Uas);
57             return Final;
58         }
59
60         public int Konversi() {
61             char NilaiFinal;
62             if (Final >= 80) {
63                 NilaiFinal = 'A';
64             } else if (Final >= 70 && Final < 80) {
65                 NilaiFinal = 'B';
66             } else if (Final >= 56 && Final < 70) {
67                 NilaiFinal = 'C';
68             } else if (Final >= 45 && Final < 56) {
69                 NilaiFinal = 'D';
70             } else {
71                 NilaiFinal = 'E';
72             }
73
74             return NilaiFinal;
75         }
76     }
77 }
```

Activate Windows  
Go to Settings to activate Windows.

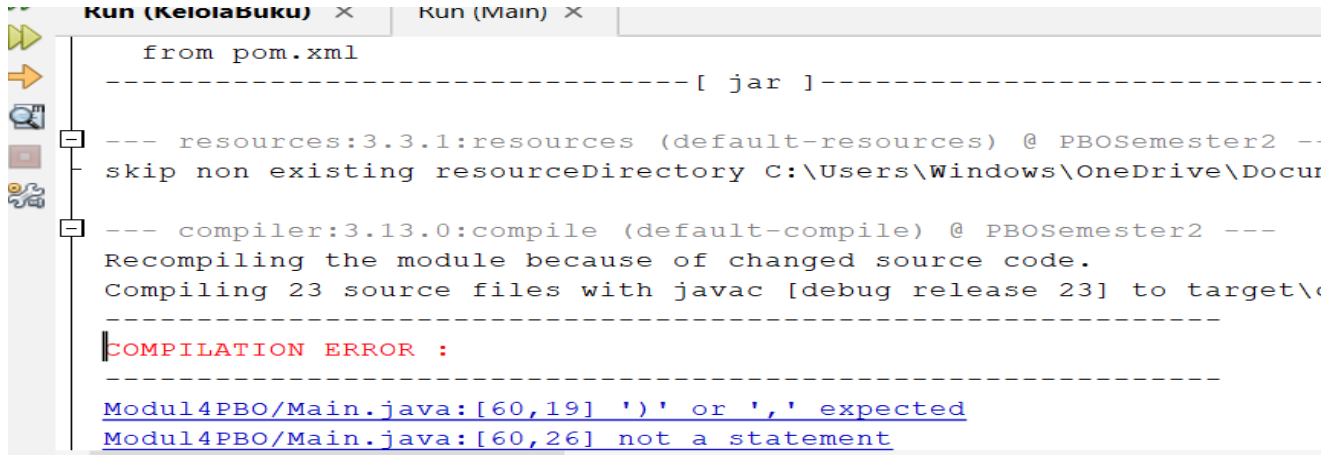
■ Main:

```

12  * @author Calvin Pati
13  */
14  public class Main {
15
16      public static void main(String[] args) {
17          Scanner scanner = new Scanner(System.in);
18
19          System.out.print("Masukkan jumlah mahasiswa: ");
20          int jumlah = scanner.nextInt();
21          scanner.nextLine();
22
23          Mahasiswa[] mahasiswa = new Mahasiswa[jumlah];
24
25
26          for (int i = 0; i < jumlah; i++) {
27              System.out.println("\nMahasiswa ke-" + (i + 1));
28              System.out.print("NIM: ");
29              String nim = scanner.nextLine();
30              System.out.print("Nama: ");
31              String nama = scanner.nextLine();
32              System.out.print("Tanggal Lahir (DD/MM/YYYY): ");
33              String tanggalLahir = scanner.nextLine();
34              System.out.print("Nilai UTS1: ");
35              double uts1 = scanner.nextDouble();
36              System.out.print("Nilai UTS2: ");
37              double uts2 = scanner.nextDouble();
38              System.out.print("Nilai UAS: ");
39              double uas = scanner.nextDouble();
40              scanner.nextLine(); // Konsumsi newline
41
42              mahasiswa[i] = new Mahasiswa(nim, nama, tanggalLahir, uts1, uts2, uas);
43          }
44
45
46          System.out.println("\nDaftar Mahasiswa:");
47          System.out.println("-----");
48          System.out.println("NIM      Nama      Tgl Lahir   UTS1   UTS2   UAS   Final   Nilai");
49          System.out.println("-----");
50          for (Mahasiswa mhs : mahasiswa) {
51              System.out.printf("    ",
52                  mhs.getNim(), mhs.getNama(), mhs.getTanggalLahir(),
53                  mhs.getUts1(), mhs.getUts2(), mhs.getUas(),
54                  mhs.hitungNilaiFinal(), mhs.konversiNilai());
55          }
56          System.out.println("-----");
57
58
59          Mahasiswa terbaik = mahasiswa[0];
60          for (Mahasiswa mhs : mahasiswa) {
61              if (mhs.hitungNilaiFinal() > terbaik.hitungNilaiFinal()) {
62                  terbaik = mhs;
63              }
64          }
65
66
67          System.out.println("\nMahasiswa dengan Nilai Tertinggi:");
68          System.out.println("NIM : " + terbaik.getNim());
69          System.out.println("Nama : " + terbaik.getNama());
70          System.out.println("Tgl Lhr : " + terbaik.getTanggalLahir());
71          System.out.println("Final : " + String.format("%.2f", terbaik.hitungNilaiFinal()));
72          System.out.println("Nilai : " + terbaik.konversiNilai());
73
74          scanner.close();
75      }
76  }

```

▪ Output:



The screenshot shows an IDE's Run console window with two tabs: 'Run (KelolaBuku)' and 'Run (Main)'. The 'Run (Main)' tab is active, displaying the following output:

```
from pom.xml
-----[ jar ]-----
--- resources:3.3.1:resources (default-resources) @ PBOSemester2 ---
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Docu

--- compiler:3.13.0:compile (default-compile) @ PBOSemester2 ---
Recompiling the module because of changed source code.
Compiling 23 source files with javac [debug release 23] to target\c
-----
||COMPILATION ERROR :
-----
Modul4PBO/Main.java:[60,19] ')' or ',' expected
Modul4PBO/Main.java:[60,26] not a statement
```



d. Perbaiki program yang dibuat di rumah disertai penjelasan dalam bentuk komentar pada program:

- kelas aribut:

```
9  * @author Calvin Pati
10 */
11 public class Mahasiswa {
12     // bagian ini untuk deklarasi atribut mahasiswa
13     String Nim;           // NIM mahasiswa
14     String Nama;          // Nama mahasiswa
15     String TanggalLahir;  // Tanggal lahir mahasiswa
16     double Uts1;          // Nilai UTS pertama
17     double Uts2;          // Nilai UTS kedua
18     double Uas;           // Nilai UAS
19     double Final;         // Nilai akhir mahasiswa (setelah perhitungan)
20     String NilaiKonversi; // Nilai huruf (A, B, C, D, atau E)
21
22     // bagian ini adalah constructor: yang dipakai untuk menginisialisasi objek Mahasiswa
23     public Mahasiswa(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) {
24         this.Nim = Nim;           // isi atribut Nim dengan nilai dari parameter
25         this>Nama = Nama;         // isi atribut Nama dengan nilai dari parameter
26         this.TanggalLahir = TanggalLahir; // isi atribut TanggalLahir dengan nilai dari parameter
27         this.Uts1 = Uts1;         // isi atribut Uts1 dengan nilai dari parameter
28         this.Uts2 = Uts2;         // isi atribut Uts2 dengan nilai dari parameter
29         this.Uas = Uas;           // isi atribut Uas dengan nilai dari parameter
30         this.Final = hitungNilaiFinal(); // hitung dan menyimpan nilai akhir ke atribut Final
31         this.NilaiKonversi = konversiNilai(); // konversi nilai akhir ke huruf dan menyimpannya
32     }
33
34     // ini adalah method untuk hitung nilai akhir mahasiswa
35     public double hitungNilaiFinal() {
36         // Perhitungan nilai akhir dengan bobot:
37         // 30% dari UTS1, 30% dari UTS2, dan 40% dari UAS
38         return (0.3 * Uts1) + (0.3 * Uts2) + (0.4 * Uas);
39     }
40
41     // ini adalah method untuk konversi nilai akhir menjadi nilai huruf (A, B, C, D, E)
42     public String konversiNilai() {
43         if (Final >= 80) return "A"; // Jika nilai akhir >= 80, maka mendapatkan A
44         else if (Final >= 70) return "B"; // Jika nilai akhir 70 - 79, maka mendapatkan B
45         else if (Final >= 56) return "C"; // Jika nilai akhir 56 - 69, maka mendapatkan C
46         else if (Final >= 45) return "D"; // Jika nilai akhir 45 - 55, maka mendapatkan D
47         else return "E"; // Jika nilai akhir < 45, maka mendapatkan E
48     }
49 } //akhir kelas mahasiswa
50
51
```

## ■ Main:

```
11  * @author Calvin Pati
12  */
13  public class Main {
14      public static void main(String[] args) {
15          Scanner input = new Scanner(System.in); // objek Scanner untuk baca input dari keyboard
16
17          // untuk memasukkan jumlah mahasiswa yang mau diinput
18          System.out.print("Masukkan jumlah mahasiswa: ");
19          int N = input.nextInt(); // untuk baca jumlah mahasiswa yang mau dimasukkan
20          input.nextLine(); // untuk newline buffer supaya input berikutnya tidak terlewat
21
22          // untuk buat array untuk simpan objek Mahasiswa sebanyak N
23          Mahasiswa[] mahasiswaList = new Mahasiswa[N];
24
25          // Perulangan untuk input data mahasiswa
26          for (int i = 0; i < N; i++) {
27              System.out.println("\nMasukkan data mahasiswa ke-" + (i + 1)); // untuk tampilkan nomor mahasiswa yang sedang d
28
29              System.out.print("NIM: ");
30              String nim = input.nextLine(); // Membaca NIM mahasiswa
31
32              System.out.print("Nama: ");
33              String nama = input.nextLine(); // Membaca Nama mahasiswa
34
35              System.out.print("Tanggal Lahir (dd/mm/yyyy): ");
36              String tglLahir = input.nextLine(); // Membaca tanggal lahir mahasiswa
37
38              System.out.print("UTS 1: ");
39              double uts1 = input.nextDouble(); // Membaca nilai UTS 1
40
41              System.out.print("UTS 2: ");
42              double uts2 = input.nextDouble(); // Membaca nilai UTS 2
43
44              System.out.print("UAS: ");
45              double uas = input.nextDouble(); // Membaca nilai UAS
46              input.nextLine(); // Membersihkan newline buffer agar input berikutnya tidak terlewat
47
48              // untuk simpan data mahasiswa ke dalam array
49              mahasiswaList[i] = new Mahasiswa(nim, nama, tglLahir, uts1, uts2, uas);
50          }
51
52          // untuk menampilkan daftar mahasiswa dalam bentuk tabel
53          System.out.println("\nDaftar Mahasiswa Informatika Peserta PBO");
54          System.out.println("-----");
55          // untuk menampilkan header tabel dengan format kolom yang rapi
56          System.out.printf("\n",
57              "No.", "NIM", "Nama", "Tgl Lahir", "UTS1", "UTS2", "UAS", "Final", "Nilai");
58          System.out.println("-----");
59
60          // Variabel untuk hitung total nilai dari semua mahasiswa
61          double totalUTS1 = 0, totalUTS2 = 0, totalUAS = 0, totalFinal = 0;
62          Mahasiswa mahasiswaTerbaik = mahasiswaList[0]; // Inisialisasi mahasiswa dengan nilai terbaik
63
64          // Perulangan untuk menampilkan data mahasiswa
65          for (int i = 0; i < N; i++) {
66              Mahasiswa mhs = mahasiswaList[i]; // Mengambil data mahasiswa dari array
67
68              // Menampilkan data mahasiswa dalam format tabel
69              System.out.printf("\n",
70                  (i + 1), mhs.Nim, mhs>Nama, mhs.TanggalLahir, mhs.Uts1, mhs.Uts2, mhs.Uas, mhs.Final, mhs.NilaiKonversi);
71
72              // Menambahkan nilai ke total untuk menghitung rata-rata
73              totalUTS1 += mhs.Uts1;
74              totalUTS2 += mhs.Uts2;
75              totalUAS += mhs.Uas;
76
77              // untuk menentukan mahasiswa dengan nilai tertinggi
78              if (mhs.Final > mahasiswaTerbaik.Final) {
79                  mahasiswaTerbaik = mhs;
80              }
81          }
82
83          // untuk menampilkan rata-rata nilai mahasiswa
84          System.out.println("-----");
85          System.out.printf("\n",
86              "Rata-rata", (totalUTS1 / N), (totalUTS2 / N), (totalUAS / N), (totalFinal / N)); // untuk cetak nilai rata-rata m
87          System.out.println("-----");
88
89          // untuk menampilkan informasi mahasiswa dengan nilai tertinggi
90          System.out.println("\nMahasiswa dengan nilai tertinggi:"); // untuk menampilkan teks Mahasiswa dengan nilai tertinggi
91          System.out.println("NIM      : " + mahasiswaTerbaik.Nim); // untuk cetak nim mahasiswa terbaik
92          System.out.println("Nama    : " + mahasiswaTerbaik>Nama); // untuk cetak nama mahasiswa terbaik
93          System.out.println("Tgl Lhr  : " + mahasiswaTerbaik.TanggalLahir); // untuk cetak tanggal lahir mahasiswa terbaik
94          System.out.println("Final    : " + mahasiswaTerbaik.Final); // untuk cetak nilai final mahasiswa terbaik
95          System.out.println("Nilai    : " + mahasiswaTerbaik.NilaiKonversi); // untuk cetak nilai konversi mahasiswa terbaik
96
97
98
```

▪ Output:

```
Run (KelolaBuku) × Run (Main) ×
--- compiler:3.13.0:compile (default-compile) @ Modul4PBOfix ---
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) @ Modul4PBOfix ---
Masukkan jumlah mahasiswa: 5

Masukkan data mahasiswa ke-1
NIM: 245314033
Nama: celvin
Tanggal Lahir (dd/mm/yyyy): 24 04 2006
UTS 1: 90
UTS 2: 98
UAS: 99

Masukkan data mahasiswa ke-2
NIM: 245314031
Nama: cia
Tanggal Lahir (dd/mm/yyyy): 21 04 2006
UTS 1: 91
UTS 2: 90
UAS: 98

Masukkan data mahasiswa ke-3
NIM: 245314030
Nama: delta
Tanggal Lahir (dd/mm/yyyy): 17 04 2006
UTS 1: 90
UTS 2: 91
UAS: 90

Masukkan data mahasiswa ke-4
NIM: 245314029
Nama: elvian
Tanggal Lahir (dd/mm/yyyy): 21
UTS 1: 90
UTS 2: 87
UAS: 99

Masukkan data mahasiswa ke-5
NIM: 245314028
Nama: Dimas
Tanggal Lahir (dd/mm/yyyy): 25 12 2004
UTS 1: 99
UTS 2: 90
UAS: 90
UAS. 90

Daftar Mahasiswa Informatika Peserta PBO
-----
No.  NIM      Nama      Tgl Lahir  UTS1  UTS2  UAS  Final  Nilai
-----
1    245314033  celvin    24 04 2006  90,00  98,00  99,00  96,00  A
2    245314031  cia       21 04 2006  91,00  90,00  98,00  93,50  A
3    245314030  delta     17 04 2006  90,00  91,00  90,00  90,30  A
4    245314029  elvian    21         90,00  87,00  99,00  92,70  A
5    245314028  Dimas     25 12 2004  99,00  90,00  90,00  92,70  A
-----
Rata-rata                92,00  91,20  95,20  93,04
-----

Mahasiswa dengan nilai tertinggi:
NIM      : 245314033
Nama     : celvin
Tgl Lhr  : 24 04 2006
Final    : 96.0
Nilai    : A
-----
BUILD SUCCESS
-----
Total time: 03:12 min
Finished at: 2025-03-17T00:47:32+07:00
-----
.
```

e. Analisis perbandingan program saat praktikum dan perbaikan program:

- **perbedaan kedua program:**

- ✓ **Struktur dan Kode yang Lebih Rapi:** Program versi perbaikan lebih terstruktur karena memakai komentar pada setiap baris program yang lebih jelas.
- ✓ **Metode yang Lebih Efisien:** Program versi bisah perbaikan langsung hitung nilai final di dalam konstruktor, sedangkan versi awal masih memisahkan proses itu tanpa pemanggilan yang tepat.
- ✓ **Perbaikan Format Output:** Di versi perbaikan saya memakai printf untuk menampilkan tabel dengan format yang rapi.
- ✓ **Menambahkan Rata-Rata Nilai:** Versi perbaikan saya dapat menghitung dan menampilkan rata-rata UTS1, UTS2, UAS, dan Final, yang tidak ada dalam versi awal.
- ✓ **Pemilihan Mahasiswa Terbaik Lebih Baik:** jadi di versi awal, pemilihan mahasiswa terbaik kurang efektif karena kesalahan dalam perbandingan nilai jadi kurang efisien.

**Kekurangan program pertama (sebelum perbaikan):**

1. Di program pertama sebelum perbaikan tidak ada perhitungan nilai final saat objek dibuat, yang menyebabkan Final selalu bernilai 0.
2. Di program sebelum perbaikan metode konversi nilai tidak mengembalikan nilai yang sesuai jadi hasil konversi tidak tepat.
3. Pada program pertama tidak ada tampilan rata-rata nilai UTS1, UTS2, UAS, dan Final jadi output program tidak sesuai dengan instruksi soal.
4. Output tabel kurang rapi dan sulit dibaca dan juga output masih eror saat program di run karena pemanggilan fungsi yang tidak tepat pada program kelas Main.
5. Pemilihan mahasiswa dengan nilai tertinggi kurang optimal karena program pertama saya tidak ada tampilan tampilan rata-rata nilai UTS1, UTS2, UAS, dan Final.

- **Kelebihan program kedua (setelah perbaikan)**

1. Program kedua memakai metode yang lebih baik dalam perhitungan nilai final dan konversi nilai sehingga hasil perhitungan lebih tepat dan efisien.
2. Di program kedua bisah menampilkan data dengan lebih rapi dan mudah dibaca menggunakan printf karena struktur program yang lebih tertata dan rapi.
3. Di program kedua bisa menampilkan informasi tambahan seperti rata-rata nilai dan mahasiswa terbaik.

4. Dan di program kedua saat program di run dapat menghasilkan aoutput tanpa eror dansesuai dengan instruksi soal yang menampilkan nilai rata-rata, data mahasiswa yang diinput, nilai final, hasil konversi dan mahasiswa dengan nilai tertinggi.

- **kesimpulan yang bisa dipetik dari perbedaan itu**

jadi dapat saya lihat dan pahami bahwa pada program versi perbaikan terdapat struktur kode yang lebih rapi, efisien dalam perhitungan, dan output yang jelas, jadi karena struktur program yang lebih baik dan efisien program kedua dapat dengan mudah dipahami, dan juga lebih akurat dalam menampilkan data serta lebih informatif.

Makadari itu dalam proses debugging juga saya dapat lebih mudah memahami kode, jadi hal ini dapat membuktikan jika perancangan program yang lebih baik dapat memberikan manfaat banyak bagi program itu sendiri dan pengguna yang menjadikannya lebih efektif, efisien, dan mudah dikembangkan.

## 2. soal nomer 2 (Soal utama praktikum)

### a. soal

Buatlah program untuk mengelola daftar buku. Setiap **buku** memiliki ciri kode buku, judul, tahun terbit, dan nama pengarang.

Kelas buku

Buatlah kelas main bernama **KelolaBuku** yang melakukan beberapa hal sebagai berikut:

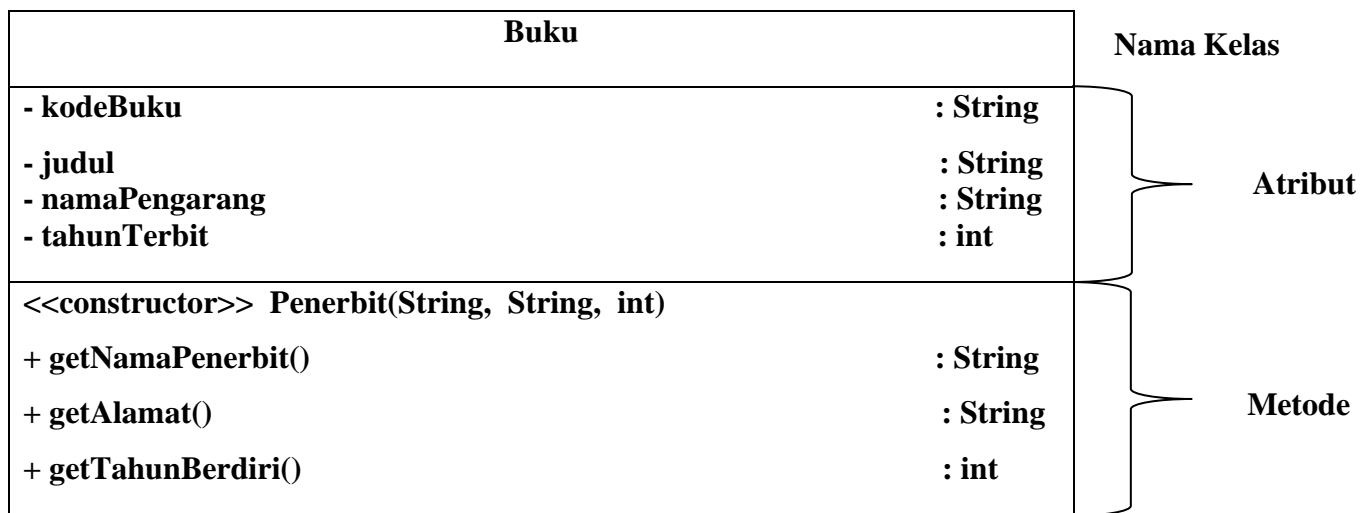
- Meminta masukan berapa jumlah buku baru yang akan disimpan.
- Membuat obyek buku dan memasukkan kode buku, judul, tahun terbit, dan nama pengarang.
- Menampilkan daftar buku lengkap.
- Menampilkan **jumlah buku** yang terbit pada **tahun tertentu**. Tahun yang dikehendaki dimasukkan lewat keyboard.
- Menampilkan **data buku** yang ditulis oleh **pengarang tertentu**. Nama pengarang dimasukkan lewat keyboard.

### b. gambaran umum:

- **Penjelasan singkat kasus yang dipecahkan (1 – 2 kalimat)**

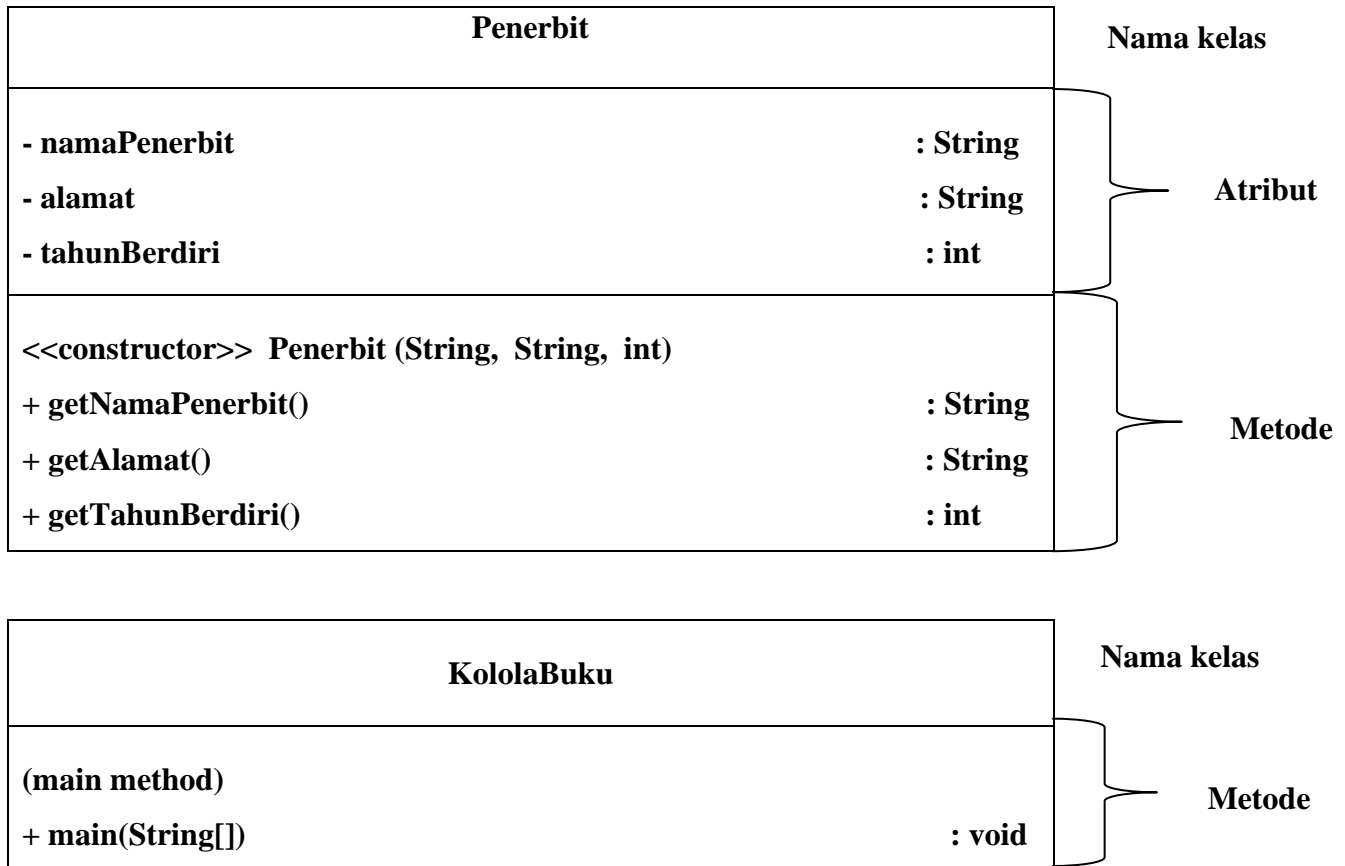
Jadi program ini bertujuan atau berfungsi untuk mengelola data buku dan penerbitnya dan juga dengan fitur menambah, menyimpan, dan menampilkan data sesuai dengan berbagai kriteria, seperti tahun terbit, dan nama pengarang. Jadi dengan ini program dapat membantu pengguna untuk mudah dalam mencari dan mengelola informasi buku dalam system katalog sederhana.

- **Diagram kelas**



Assosiation





- Penjelasan diagram kelas yang dirancang (kelas-kelas yg ada dan artinya, relasi antar kelas)

## 1. Kelas Buku

### Fungsi:

Kelas buku berfungsi untuk representasi data suau buku, dan juga kode buku, judu, nama pengarang, dan tahun terbit, dan juga tersedia metode untuk menampilkan informasi buku.

### Atribut:

- kodeBuku Bersifat private, bertipe String, dan berfungsi untuk menyimpan kode buku.
- judul bersifat private, bertipe String, dan berfungsi untuk menyimpan judul buku.
- namaPengarang bersifat private, bertipe String, dan berfungsi untuk menyimpan nama penulis buku.
- tahunTerbit besifat private, bertipe int, dan berfungsi untuk menyimpan tahun terbit.

### Metode:

- Buku(String, String, String, int) bersifat public, bertipe konstruktor, dan berfungsi untuk inialisasi objek Buku dengan atribut kodeBuku, judul, namaPengarang, dan tahunTerbit.
- getKodeBuku() bersifat public, bertipe String, dan berfungsi untuk mengembalikan kode buku.
- getJudul() bersifat public, bertipe String, dan berfungsi untuk mengembalikan judul buku.
- getNamaPengarang() bersifat public, bertipe String, dan berfungsi untuk mengembalikan nama pengarang.

- getTahunTerbit() bersifat public, bertipe String, dan berfungsi untuk mengembalikan tahun terbit buku.
- tampilkanBuku() bersifat public, bertipe Void, dan berfungsi untuk menampilkan informasi buku dalam format table.

## 2. Kelas Penerbit

### Fungsi:

Kelas ini dipakai untuk menyimpan informasi tentang penerbit buku, yaitu seperti nama penerbit, alamat, dan tahun berdiri, dan juga metode untuk mengakses data tersebut.

### Atribut:

- namaPenerbit bersifat private, bertipe String, dan berfungsi untuk menyimpan nama penerbit.
- alamat bersifat private, bertipe String, dan berfungsi untuk menyimpan alamat penerbit.
- tahunBerdiri bersifat private, bertipe int, dan berfungsi untuk menyimpan tahun berdirinya penerbit.

### Metode:

- Penerbit(String, String, int) bersifat public, bertipe konstruktor, dan berfungsi untuk menginisialisasi objek Penerbit dengan atribut namaPenerbit, alamat, dan tahunBerdiri.
- getNamaPenerbit() bersifat public, bertipe String, dan berfungsi untuk mengembalikan nama penerbit.
- getAlamat() bersifat public, bertipe String, dan berfungsi untuk mengembalikan alamat penerbit.
- getTahunBerdiri() bersifat public, bertipe int, dan berfungsi untuk mengembalikan tahun berdiri penerbit.

## 3. Kelas main

### Fungsi:

Kelas Main berfungsi sebagai kelas utama yang mengelola daftar buku, dan dapat membantu pengguna untuk menambah buku, menampilkan daftar buku, dan juga mencari buku sesuai dengan tahun terbit dan nama pengarang.

- + main(String[] args) bersifat public, bertipe void, dan berfungsi untuk menjalankan program, menerima input dari pengguna dan mencetak hasil proses program.

## 4. Relasi antara kelas:

- Relasi antara **KelolaBuku** dan **Buku**
  - Kelas KelolaBuku mempunyai array Buku[ ] daftarBuku, yang artinya kelas ini terdapat kumpulan objek Buku.
  - Setiap buku di daftar mempunyai informasi yang bisa ditampilkan dan dicari sesuai dengan tahun atau pengarang.
- Relasi antara **Buku** dan **penerbit**
  - Jadi relasi antara Buku dan Penerbit memiliki satu penerbit yang artinya setiap



objek Buku bisah mempunyai informasi penerbit yang berasal dari kelas Penerbit.

- Jadi kesimpulannya kelas `kelolaBuku` mempunyai `Buku[ ]` `daftarBuku` / relasi komposisi, dan kelas `Buku` mempunyai `Penerbit` (jika ditambahkan) dan jika tanpa kelas `Buku`, kode `KelolaBuu` tidak bisah berjalan, karena memakai objek `Buku`.

c. Kode program saat praktikum dan versi perbaikan

- **Sebelum perbaikan**
- **Kelas Buku:**

```
Source History
9      * @author Calvin Pati
10     */
11     public class Buku {
12         String kodeBuku, judul, namaPengarang;
13         int tahunTerbit;
14
15         // Konstruktor
16         public Buku(String kodeBuku, String judul, String namaPengarang, int tahunTerbit) {
17             this.kodeBuku = kodeBuku;
18             this.judul = judul;
19             this.namaPengarang = namaPengarang;
20             this.tahunTerbit = tahunTerbit;
21         }
22
23         // Getter untuk atribut buku
24         public String getKodeBuku() {
25             return kodeBuku;
26         }
27
28         public String getJudul() {
29             return judul;
30         }
31
32         public String getNamaPengarang() {
33             return namaPengarang;
34         }
35
36         public int getTahunTerbit() {
37             return tahunTerbit;
38         }
39
40         // Metode untuk menampilkan informasi buku dalam format tabel
41         public void tampilkanBuku() {
42             System.out.printf("%-10s %-20s %-15s %-10d\n", kodeBuku, judul, namaPengarang, tahunTerbit);
43         }
44     }
45
46     // Kelas Penerbit untuk menyimpan informasi tentang penerbit buku
47     class Penerbit {
48         String namaPenerbit, alamat;
49         int tahunBerdiri;
50
51         // Konstruktor
52         public Penerbit(String namaPenerbit, String alamat, int tahunBerdiri) {
53             this.namaPenerbit = namaPenerbit;
54             this.alamat = alamat;
55             this.tahunBerdiri = tahunBerdiri;
56         }
57
58         // Getter untuk atribut penerbit
59         public String getNamaPenerbit() {
60             return namaPenerbit;
61         }
62
63         public String getAlamat() {
64             return alamat;
65         }
66
67         public int getTahunBerdiri() {
68             return tahunBerdiri;
69         }
70     }
```

## ■ Kelas KelolaBuku

```
11  * @author Calvin Pati
12  */
13  public class kelolaBuku {
14      public static void main(String[] args) {
15          Scanner input = new Scanner(System.in);
16
17          // Meminta jumlah buku yang akan dimasukkan
18          System.out.print("Masukkan jumlah buku: ");
19          int jumlahBuku = input.nextInt();
20          input.nextLine(); // Membersihkan newline buffer
21
22          Buku[] daftarBuku = new Buku[jumlahBuku]; // Array untuk menyimpan daftar
23
24          // Input data buku
25          for (int i = 0; i < jumlahBuku; i++) {
26              System.out.println("\nMasukkan data buku ke-" + (i + 1));
27
28              System.out.print("Kode Buku: ");
29              String kode = input.nextLine();
30
31              System.out.print("Judul: ");
32              String judul = input.nextLine();
33
34              System.out.print("Nama Pengarang: ");
35              String pengarang = input.nextLine();
36
37              System.out.print("Tahun Terbit: ");
38              int tahun = input.nextInt();
39              input.nextLine(); // Membersihkan newline buffer
40
41              // Menyimpan objek buku ke dalam array daftarBuku
42              daftarBuku[i] = new Buku(kode, judul, pengarang, tahun);
43          }
44
45          // Menampilkan daftar buku dalam bentuk tabel
46          System.out.println("\nDaftar Buku:");
47          System.out.println("-----");
48          System.out.printf("%-10s %-20s %-15s %-10s\n", "Kode", "Judul", "Pengarang", "Tahun Terbit");
49          System.out.println("-----");
50          for (Buku buku : daftarBuku) {
51              buku.tampilkanBuku();
52          }
53          System.out.println("-----");
54
55          // Menampilkan jumlah buku yang terbit pada tahun tertentu
56          System.out.print("\nMasukkan tahun terbit yang ingin dicari: ");
57          int tahunCari = input.nextInt();
58          int jumlahTahun = 0;
59
60          System.out.println("\nBuku yang terbit pada tahun " + tahunCari + ":");
61          for (Buku buku : daftarBuku) {
62              if (buku.getTahunTerbit() == tahunCari) {
63                  buku.tampilkanBuku();
64                  jumlahTahun++;
65              }
66          }
67
68          if (jumlahTahun == 0) {
69              System.out.println("Tidak ada buku yang terbit pada tahun tersebut.");
70          }
71
72          // Menampilkan data buku berdasarkan pengarang tertentu
73          input.nextLine(); // Membersihkan newline buffer
74          System.out.print("\nMasukkan nama pengarang yang ingin dicari: ");
75          String pengarangCari = input.nextLine();
76          int jumlahPengarang = 0;
77
78          System.out.println("\nBuku yang ditulis oleh " + pengarangCari + ":");
79          for (Buku buku : daftarBuku) {
80              if (buku.getNamaPengarang().equalsIgnoreCase(pengarangCari)) {
81                  buku.tampilkanBuku();
82                  jumlahPengarang++;
83              }
84          }
85
86          if (jumlahPengarang == 0) {
87              System.out.println("Tidak ada buku yang ditulis oleh pengarang tersebut.");
88          }
89
90          input.close(); // Menutup scanner untuk menghindari kebocoran resource
91      }
92  }
```

## ■ Output

```
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\NetBeansProjects\PBOSemester2

--- compiler:3.13.0:compile (default-compile) @ PBOSemester2 ---
Recompiling the module because of changed source code.
Compiling 22 source files with javac [debug release 23] to target\classes

--- exec:3.1.0:exec (default-cli) @ PBOSemester2 ---
Masukkan jumlah buku: 3

Masukkan data buku ke-1
Kode Buku: 200
Judul: belajar java
Nama Pengarang: Cia
Tahun Terbit: 2015

Masukkan data buku ke-2
Kode Buku: 203
Judul: belajar python
Nama Pengarang: celvin
Tahun Terbit: 2023

Masukkan data buku ke-3
Kode Buku: 201
Judul: belajar C++
Nama Pengarang: Delvitri
Tahun Terbit: 2020

Daftar Buku:
-----
Kode      Judul          Pengarang      Tahun Terbit
-----
200      belajar java    Cia            2015
203      belajar python  celvin         2023
201      belajar C++     Delvitri       2020
-----

Masukkan tahun terbit yang ingin dicari: 2015

Buku yang terbit pada tahun 2015:
200      belajar java    Cia            2015

Masukkan nama pengarang yang ingin dicari: celvin

Buku yang ditulis oleh celvin:
203      belajar python  celvin         2023
-----
BUILD SUCCESS
-----

Total time: 02:50 min
Finished at: 2025-03-17T13:33:32+07:00
-----
```

- Versi perbaikan
- Kelas Buku

```
10  * @author Calvin Pati
11  */
12  public class Buku { // awal Kelas Buku untuk objek buku
13      // Atribut untuk simpan informasi buku
14      String kodeBuku, judul, namaPengarang;
15      int tahunTerbit;
16
17      // Konstruktor untuk menginisialisasi objek Buku dengan data tertentu
18      public Buku(String kodeBuku, String judul, String namaPengarang, int tahunTerbit) {
19          this.kodeBuku = kodeBuku; // untuk simpan kode buku
20          this.judul = judul; // untuk simpan judul buku
21          this.namaPengarang = namaPengarang; // untuk simpan nama pengarang
22          this.tahunTerbit = tahunTerbit; // untuk simpan tahun terbit buku
23      }
24
25      // Getter untuk mendapatkan kode buku
26      public String getKodeBuku() {
27          return kodeBuku;
28      }
29
30      // Getter untuk mendapatkan judul buku
31      public String getJudul() {
32          return judul;
33      }
34
35      // Getter untuk mendapatkan nama pengarang buku
36      public String getNamaPengarang() {
37          return namaPengarang;
38      }
39
40      // Getter untuk mendapatkan tahun terbit buku
41      public int getTahunTerbit() {
42          return tahunTerbit;
43      }
44
45      // Metode untuk menampilkan informasi buku dalam format tabel
46      public void tampilkanBuku() {
47          // untuk menampilkan data buku dengan format tabel menggunakan printf
48          System.out.printf("%-10s %-20s %-15s %-10d\n", kodeBuku, judul, namaPengarang, tahunTerbit);
49      }
50  }
51
52  // Kelas Penerbit untuk simpan informasi tentang penerbit buku
53  class Penerbit {
54      // Atribut untuk menyimpan informasi penerbit
55      String namaPenerbit, alamat;
56      int tahunBerdiri;
57
58      // Konstruktor untuk menginisialisasi objek Penerbit dengan data tertentu
59      public Penerbit(String namaPenerbit, String alamat, int tahunBerdiri) {
60          this.namaPenerbit = namaPenerbit; // untuk simpan nama penerbit
61          this.alamat = alamat; // untuk simpan alamat penerbit
62          this.tahunBerdiri = tahunBerdiri; // untuk simpan tahun berdiri penerbit
63      }
64
65      // Getter untuk mendapatkan nama penerbit
66      public String getNamaPenerbit() {
67          return namaPenerbit;
68      }
69
70      // Getter untuk mendapatkan alamat penerbit
71      public String getAlamat() {
72          return alamat;
73      }
74
75      // Getter untuk mendapatkan tahun berdiri penerbit
76      public int getTahunBerdiri() {
77          return tahunBerdiri;
78      }
79  } // akhir kelas Buku
```

## ■ Kelas KelolaBuku

```
11  * @author Calvin Pati
12  */
13  public class KelolaBuku { // Kelas awal utama untuk kelola daftar buku
14  public static void main(String[] args) {
15      // Membuat objek Scanner untuk menerima input dari pengguna
16      Scanner input = new Scanner(System.in);
17
18      // untuk meminta jumlah buku yang mau dimasukkan
19      System.out.print("Masukkan jumlah buku: ");
20      int jumlahBuku = input.nextInt(); // untuk baca jumlah buku dari input pengguna
21      input.nextLine(); // Membersihkan newline buffer agar tidak terlewat saat membaca input berikutnya
22
23      // Array untuk simpan daftar buku yang dimasukkan oleh pengguna
24      Buku[] daftarBuku = new Buku[jumlahBuku];
25
26      // Proses input data buku dari pengguna
27      for (int i = 0; i < jumlahBuku; i++) {
28          System.out.println("\nMasukkan data buku ke-" + (i + 1));
29
30          // untuk memasukkan informasi buku
31          System.out.print("Kode Buku: ");
32          String kode = input.nextLine(); // untuk baca kode buku dari input pengguna
33
34          System.out.print("Judul: ");
35          String judul = input.nextLine(); // untuk baca judul buku
36
37          System.out.print("Nama Pengarang: ");
38          String pengarang = input.nextLine(); // untuk baca nama pengarang buku
39
40          System.out.print("Tahun Terbit: ");
41          int tahun = input.nextInt(); // untuk baca tahun terbit buku
42          input.nextLine(); // untuk membersihkan newline buffer agar tidak mengganggu input berikutnya
43
44          // untuk simpan objek buku ke dalam array daftarBuku
45          daftarBuku[i] = new Buku(kode, judul, pengarang, tahun);
46      }
47
48      // Menampilkan daftar buku yang telah dimasukkan dalam bentuk tabel
49      System.out.println("\nDaftar Buku:");
50      System.out.println("-----");
51      System.out.printf("%-10s %-20s %-15s %-10s\n", "Kode", "Judul", "Pengarang", "Tahun Terbit");
52      System.out.println("-----");
53
54      // Loop untuk menampilkan setiap buku yang sudah dimasukkan
55      for (Buku buku : daftarBuku) {
56          buku.tampilkanBuku();
57      }
58
59      System.out.println("-----");
60
61      // untuk menampilkan jumlah buku yang terbit pada tahun tertentu
62      System.out.print("\nMasukkan tahun terbit yang ingin dicari: ");
63      int tahunCari = input.nextInt(); // untuk baca tahun yang ingin dicari
64      int jumlahTahun = 0; // Variabel untuk hitung jumlah buku yang terbit pada tahun tertentu
65
66      System.out.println("\nBuku yang terbit pada tahun " + tahunCari + ":");
67
68      // Loop untuk cari buku yang terbit pada tahun tertentu
69      for (Buku buku : daftarBuku) {
70          if (buku.getTahunTerbit() == tahunCari) {
71              buku.tampilkanBuku(); // Menampilkan buku yang ditemukan
72              jumlahTahun++;
73          }
74      }
75
76      // Jika tidak ada buku yang terbit pada tahun tersebut
77      if (jumlahTahun == 0) {
78          System.out.println("Tidak ada buku yang terbit pada tahun tersebut.");
79      }
80
81      // untuk menampilkan data buku berdasarkan pengarang tertentu
82      input.nextLine(); // Membersihkan newline buffer sebelum membaca input berikutnya
83      System.out.print("\nMasukkan nama pengarang yang ingin dicari: ");
84      String pengarangCari = input.nextLine(); // untuk baca nama pengarang yang ingin dicari
85      int jumlahPengarang = 0; // Variabel untuk hitung jumlah buku sesuai dengan pengarang
86
87      System.out.println("\nBuku yang ditulis oleh " + pengarangCari + ":");
88
89      // Loop untuk cari buku sesuai dengan pengarang tertentu
90      for (Buku buku : daftarBuku) {
91          if (buku.getNamaPengarang().equalsIgnoreCase(pengarangCari)) { // untuk cek kesamaan nama pengarang (case-i
92              buku.tampilkanBuku(); // Menampilkan buku yang ditulis oleh pengarang tersebut
93              jumlahPengarang++;
94          }
95      }
96
97      // Jika tidak ada buku yang ditulis oleh pengarang tersebut
98      if (jumlahPengarang == 0) {
99          System.out.println("Tidak ada buku yang ditulis oleh pengarang tersebut.");
100     }
101
102     // untuk tutup scanner untuk menghindari kebocoran resource
103     input.close();
104 }
```

## ▪ Output

```
Output Run (PBOSemester2)
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\NetBeansProjects\PBOSemester2
--- compiler:3.13.0:compile (default-compile) @ PBOSemester2 ---
Recompiling the module because of changed source code.
Compiling 22 source files with javac [debug release 23] to target\classes
--- exec:3.1.0:exec (default-cli) @ PBOSemester2 ---
Masukkan jumlah buku: 3

Masukkan data buku ke-1
Kode Buku: 200
Judul: belajar java
Nama Pengarang: Cia
Tahun Terbit: 2015

Masukkan data buku ke-2
Kode Buku: 203
Judul: belajar python
Nama Pengarang: celvin
Tahun Terbit: 2023

Masukkan data buku ke-3
Kode Buku: 201
Judul: belajar C++
Nama Pengarang: Delvitri
Tahun Terbit: 2020

Daftar Buku:
-----
Kode      Judul          Pengarang      Tahun Terbit
-----
200       belajar java   Cia            2015
203       belajar python celvin         2023
201       belajar C++    Delvitri       2020
-----

Masukkan tahun terbit yang ingin dicari: 2015

Buku yang terbit pada tahun 2015:
200      belajar java      Cia            2015

Masukkan nama pengarang yang ingin dicari: celvin

Buku yang ditulis oleh celvin:
203      belajar python    celvin         2023
-----
BUILD SUCCESS
-----
Total time: 02:50 min
Finished at: 2025-03-17T13:33:32+07:00
-----
```

d. Analisis perbandingan program saat praktikum dan perbaikan program:

- **Perbedaan kedua program**

Perbedaan utama antara kedua program yaitu versi perbaikan memiliki komentar yang lebih banyak, yang bisah membantu pengguna untuk memahami setiap bagian kode secara detail, jadi lebih mudah dipahami. Dan juga, ada sedikit perbaikan dalam struktur kode seperti penamaan kelas utama yang diperbaiki menjadi `KelolaBuku` jadi sesuai dengan konversi java.

- **Kekurangan program pertama:**

1. Di program pertama kurang banyak komentar yang menjelaskan fungsi dari setiap bagian kode, jadi lebih sulit dipahami oleh pembaca baru.
2. Nama kelas utama (`kelolaBuku`) tidak sesuai dengan konvensi Java yang seharusnya memakai huruf kapital di awal (`KelolaBuku`).
3. Di program pertama penekanan masih kurang pada pembersihan newline buffer (`input.nextLine();`), jadi bisah menyebabkan error saat membaca input berikutnya.
4. Di nprogram pertama hasil / output masih kurang rapih karena struktur program yang kurang efisien

- **Kelebihan program versi perbaikan**

1. Di program versi perbaikan lebih banyak menambahkan komentar di setiap bagian kode untuk memperjelas fungsinya jadi siapapun bisah dengan mudah memahami struktur code program.
2. Perbaikan penamaan kelas (`KelolaBuku`) di program versi perbaikan lebih sesuai dengan konversi java jadi lebih efisien karena penamaan kelas diawali huruf kapital.
3. Pembersihan buffer `input.nextLine();` lebih jelas jadi program bisa menghindari kesalahan saat menerima input.
4. Struktur kode lebih rapi dan mudah dipahami dan karena itu juga jadi saat program di run outputnya berhasil tanpa eror dan rapih

- **Kesimpulan dari perbedaan program:**

- Program yang dibuat di rumah lebih rapi, lebih mudah dipahami, dan sudah sesuai dengan standar Java dibandingkan dengan program saat praktikum.
- penulisan kode sesuai konvensi (contohnya seperti penamaan kelas, metode, dan variabel) akan meningkatkan keterbacaan dan pemeliharaan kode.
- Menambahkan komentar dalam kode jadi kode bisah dipahami dengan lebih mudah.
- Pada dasarnya versi perbaikan lebih baik terkait keterbacaan dan struktur kode karena penambahan komentar yang membantu pemahaman, perbaikan penamaan kelas, dan juga penanganan input yang lebih baik. Jadi dapat membuat program lebih mudah untuk dipelihara



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

- MODUL IV PRAKTIKUM PBO ARRAY OF OBJECTS
- Poe AI
- <https://www.petanikode.com/java-oop/>

D. Refleksi proses belajar

- Apakah Anda semakin paham ATAU semakin tidak paham? : Ya....
- Mengapa? : karena saat saya dapat membandingkan kekurangan kelebihan program ang saya buat saat praktikum dengan program versi perbaikan sehingga saya bisah evalusasi dan upgrade pemahaman saya.
- Apa rencana Anda selanjutnya terkait topik yang sudah dibahas maupunyaiakan dibahas minggu berikutnya dalam matakuliah PBO? : rencana saya saya mau lebih mengoptimalkan persiapan saya sebelum masuk perkuliah agar dapat memberikan hasil maksimal terkait pree test dan saat mengerjakan program saat praktikum.