

Laporan Modul 2: Dasar Pemrograman Java

Mata Kuliah: Praktikum Pemrograman Berorientasi Objek

Nama: Muhammad Alif Arrayyan

NIM: 2024573010085

Kelas: Teknik Informatika - 2E

1. Abstrak

Laporan ini membahas konsep-konsep dasar dalam bahasa pemrograman Java yang mencakup variabel, tipe data, operator, struktur kontrol keputusan (if-else, switch-case), dan struktur kontrol perulangan (for, while, do-while). Tujuan dari laporan ini adalah untuk mendokumentasikan pemahaman dan hasil dari praktikum dasar-dasar Java sebagai fondasi untuk mempelajari pemrograman berorientasi objek lebih lanjut.

2. Praktikum

Praktikum 1 - Variabel dan Tipe Data

Dasar Teori

Variabel adalah sebuah nama yang digunakan sebagai wadah atau lokasi di dalam memori komputer untuk menyimpan sebuah nilai. Nilai yang disimpan ini dapat diubah selama program berjalan. Agar komputer tahu jenis nilai apa yang boleh disimpan, setiap variabel harus memiliki tipe data. Tipe data mendefinisikan jenis dan ukuran data, seperti int untuk bilangan bulat, double untuk bilangan desimal, char untuk satu karakter, dan String untuk kumpulan karakter atau teks. Deklarasi variabel dengan tipe data yang tepat sangat penting untuk memastikan operasi yang dilakukan pada data tersebut valid dan efisien.

Langkah Praktikum

1. Pertama-tama, di dalam package modul_2 tersebut, buatlah sebuah class Java baru dan simpan dengan nama VariabelDemo.java.
2. Setelah file berhasil dibuat, isikan dengan potongan kode yang telah disiapkan untuk mempraktikkan cara deklarasi variabel dan penerapan tipe data.

Screenshoot Hasil

Program-VariabelDemo.java

```
1 package modul_2;
2
3 public class VariabelDemo {
4     public static void main(String[] args) {
5         // Deklarasi dan inisialisasi variabel
6         int umur = 18;
7         double tinggi = 164.5;
8         char grade = 'A';
9         boolean lulus = true;
10        String nama = "Muhammad Alif Arsyavan";
11
12        // Menampilkan nilai variabel
13        System.out.println("Nama: " + nama);
14        System.out.println("Umur: " + umur + " tahun");
15        System.out.println("Tinggi: " + tinggi + " cm");
16        System.out.println("Grade: " + grade);
17        System.out.println("Lulus: " + lulus);
18
19        int a = 10;
20        int b = 3;
21
22        System.out.println("\n=== Operasi Aritmatika ===");
23        System.out.println("a = " + a);
24        System.out.println("b = " + b);
25        System.out.println("a + b = " + (a + b));
26        System.out.println("a - b = " + (a - b));
27        System.out.println("a * b = " + (a * b));
28        System.out.println("a / b = " + (a / b));
29        System.out.println("a % b = " + (a % b));
30
31        System.out.println("\n=== Casting Tipe Data ===");
32        double nilaiDouble = 9.8;
33        int nilaiInt = (int) nilaiDouble;
34
35        System.out.println("Nilai double: " + nilaiDouble);
36        System.out.println("Setelah casting ke int: " + nilaiInt);
37
38        // Automatic promotion
39        byte byteVar = 50;
40        short shortVar = 100;
41        int hasil = byteVar + shortVar;
42        System.out.println("byte + short = " + hasil + " (tipe: int)");
43    }
44 }
45 }
```

The screenshot shows an IDE with a project named 'praktikum-pbo-20245'. The file 'VariabelDemo.java' is open, showing the following code:

```

1 package modul_2;
2
3 public class VariabelDemo { new *
4     public static void main(String[] args) { new *
5         // Deklarasi dan inisialisasi variabel
6         int umur = 18;
    }
}

```

The Run console shows the output of the program:

```

"E:\KULIAH SMT 3\Prak Beriontasi Objek - Pak Reza\Amazon Corretto\jdk21.0.8_9\bin\java
Nama: Muhammad Alif Arrayyan
Umur: 18 tahun
Tinggi: 164.5 cm
Grade: A
Lulus: true

=== Operasi Aritmatika ===
a = 10
b = 3
a + b = 13
a - b = 7
a * b = 30
a / b = 3
a % b = 1

=== Casting Tipe Data ===
Nilai double: 9.8
Setelah casting ke int: 9
byte + short = 150 (tipe: int)

```

Analisa dan Pembahasan

Pada praktikum ini, program berhasil mendemonstrasikan konsep paling fundamental dalam Java, yaitu deklarasi dan inisialisasi variabel dengan tipe data yang sesuai. Penggunaan tipe data yang berbeda seperti String untuk teks, int untuk bilangan bulat, dan double untuk desimal menunjukkan sifat statically typed pada Java, di mana setiap variabel harus memiliki tipe yang jelas. Operasi aritmatika yang dilakukan juga berjalan sesuai dengan aturan matematika dasar. Praktikum ini menegaskan bahwa pemahaman variabel dan tipe data adalah langkah awal yang krusial karena menjadi dasar untuk menyimpan, mengelola, dan memanipulasi semua data di dalam program.

Praktikum 2 - Input, Output dan Scanner

Dasar Teori

Dalam pemrograman, Input adalah proses menerima data dari luar program, misalnya dari ketikan pengguna di keyboard. Sebaliknya, Output adalah proses mengirimkan data atau informasi dari dalam program ke luar, seperti menampilkannya di layar. Di Java, salah satu cara paling umum untuk menangani input dari pengguna adalah dengan menggunakan kelas Scanner. Kelas Scanner menyediakan fungsi-fungsi untuk membaca berbagai jenis data (seperti teks, angka, dll.) dari sumber input, sehingga program dapat berinteraksi dengan pengguna secara dinamis.

Langkah Praktikum

1. Awali praktikum ini dengan membuat sebuah file Java baru di dalam package modul_2 Anda, lalu berikan nama InputOutputDemo.java.
2. Di dalam file tersebut, implementasikan kode program yang memanfaatkan class Scanner untuk menangkap data yang diketik oleh pengguna melalui konsol.

Screenshoot Hasil

Program-InputOutputDemo.Java

```
1 package modul_2;
2
3 import java.util.Scanner;
4
5 public class InputOutputDemo { new *
6     public static void main(String[] args) { new *
7         // Membuat objek Scanner
8         Scanner input = new Scanner(System.in);
9
10        // Membaca input string
11        System.out.print("Masukkan nama Anda: ");
12        String nama = input.nextLine();
13
14        // Membaca input integer
15        System.out.print("Masukkan umur Anda: ");
16        int umur = input.nextInt();
17
18        // Membaca input double
19        System.out.print("Masukkan tinggi Anda (cm): ");
20        double tinggi = input.nextDouble();
21
22        // Menampilkan output
23        System.out.println("\n=== DATA ANDA ===");
24        System.out.println("Nama: " + nama);
25        System.out.println("Umur: " + umur + " tahun");
26        System.out.println("Tinggi: " + tinggi + " cm");
27
28        // Menutup Scanner
29        input.close();
30    }
31 }
32
```

```

5 public class InputOutputDemo { new *
6     public static void main(String[] args) { new *
13
14         // Membaca input integer
15         System.out.print("Masukkan umur Anda: ");
16         int umur = input.nextInt();
17
18         // Membaca input double
19         System.out.print("Masukkan tinggi Anda (cm): ");
20         double tinggi = input.nextDouble();
21
22         // Menampilkan output
23         System.out.println("\n=== DATA ANDA ===");
24         Svstem.out.println("Nama: " + nama);

```

Run InputOutputDemo x

```

"E:\KULIAH SMT 3\Prak Beriontasi Objek - Pak Reza\Amazon Corretto\jdk21.0.8_9\bin\java.exe"
Masukkan nama Anda: Muhammad Alif Arrayyan
Masukkan umur Anda: 18
Masukkan tinggi Anda (cm): 164.5

=== DATA ANDA ===
Nama: Muhammad Alif Arrayyan
Umur: 18 tahun
Tinggi: 164.5 cm

```

Analisa dan Pembahasan

Program berhasil menunjukkan kemampuan interaksi dasar dengan pengguna melalui console. Penggunaan kelas Scanner terbukti efektif untuk menangkap berbagai jenis input (String, int, double) dari keyboard. Analisanya adalah program ini berhasil menjembatani antara pengguna dan mesin, di mana data tidak lagi bersifat statis (ditulis langsung di kode), melainkan dinamis sesuai masukan pengguna. Pembahasan pentingnya adalah keharusan untuk mengimpor pustaka `java.util.Scanner` sebelum digunakan dan pentingnya menggunakan metode yang tepat (`nextLine()`, `nextInt()`, `nextDouble()`) untuk menghindari error saat parsing data. Praktikum ini adalah fondasi untuk membuat aplikasi console yang interaktif.

Praktikum 3 - Struktur Kontrol: Percabangan

Dasar Teori

Struktur Kontrol Percabangan adalah mekanisme dalam pemrograman yang memungkinkan alur eksekusi program untuk memilih jalur yang berbeda berdasarkan kondisi tertentu. Jika suatu kondisi bernilai benar (`true`), maka blok kode tertentu akan dieksekusi. Jika bernilai salah (`false`), blok kode tersebut akan dilewati atau blok kode alternatif yang akan dijalankan. Struktur ini adalah fondasi untuk membuat program yang dapat mengambil keputusan. Contoh paling umum dari struktur ini adalah perintah `if-else` dan `switch-case`.

Langkah Praktikum

1. Program Penentu Grade: Untuk studi kasus pertama, ciptakan file baru bernama GradeDemo.java. Di dalamnya, tuliskan logika program yang menggunakan struktur if-else if-else untuk mengonversi nilai menjadi sebuah predikat (misalnya, A, B, C).
2. Program Menu dengan Switch: Lanjutkan dengan membuat file bernama MenuDemo.java. Kali ini, manfaatkan switch statement untuk membangun sebuah sistem pemilihan menu yang sederhana dan efisien.
3. Program Nested If: Terakhir, buatlah file NestedIfDemo.java. Di sini, Anda akan mengimplementasikan sebuah program yang menggunakan if bersarang untuk menentukan kategori usia berdasarkan masukan dari pengguna.

Screenshoot Hasil

Program-GradeDemo.Java

```
1 package modul_2;
2
3 import java.util.Scanner;
4
5 public class GradeDemo {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8
9         System.out.print("Masukkan nilai (0-100): ");
10        int nilai = input.nextInt();
11
12        char grade;
13        String keterangan;
14
15        // Menggunakan if-else if-else
16        if (nilai >= 85) {
17            grade = 'A';
18            keterangan = "Excellent";
19        } else if (nilai >= 75) {
20            grade = 'B';
21            keterangan = "Good";
22        } else if (nilai >= 65) {
23            grade = 'C';
24            keterangan = "Fair";
25        } else if (nilai >= 55) {
26            grade = 'D';
27            keterangan = "Poor";
28        } else {
29            grade = 'E';
30            keterangan = "Fail";
31        }
32
33        System.out.println("Nilai: " + nilai);
34        System.out.println("Grade: " + grade);
35        System.out.println("Keterangan: " + keterangan);
36
37        input.close();
38    }
39 }
```

Output-GradeDemo.Java

The screenshot shows an IDE window with the following components:

- Project Explorer:** Displays the project structure for 'praktikum-pbo-2024573010085'. It includes folders like '.idea', '.jekyll-cache', '_site', 'laporan', and 'laporan1'. Under 'laporan1', there is a 'gambar' folder and files 'InputOutputDemo.p' and 'VariabelDemo.png'. The 'src' folder is also visible.
- Code Editor:** Shows the 'GradeDemo.java' file. The code is as follows:

```
5 public class GradeDemo { new *
6     public static void main(String[] args) { new *
33         System.out.println("Nilai: " + nilai);
34         System.out.println("Grade: " + grade);
35         System.out.println("Keterangan: " + keterangan);
36
37         input.close();
38     }
39 }
```
- Run Console:** Shows the execution of the 'GradeDemo' class. The command used is `"E:\KULIAH SMT 3\Prak Beriontasi Objek - Pak Reza\Amazon Corretto\jdk21.0.8_9\bin\java.exe"`. The output is:

```
Masukkan nilai (0-100): 91
Nilai: 91
Grade: A
Keterangan: Excellent
```

Program-MenuDemo.Java

```

2
3 import java.util.Scanner;
4
5 public class MenuDemo {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8
9         System.out.println("=== MENU PILIHAN ===");
10        System.out.println("1. Hitung Luas Persegi");
11        System.out.println("2. Hitung Luas Lingkaran");
12        System.out.println("3. Hitung Luas Segitiga");
13        System.out.println("4. Keluar");
14
15        System.out.print("Pilih menu (1-4): ");
16        int pilihan = input.nextInt();
17
18        switch (pilihan) {
19            case 1:
20                System.out.print("Masukkan sisi persegi: ");
21                double sisi = input.nextDouble();
22                double luasPersegi = sisi * sisi;
23                System.out.println("Luas persegi: " + luasPersegi);
24                break;
25
26            case 2:
27                System.out.print("Masukkan jari-jari lingkaran: ");
28                double jariJari = input.nextDouble();
29                double luasLingkaran = Math.PI * jariJari * jariJari;
30                System.out.println("Luas lingkaran: " + luasLingkaran);
31                break;
32
33            case 3:
34                System.out.print("Masukkan alas segitiga: ");
35                double alas = input.nextDouble();
36                System.out.print("Masukkan tinggi segitiga: ");
37                double tinggi = input.nextDouble();
38                double luasSegitiga = 0.5 * alas * tinggi;
39                System.out.println("Luas segitiga: " + luasSegitiga);
40                break;
41
42            case 4:
43                System.out.println("Terima kasih!");
44                break;
45
46            default:
47                System.out.println("Pilihan tidak valid!");

```

Output-MenuDemo.Java


```
1 package modul_2;
2
3 import java.util.Scanner;
4
5 public class MenuDemo {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8
9         System.out.println("=== MENU PILIHAN ===");
10        System.out.println("1. Hitung Luas Persegi");
11        System.out.println("2. Hitung Luas Lingkaran");
12        System.out.println("3. Hitung Luas Segitiga");
13        System.out.println("4. Keluar");
14    }
```

Run MenuDemo x

"E:\KULIAH SMT 3\Prak Beriontasi Objek - Pak Reza\Amazon Corretto\jdk21.0.8_9\bin\java.exe

=== MENU PILIHAN ===

1. Hitung Luas Persegi

2. Hitung Luas Lingkaran

3. Hitung Luas Segitiga

4. Keluar

Pilih menu (1-4): 1

Masukkan sisi persegi: 4

Luas persegi: 16.0

Program-NestedIfDemo.Java

```

1 package modul_2;
2
3 import java.util.Scanner;
4
5 public class NestedIfDemo {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8
9         System.out.print("Masukkan umur: ");
10        int umur = input.nextInt();
11
12        if (umur >= 0) {
13            if (umur <= 2) {
14                System.out.println("Kategori: Bayi");
15            } else if (umur <= 5) {
16                System.out.println("Kategori: Balita");
17            } else if (umur <= 12) {
18                System.out.println("Kategori: Anak-anak");
19            } else if (umur <= 19) {
20                System.out.println("Kategori: Remaja");
21            } else if (umur <= 59) {
22                System.out.println("Kategori: Dewasa");
23            } else {
24                System.out.println("Kategori: Lansia");
25            }
26        } else {
27            System.out.println("Umur tidak valid!");
28        }
29
30        input.close();
31    }
32 }

```

Output-NestedIfDemo.Java

```
1 package modul_2;
2
3 import java.util.Scanner;
4
5 public class NestedIfDemo { new *
6     public static void main(String[] args) { new *
7         Scanner input = new Scanner(System.in);
8
9         System.out.print("Masukkan umur: ");
10        int umur = input.nextInt();
11
12        if (umur >= 0) {
13            if (umur <= 2) {
14                System.out.println("Kategori: Remaja");
15            }
16        }
17    }
18 }
```

Run NestedIfDemo x

"E:\KULIAH SMT 3\Prak Beriontasi Objek - Pak Reza\Amazon Corretto\jdk21.0.8_9\bin\java.exe"

Masukkan umur: 18

Kategori: Remaja

Analisa dan Pembahasan

Praktikum ini berhasil mengimplementasikan logika pengambilan keputusan untuk mengontrol alur program.

- Pada GradeDemo.java, penggunaan struktur if-else if-else sangat efektif untuk menangani kondisi berupa rentang nilai, yang tidak bisa ditangani oleh switch-case.
- Pada MenuDemo.java, struktur switch-case terbukti lebih rapi dan mudah dibaca dibandingkan if-else if untuk kondisi yang memeriksa satu variabel dengan banyak kemungkinan nilai konstan.
- Pada NestedIfDemo.java, penggunaan if bersarang menunjukkan cara membangun logika yang lebih kompleks dan bertingkat, seperti validasi multi-langkah. Pembahasannya, pemilihan antara if-else dan switch bergantung pada kasusnya. if-else lebih fleksibel untuk kondisi boolean yang kompleks, sementara switch lebih unggul untuk percabangan berdasarkan nilai tunggal yang diskrit.

Praktikum 4 - Struktur Kontrol: Perulangan

Dasar Teori

Struktur Kontrol Perulangan (atau looping) digunakan untuk mengeksekusi blok kode yang sama secara berulang-ulang selama suatu kondisi masih terpenuhi. Ini sangat efisien untuk tugas-tugas yang repetitif, seperti memproses semua elemen dalam daftar atau menjalankan simulasi hingga mencapai target tertentu. Tanpa perulangan, kita harus menulis kode yang sama berkali-kali. Jenis-jenis perulangan yang umum digunakan di Java meliputi for, while, dan do-while, yang masing-masing memiliki cara tersendiri dalam mendefinisikan kondisi berhenti dan iterasinya.

Langkah Praktikum

1. Perulangan For: Silakan buat file baru bernama ForLoopDemo.java. Isilah file tersebut dengan berbagai contoh implementasi for loop untuk memahami strukturnya.
2. Perulangan While dan Do-While: Berikutnya, buat file bernama WhileLoopDemo.java. Tuliskan kode untuk mendemonstrasikan penggunaan while dan do-while, serta pahami perbedaan utama antara keduanya.
3. Nested Loop (Perulangan Bersarang): Sebagai penutup, buatlah sebuah file bernama NestedLoopDemo.java. Implementasikan contoh kode yang menunjukkan bagaimana perulangan bisa dijalankan di dalam perulangan lain untuk menyelesaikan masalah yang lebih kompleks.

Screenshoot Hasil

Program-ForLoopDemo.Java

```
1 package modul_2;
2
3 public class ForLoopDemo { new *
4     public static void main(String[] args) { new *
5         // Contoh 1: Menampilkan angka 1-10
6         System.out.println("=== Angka 1-10 ===");
7         for (int i = 1; i <= 10; i++) {
8             System.out.print(i + " ");
9         }
10        System.out.println();
11
12        // Contoh 2: Menampilkan angka ganj-
13        System.out.println("\n=== Angka Ganj 2-20 ===");
14        for (int i = 2; i <= 20; i += 2) {
15            System.out.print(i + " ");
16        }
17        System.out.println();
18
19        // Contoh 3: Countdown
20        System.out.println("\n=== Countdown ===");
21        for (int i = 10; i >= 1; i--) {
22            System.out.print(i + " ");
23        }
24        System.out.println("Start!");
25
26        // Contoh 4: Tabel perkalian
27        System.out.println("\n=== Tabel Perkalian 5 ===");
28        for (int i = 1; i <= 10; i++) {
29            System.out.println("5 x " + i + " = " + (5 * i));
30        }
31    }
32 }
```

Output-ForLoopDemo.Java

The screenshot shows an IDE with a project named "praktikum-pbo-2024573010085". The main file is "WhileLoopDemo.java". The code in the editor is as follows:

```
1 package modul_2;
2
3 public class ForLoopDemo { new *
4     public static void main(String[] args) { new *
5         // Contoh 1: Menampilkan angka 1-10
6         System.out.println("=== Angka 1-10 ===");
```

The Run console shows the following output:

```
=== Angka 1-10 ===
1 2 3 4 5 6 7 8 9 10

=== Angka Genap 2-20 ===
2 4 6 8 10 12 14 16 18 20

=== Countdown ===
10 9 8 7 6 5 4 3 2 1 Start!

=== Tabel Perkalian 5 ===
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Program-WhileLoopDemo.Java

```

5 public class WhileLoopDemo { new *
6     public static void main(String[] args) { new *
10         System.out.println("=== While Loop - Tebak Angka ===");
11         int angkaRahasia = 7;
12         int tebakan = 0; // Inisialisasi dengan nilai yang salah
13
14         while (tebakan != angkaRahasia) {
15             System.out.print("Tebak angka (1-10): ");
16             tebakan = input.nextInt();
17
18             if (tebakan < angkaRahasia) {
19                 System.out.println("Terlalu kecil!");
20             } else if (tebakan > angkaRahasia) {
21                 System.out.println("Terlalu besar!");
22             } else {
23                 System.out.println("Benar! Angka rahasianya adalah " + angkaRahasia);
24             }
25         }
26
27         // Contoh Do-While Loop
28         System.out.println("\n=== Do-While Loop - Menu ===");
29         int pilihan;
30
31         do {
32             System.out.println("\n1. Tampilkan pesan");
33             System.out.println("2. Hitung faktorial");
34             System.out.println("3. Keluar");
35             System.out.print("Pilih menu: ");
36             pilihan = input.nextInt();
37
38             switch (pilihan) {
39                 case 1:
40                     System.out.println("Hello, World!");
41                     break;
42                 case 2:
43                     System.out.print("Masukkan angka: ");
44                     int n = input.nextInt();
45                     long faktorial = 1;
46                     for (int i = 1; i <= n; i++) {
47                         faktorial = faktorial * i;
48                     }
49                     System.out.println("Faktorial " + n + " = " + faktorial);
50                     break;
51                 case 3:
52                     System.out.println("Terima kasih!");
53                     break;

```

Output-WhileLoopDemo.Java

Project ▾ M+ laporan.md WhileLoopDemo.java x Nested

gambiar

- ForLoopDemo.png
- GradeDemo.png
- InputOutputDemo.png
- MenuDemo.png
- NestedIfDemo.png

Run WhileLoopDemo x

```
"E:\KULIAH SMT 3\Prak Beriontasi Objek - Pak Reza\Amazon Corretto\jdk21.0.8.
=== While Loop - Tebak Angka ===
Tebak angka (1-10): 6
Terlalu kecil!
Tebak angka (1-10): 8
Terlalu besar!
Tebak angka (1-10): 7
Benar! Angka rahasianya adalah 7

=== Do-While Loop - Menu ===

1. Tampilkan pesan
2. Hitung faktorial
3. Keluar
Pilih menu: 3
Terima kasih!

Process finished with exit code 0
```

Program-NestedLoopDemo.Java

```

1 package modul_2;
2
3 public class NestedLoopDemo {
4     public static void main(String[] args) {
5         // Contoh 1: Pola Bintang
6         System.out.println("=== Pola Bintang Segitiga ===");
7         for (int i = 1; i <= 5; i++) {
8             for (int j = 1; j <= i; j++) {
9                 System.out.print("* ");
10            }
11            System.out.println();
12        }
13
14        // Contoh 2: Tabel Perkalian
15        System.out.println("\n=== Tabel Perkalian 1-5 ===");
16        for (int i = 1; i <= 5; i++) {
17            for (int j = 1; j <= 5; j++) {
18                System.out.printf("%3d ", (i * j));
19            }
20            System.out.println();
21        }
22
23        // Contoh 3: Pola Angka
24        System.out.println("\n=== Pola Angka ===");
25        for (int i = 1; i <= 4; i++) {
26            for (int j = 1; j <= i; j++) {
27                System.out.print(j + " ");
28            }
29            System.out.println();
30        }
31    }
32 }

```

Output-NestedLoopDemo.Java

The screenshot shows an IDE with a project named 'praktikum-pbo-2024573010085'. The file 'NestedLoopDemo.java' is open, showing the following code:

```
1 package modul_2;
2
3 public class NestedLoopDemo { new *
4     public static void main(String[] args) { new *
5         // Contoh 1: Pola Bintang
6         System.out.println("=== Pola Bintang Segitiga ===");
```

The Run console shows the output of the program:

```
"E:\KULIAH SMT 3\Prak Beriontasi Objek - Pak Reza\Amazon Corretto\jdk21.0.8_9\bin\java.exe" "-j
=== Pola Bintang Segitiga ===
*
* *
* * *
* * * *
* * * * *

=== Tabel Perkalian 1-5 ===
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25

=== Pola Angka ===
1
1 2
1 2 3
1 2 3 4
```

Analisa dan Pembahasan

Program berhasil mendemonstrasikan cara melakukan tugas berulang secara efisien.

- ForLoopDemo.java: Perulangan for terbukti sangat cocok untuk iterasi yang jumlahnya sudah diketahui secara pasti (misalnya, mengulang 10 kali), karena inisialisasi, kondisi, dan increment didefinisikan dalam satu baris yang ringkas.
- WhileLoopDemo.java: Perulangan while menunjukkan fungsinya dalam kasus di mana perulangan bergantung pada sebuah kondisi yang bisa berubah saat runtime dan jumlah iterasinya tidak pasti.
- NestedLoopDemo.java: Perulangan bersarang berhasil digunakan untuk memproses data dalam format dua dimensi, seperti membuat pola bintang. Logikanya, loop luar mengontrol baris, dan loop dalam mengontrol kolom. Pembahasannya, ketiga jenis perulangan memiliki kegunaan spesifik. Kesalahan dalam mendefinisikan kondisi pada while dapat menyebabkan infinite loop, sedangkan perulangan bersarang menunjukkan bagaimana kompleksitas sebuah algoritma dapat meningkat secara signifikan.

3. Kesimpulan

Praktikum Modul 2 ini memberikan pemahaman yang esensial mengenai fondasi pemrograman Java. Melalui serangkaian latihan, praktikan telah berhasil memahami konsep inti mulai dari cara menyimpan data menggunakan variabel dan tipe data, hingga membuat program menjadi interaktif dengan mekanisme input dan output menggunakan class Scanner.

Lebih lanjut, praktikum ini juga melatih kemampuan logika dalam mengontrol alur eksekusi program melalui struktur percabangan (if-else, switch) untuk pengambilan keputusan, serta struktur perulangan (for, while, do-while) untuk menangani tugas-tugas repetitif secara efisien. Penguasaan konsep-konsep dasar ini merupakan pilar utama yang akan menjadi bekal untuk membangun aplikasi Java yang lebih kompleks dan fungsional di masa mendatang.

5. Referensi

Berikut adalah beberapa sumber yang relevan dengan materi praktikum:

1. W3Schools Java Tutorial — <https://www.w3schools.com/java/>
2. The Java™ Tutorials: Language Basics (Oracle Documentation) — <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>