

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMOGRAMAN
PERULANGAN
PEKAN 5

Disusun Oleh :

KINAYA NOVRYA MANDA
(2511531016)

Dosen Pengampu :

DR. WAHYUDI, S.T, M.T

Asisten Praktikum :

MUHAMMAD ZAKI AL HAFIZ



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis sampaikan kehadirat Allah SWT. Salawat dan salam disampaikan kepada Nabi Muhammad SAW. Karena thaufik dan hidayah-Nya, laporan praktikum Java ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban dari kegiatan praktikum yang telah dilaksanakan, sekaligus sebagai sarana untuk memperdalam pemahaman mengenai konsep dasar serta penerapan bahasa pemrograman Java.

Melalui praktikum ini, penulis memperoleh pengalaman langsung dalam memahami struktur perulangan yang memungkinkan program membuat keputusan dan menjalankan blok kode yang berbeda berdasarkan kondisi tertentu menggunakan Java. Diharapkan laporan ini dapat memberikan gambaran yang jelas mengenai materi yang telah dipelajari serta hasil dari percobaan yang dilakukan selama praktikum berlangsung.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa yang akan datang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten laboratorium, serta semua pihak yang telah memberikan bimbingan, arahan, dan dukungan sehingga laporan praktikum ini dapat tersusun. Semoga laporan ini dapat bermanfaat bagi pembaca, khususnya dalam memahami dasar-dasar pemrograman Java.

Padang, Oktober 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
DAFTAR LAMPIRAN	iii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	2
1.3 Manfaat Praktikum.....	2
BAB 2 PEMBAHASAN	
2.1 Kode Program.....	3
2.2 Langkah Kerja	10
2.3 Analisis Hasil.....	18
BAB 3 KESIMPULAN	
3.1 Hasil Praktikum	20
3.2 Saran Pengembangan	24
DAFTAR KEPUSTAKAAN	26

DAFTAR LAMPIRAN

Kode Program 2.1 (Kode Program PerulanganFor1)	3
Kode Program 2.2 (Kode Program PerulanganFor2)	4
Kode Program 2.3 (Kode Program PerulanganFor3)	5
Kode Program 2.4 (Kode Program PerulanganFor4)	6
Kode Program 2.5 (Kode Program nestedFor0).....	7
Kode Program 2.6 (Kode Program nestedFor1).....	8
Kode Program 2.7 (Kode Program nestedFor2).....	9
Gambar 2.1 (Langkah 1 PerulanganFor1).....	10
Gambar 2.2 (Langkah 2 PerulanganFor1).....	10
Gambar 2.3 (Langkah 3 PerulanganFor1).....	11
Gambar 2.4 (Langkah 1 PerulanganFor2).....	11
Gambar 2.5 (Langkah 2 PerulanganFor2).....	11
Gambar 2.6 (Langkah 3 PerulanganFor2).....	12
Gambar 2.7 (Langkah 1 PerulanganFor3).....	12
Gambar 2.8 (Langkah 2 PerulanganFor3).....	12
Gambar 2.9 (Langkah 3 PerulanganFor3).....	13
Gambar 2.10 (Langkah 4 PerulanganFor3).....	13
Gambar 2.11 (Langkah 1 PerulanganFor4).....	13
Gambar 2.12 (Langkah 2 PerulanganFor4).....	14
Gambar 2.13 (Langkah 3 PerulanganFor4).....	14
Gambar 2.14 (Langkah 4 PerulanganFor4).....	14
Gambar 2.15 (Langkah 1 nestedFor0)	15
Gambar 2.16 (Langkah 2 nestedFor0)	15
Gambar 2.17 (Langkah 3 nestedFor0)	15

Gambar 2.18 (Langkah 4 nestedFor0)	16
Gambar 2.19 (Langkah 1 nestedFor1)	16
Gambar 2.20 (Langkah 2 nestedFor1)	16
Gambar 2.21 (Langkah 3 nestedFor1)	17
Gambar 2.22 (Langkah 4 nestedFor1)	17
Gambar 2.23 (Langkah 1 nestedFor2)	17
Gambar 2.24 (Langkah 2 nestedFor2)	18
Gambar 2.25 (Langkah 3 nestedFor2)	18
Gambar 3.1 (Hasil Praktikum PerulanganFor1).....	20
Gambar 3.2 (Hasil Praktikum PerulanganFor2).....	20
Gambar 3.3 (Hasil Praktikum PerulanganFor3).....	21
Gambar 3.4 (Hasil Praktikum PerulanganFor4).....	22
Gambar 3.5 (Hasil Praktikum nestedFor0)	22
Gambar 3.6 (Hasil Praktikum nestedFor1)	23
Gambar 3.7 (Hasil Praktikum nestedFor2)	23

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pemrograman merupakan salah satu dasar penting dalam dunia teknologi informasi, di mana setiap sistem komputer bergantung pada logika dan algoritma untuk menjalankan perintah. Salah satu konsep penting yang harus dipahami dalam pemrograman adalah struktur kendali perulangan (*looping*). Perulangan digunakan untuk mengeksekusi perintah yang sama secara berulang tanpa harus menulis kode secara manual berulang kali. Melalui pemahaman struktur ini, programmer dapat menulis kode yang lebih efisien, ringkas, dan mudah dipahami.

Dalam praktikum ini, mahasiswa mempelajari berbagai contoh penerapan perulangan menggunakan struktur *for* dan *nested loop* (perulangan bersarang). Melalui tujuh program yang berbeda, mahasiswa diajak untuk memahami bagaimana perulangan digunakan untuk menampilkan data berurutan, melakukan operasi aritmetika, serta mencetak pola seperti segitiga dan persegi dari simbol tertentu. Praktikum ini juga memperkenalkan penggunaan kelas *Scanner* untuk mengambil input dari pengguna, sehingga program menjadi lebih interaktif dan dinamis. Dengan praktikum ini, mahasiswa dapat menguasai logika dasar dalam membangun algoritma menggunakan perulangan.

Selain itu, pemahaman tentang perulangan juga menjadi pondasi penting dalam pengembangan program yang lebih kompleks di masa depan. Struktur perulangan tidak hanya digunakan untuk menampilkan data sederhana, tetapi juga berperan penting dalam pembuatan algoritma seperti pencarian dan pengurutan. Oleh karena itu, praktikum ini bertujuan untuk membangun kemampuan berpikir logis dan analitis mahasiswa dalam memahami cara kerja komputer menjalankan instruksi berulang secara efisien dan terstruktur.

1.2 Tujuan Praktikum

Tujuan dilakukannya praktikum ini adalah sebagai berikut :

1. Memahami konsep dasar perulangan *for* dalam bahasa pemrograman Java.
2. Mengimplementasikan perulangan untuk menampilkan data secara berurutan dan melakukan perhitungan otomatis.
3. Mempelajari cara kerja *nested loop* untuk membentuk pola atau tampilan tertentu di layar.
4. Mengembangkan kemampuan logika pemrograman melalui penyusunan algoritma berbasis perulangan.

1.3 Manfaat Praktikum

Manfaat dilakukannya praktikum ini adalah sebagai berikut :

1. Mahasiswa memahami fungsi dan cara kerja struktur perulangan dalam pemrograman Java.
2. Mahasiswa mampu menulis program yang lebih efisien dengan menggunakan konsep pengulangan.
3. Mahasiswa dapat membedakan penggunaan *System.out.print* dan *System.out.println* untuk mengatur tampilan output.
4. Mahasiswa memperoleh pengalaman menerapkan logika berpikir sistematis dalam menyelesaikan permasalahan berulang.

BAB II

PEMBAHASAN

2.1 Kode Program

2.1.1 PerulanganFor1

```

1  package Pekan5;
2
3  public class PerulanganFor1 {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7          for (int i = 1; i <= 10; i++) {
8              System.out.println(i);
9          }
10     }
11
12 }
```

Kode Program 2.1 (Kode Program PerulanganFor1)

Uraian Kode Program :

1. *package Pekan5;* → Menunjukkan bahwa program ini berada dalam paket bernama *Pekan5*.
2. *public class PerulanganFor1* → Mendefinisikan kelas utama bernama *PerulanganFor1*.
3. *public static void main(String[] args)* → Merupakan metode utama yang dijalankan pertama kali ketika program dieksekusi.
4. *for (int i = 1; i <= 10; i++)* → Membuat perulangan (*looping*) sebanyak 10 kali, dimulai dari nilai *i = 1* hingga *i = 10*.
5. *System.out.println(i);* → Menampilkan nilai *i* pada setiap iterasi ke layar (output).

2.1.2 PerulanganFor2

```

1 package Pekan5;
2
3 public class PerulanganFor2 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int i = 1; i <= 10; i++) {
8             System.out.print(i+" ");
9         }
10    }
11
12 }

```

Kode Program 2.2 (Kode Program PerulanganFor2)

Uraian Kode Program :

1. *package Pekan5;* → Menunjukkan bahwa program ini berada dalam paket bernama *Pekan5*.
2. *public class PerulanganFor2* → Mendefinisikan kelas utama bernama *PerulanganFor2*.
3. *public static void main(String[] args)* → Merupakan metode utama yang dijalankan pertama kali saat program dijalankan.
4. *for (int i = 1; i <= 10; i++)* → Membuat perulangan sebanyak 10 kali, dimulai dari nilai *i = 1* hingga *i = 10*.
5. *System.out.print(i + " ");* → Menampilkan nilai *i* pada setiap iterasi di satu baris, dengan spasi di antara angka.

2.1.3 PerulanganFor3

```

1 package Pekan5;
2
3 public class PerulanganFor3 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int jumlah=0;
8         for (int i=1;i<=10;i++) {
9             System.out.print(i);
10            jumlah= jumlah+i;
11            if (i<10) {
12                System.out.print(" + ");
13            }
14        }
15        System.out.println();
16        System.out.println("Jumlah = "+jumlah);
17    }
18
19 }

```

Kode Program 2.3 (Kode Program PerulanganFor3)

Uraian Kode Program :

1. *package Pekan5;* → Menunjukkan bahwa program berada dalam paket bernama *Pekan5*.
2. *public class PerulanganFor3* → Mendefinisikan kelas utama bernama *PerulanganFor3*.
3. *int jumlah = 0;* → Mendeklarasikan variabel untuk menyimpan total penjumlahan angka.
4. *for (int i = 1; i <= 10; i++)* → Melakukan perulangan dari angka 1 sampai 10.
5. *System.out.print(i);* → Menampilkan nilai *i* tanpa pindah baris.
6. *jumlah = jumlah + i;* → Menambahkan nilai *i* ke dalam variabel *jumlah*.
7. *if (i < 10)* → Menambahkan spasi setelah setiap angka kecuali angka terakhir.
8. *System.out.println();* → Pindah ke baris baru setelah perulangan selesai.
9. *System.out.println("Jumlah = " + jumlah);* → Menampilkan hasil total penjumlahan semua angka.

2.1.4 PerulanganFor4

```

1 package Pekan5;
2 import java.util.Scanner;
3 public class PerulanganFor4 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int jumlah=0;
8         int batas;
9         Scanner input= new Scanner (System.in);
10        System.out.print("Masukkan nilai batas = ");
11        batas= input.nextInt();
12        input.close();
13        for (int i=1;i<=batas;i++) {
14            System.out.print(i);
15            jumlah= jumlah+i;
16            if (i<batas) {
17                System.out.print(" + ");
18            } else {
19                System.out.print(" = ");
20            }
21        }
22        System.out.println(jumlah);
23    }
24
25 }

```

Kode Program 2.4 (Kode Program PerulanganFor4)

Uraian Kode Program :

1. *package Pekan5;* → Menunjukkan bahwa program berada dalam paket bernama *Pekan5*.
2. *import java.util.Scanner;* → Mengimpor kelas *Scanner* agar program dapat menerima input dari pengguna.
3. *int jumlah = 0;* dan *int batas;* → Mendeklarasikan variabel untuk menyimpan total penjumlahan dan batas perulangan.
4. *Scanner input = new Scanner(System.in);* → Membuat objek *Scanner* untuk membaca input dari keyboard.
5. *System.out.print("Masukkan nilai batas = ");* → Menampilkan pesan agar pengguna memasukkan nilai batas.
6. *batas = input.nextInt();* → Menyimpan nilai batas yang dimasukkan oleh pengguna.
7. *for (int i = 1; i <= batas; i++)* → Melakukan perulangan dari angka 1 sampai nilai batas yang diinputkan.

8. `System.out.print(i);` → Menampilkan angka setiap kali perulangan.
9. `jumlah = jumlah + i;` → Menambahkan nilai `i` ke variabel `jumlah` untuk menghitung total.
10. `if (i < batas)` → Menambahkan tanda “+” di antara angka, dan jika sudah mencapai batas, diganti dengan tanda “=”.
11. `System.out.println(jumlah);` → Menampilkan hasil total penjumlahan semua angka.

2.1.5 nestedFor0

```

1 package Pekan5;
2
3 public class nestedFor0 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int line = 1; line <= 5; line++) {
8             for (int j = 1; j <= (-1 * line + 5); j++) {
9                 System.out.print(".");
10            }
11            System.out.print(line);
12            System.out.println();
13        }
14    }
15 }
16 }

```

Kode Program 2.5 (Kode Program nestedFor0)

Uraian Kode Program :

1. `package Pekan5;` → Menunjukkan bahwa program berada dalam paket bernama `Pekan5`.
2. `public class nestedFor0` → Mendefinisikan kelas utama bernama `nestedFor0`.
3. `public static void main(String[] args)` → Merupakan metode utama yang dijalankan pertama kali saat program dijalankan.
4. `for (int line = 1; line <= 5; line++)` → Merupakan perulangan luar (*outer loop*) yang mengatur jumlah baris dari 1 sampai 5.

5. `for (int j = 1; j <= (-1 * line + 5); j++)` → Merupakan perulangan dalam (*inner loop*) yang mencetak titik (.) dengan jumlah yang berkurang setiap barisnya.
6. `System.out.print(line);` → Setelah mencetak titik, program menampilkan angka baris saat ini.
7. `System.out.println();` → Pindah ke baris berikutnya setelah satu baris selesai dicetak.

2.1.6 nestedFor1

```

1 package Pekan5;
2
3 public class nestedFor1 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int i = 1; i <= 5; i++) {
8             for (int j = 1; j <=5; j++) {
9                 System.out.print("*");
10            }
11            System.out.println();
12            // to end the line
13        }
14    }
15
16 }

```

Kode Program 2.6 (Kode Program nestedFor1)

Uraian Kode Program :

1. `package Pekan5;` → Menunjukkan bahwa program berada dalam paket bernama *Pekan5*.
2. `public class nestedFor1` → Mendefinisikan kelas utama bernama *nestedFor1*.
3. `public static void main(String[] args)` → Merupakan metode utama yang dijalankan pertama kali saat program dijalankan.
4. `for (int i = 1; i <= 5; i++)` → Merupakan perulangan luar (*outer loop*) yang mengatur jumlah baris, dari 1 hingga 5.

5. `for (int j = 1; j <= 5; j++)` → Merupakan perulangan dalam (*inner loop*) yang mencetak simbol bintang (*) sebanyak 5 kali di setiap baris.
6. `System.out.print("*");` → Menampilkan simbol bintang tanpa pindah baris.
7. `System.out.println();` → Digunakan untuk pindah ke baris baru setelah satu baris penuh bintang selesai dicetak.

2.1.7 nestedFor2

```

1  package Pekan5;
2
3  public class nestedFor2 {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7          for (int i = 0; i <= 5; i++) {
8              for (int j = 0; j <=5; j++) {
9                  System.out.print(i+j+ " ");
10             }
11             System.out.println();
12             // to end the line
13         }
14     }
15 }
16 }

```

Kode Program 2.7 (Kode Program nestedFor2)

Uraian Kode Program :

1. `package Pekan5;` → Menunjukkan bahwa program berada dalam paket bernama *Pekan5*.
2. `public class nestedFor2` → Mendefinisikan kelas utama bernama *nestedFor2*.
3. `public static void main(String[] args)` → Metode utama yang dijalankan pertama kali saat program dijalankan.
4. `for (int i = 0; i <= 5; i++)` → Merupakan perulangan luar (*outer loop*) yang mengatur jumlah baris dari 0 hingga 5.

5. `for (int j = 0; j <= 5; j++)` → Merupakan perulangan dalam (*inner loop*) yang mencetak nilai hasil penjumlahan antara i dan j.
6. `System.out.print(i + j + " ");` → Menampilkan hasil penjumlahan i + j di setiap kolom, dipisahkan dengan spasi.
7. `System.out.println();` → Berfungsi untuk pindah ke baris baru setelah satu baris selesai dicetak.

2.2 Langkah Kerja

2.2.1 PerulanganFor1

1. Membuat kelas baru bernama *PerulanganFor1* dalam *package Pekan5*.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass:

Interfaces:

Which method stubs would you like to create?

☒ `public static void main(String[] args)`
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.1 (Langkah 1 PerulanganFor1)

2. Program dijalankan dan masuk ke *main*.

```
public static void main(String[] args) {
```

Gambar 2.2 (Langkah 2 PerulanganFor1)

3. Inisialisasi *loop for* dengan $i = 1$. Cek kondisi $i \leq 10$. Jika benar, lanjut; jika salah, keluar *loop*. Jalankan `System.out.println(i);` → cetak i lalu pindah baris. *Increment* i ($i++$) dan ulangi langkah cek sampai $i > 10$.

```
for (int i = 1; i <= 10; i++) {
    System.out.println(i);
}
```

Gambar 2.3 (Langkah 3 PerulanganFor1)

2.2.2 PerulanganFor2

1. Membuat kelas baru bernama *PerulanganFor2* dalam *package Pekan5*.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass:

Interfaces:

Which method stubs would you like to create?

☒ public static void main(String[] args)

☐ Constructors from superclass

☐ Inherited abstract methods

Gambar 2.4 (Langkah 1 PerulanganFor2)

2. Program dijalankan dan masuk ke *main*.

```
public static void main(String[] args) {
```

Gambar 2.5 (Langkah 2 PerulanganFor2)

3. Program melakukan perulangan menggunakan struktur *for* dengan variabel *i* yang dimulai dari 1 sampai 10. Pada setiap iterasi, perintah `System.out.print(i + " ");` menampilkan nilai *i* diikuti dengan spasi tanpa pindah ke baris baru. Karena menggunakan *print* bukan *println*, seluruh angka akan muncul di satu baris yang sama.

```
for (int i = 1; i <= 10; i++) {
    System.out.print(i + " ");
}
```

Gambar 2.6 (Langkah 3 PerulanganFor2)

2.2.3 PerulanganFor3

1. Membuat kelas baru bernama *PerulanganFor3* dalam package *Pekan5*.

Source folder: Prakalpro_2025_B_2511531016/src Browse...

Package: Pekan5 Browse...

☐ Enclosing type: Browse...

Name: PerulanganFor3

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.7 (Langkah 1 PerulanganFor3)

2. Program dijalankan dan masuk ke *main*.

```
public static void main(String[] args) {
```

Gambar 2.8 (Langkah 2 PerulanganFor3)

3. Buat variabel `jumlah = 0`, mulai *for* dari `i=1` sampai `I<=10`, cetak `I` tanpa *newline* (`System.out.print(I)`), tambahkan `I` ke `jumlah` (`jumlah = jumlah + 1`). Jika `I<10` cetak `" + "`; jika `I==10` tidak mencetak `+`.

```
int jumlah=0;
for (int i=1;i<=10;i++) {
    System.out.print(i);
    jumlah= jumlah+i;
    if (i<10) {
        System.out.print(" + ");
    }
}
```

Gambar 2.9 (Langkah 3 PerulanganFor3)

4. Setelah *loop*, `System.out.println();` untuk pindah baris. Cetak `System.out.println(jumlah = " + jumlah);` untuk menampilkan total.

```
System.out.println();
System.out.println("Jumlah = "+jumlah);
```

Gambar 2.10 (Langkah 4 PerulanganFor3)

2.2.4 PerulanganFor4

1. Membuat kelas baru bernama *PerulanganFor4* dalam *package Pekan5*.

Source folder: Browse...

Package: Browse...

☐ Enclosing type: Browse...

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.11 (Langkah 1 PerulanganFor4)

2. Program dijalankan dan masuk ke *main*.

```
public static void main(String[] args) {
```

Gambar 2.12 (Langkah 2 PerulanganFor4)

3. Inisialisasi jumlah=0 dan deklarasi batas. Buat Scanner input dan tampilkan prompt Masukkan nilai batas =. Lalu, baca integer batas = input.nextInt() lalu tutup scanner.

```
int jumlah=0;
int batas;
Scanner input= new Scanner (System.in);
System.out.print("Masukkan nilai batas = ");
batas= input.nextInt();
input.close();
```

Gambar 2.13 (Langkah 3 PerulanganFor4)

4. Program melakukan perulangan *for* dari angka 1 sampai batas yang dimasukkan. Pada setiap iterasi, program mencetak angka dan menambahkannya ke variabel jumlah. Jika angka belum mencapai batas, akan ditampilkan tanda "+", namun jika sudah mencapai batas terakhir, akan diganti dengan tanda "=" . Terakhir, program menampilkan total hasil penjumlahan dari seluruh angka yang telah dicetak.

```
for (int i=1;i<=batas;i++) {
    System.out.print(i);
    jumlah= jumlah+i;
    if (i<batas) {
        System.out.print(" + ");
    } else {
        System.out.print(" = ");
    }
}
System.out.println(jumlah);
```

Gambar 2.14 (Langkah 4 PerulanganFor4)

2.2.5 nestedFor0

1. Membuat kelas baru bernama *nestedFor0* dalam *package Pekan5*.

Source folder: Prakalpro_2025_B_2511531016/src Browse...

Package: Pekan5 Browse...

☐ Enclosing type: Browse...

Name: nestedFor0

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.15 (Langkah 1 nestedFor0)

2. Program dijalankan dan masuk ke *main*.

```
public static void main(String[] args) {
```

Gambar 2.16 (Langkah 2 nestedFor0)

3. *Loop* luar *for* (*line*=1; *line* <=5; *line*++) untuk tiap baris. Di setiap baris, *loop* dalam *for* (*j*=1; *j* <= (-1**line*+5); *j*++) dan cetak "." sebanyak nilai ekspresi (-1**line*+5).

```
for (int line = 1; line <= 5; line++) {
    for (int j = 1; j <= (-1 * line + 5); j++) {
        System.out.print(".");
    }
}
```

Gambar 2.17 (Langkah 3 nestedFor0)

4. Setelah mencetak titik-titik, program menampilkan nilai line di akhir setiap baris. Kemudian program berpindah ke baris berikutnya hingga semua lima baris selesai dicetak, menghasilkan pola menurun dari titik yang diakhiri angka.

```
System.out.print(line);
System.out.println();
```

Gambar 2.18 (Langkah 4 nestedFor0)

2.2.6 nestedFor1

1. Membuat kelas baru bernama *nestedFor1* dalam package *Pekan5*.

Gambar 2.19 (Langkah 1 nestedFor1)

2. Program dijalankan dan masuk ke *main*.

```
public static void main(String[] args) {
```

Gambar 2.20 (Langkah 2 nestedFor1)

3. *Loop* luar *for* (*i*=1; *line* <=5; *i*++) untuk mengatur baris menjadi 5 baris. Untuk setiap baris, *loop* dalam *for* (*j*=1; *j*<=5; *j*++), cetak * tanpa *newline*.

```
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <=5; j++) {
        System.out.print("*");
    }
}
```

Gambar 2.21 (Langkah 3 nestedFor1)

4. Setiap kali inner loop selesai, perintah *System.out.println()*; dijalankan agar kursor berpindah ke baris berikutnya. Proses ini terus diulang hingga terbentuk pola persegi bintang berukuran 5x5.

```
System.out.println();
// to end the line
```

Gambar 2.22 (Langkah 4 nestedFor1)

2.2.7 nestedFor2

1. Membuat kelas baru bernama *nestedFor2* dalam *package Pekan5*.

Source folder: Browse...

Package: Browse...

☐ Enclosing type: Browse...

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.23 (Langkah 1 nestedFor2)

2. Program dijalankan dan masuk ke *main*.

```
public static void main(String[] args) {
```

Gambar 2.24 (Langkah 2 nestedFor2)

3. Program menghitung hasil penjumlahan antara *i* dan *j*, lalu mencetak hasilnya diikuti spasi. Setelah satu baris selesai dicetak, program berpindah ke baris baru menggunakan *System.out.println()*. Proses ini menghasilkan tabel penjumlahan dua angka dari 0 hingga 5, di mana setiap baris menampilkan hasil penjumlahan nilai *i* dengan semua nilai *j*.

```
for (int i = 0; i <= 5; i++) {
    for (int j = 0; j <= 5; j++) {
        System.out.print(i+j+ " ");
    }
    System.out.println();
    // to end the line
```

Gambar 2.25 (Langkah 3 nestedFor2)

2.3 Analisis Hasil

Berdasarkan hasil percobaan dari ketujuh kode program yang telah dijalankan, seluruh program menunjukkan penerapan teori dasar tentang struktur kendali perulangan (*looping control structure*) dalam bahasa pemrograman Java. Perulangan digunakan untuk mengeksekusi suatu blok perintah secara berulang hingga kondisi tertentu tercapai.

Pada program pertama hingga keempat, konsep perulangan tunggal (*single loop*) digunakan untuk menampilkan angka dan melakukan operasi penjumlahan

otomatis. Program ini mencerminkan teori dari Schildt (2018) yang menjelaskan bahwa struktur *for* merupakan bentuk perulangan yang efisien karena memiliki inisialisasi, kondisi, dan iterasi dalam satu baris, sehingga mudah dikontrol dan dibaca.

Selanjutnya, program kelima hingga ketujuh menunjukkan penerapan perulangan bersarang (*nested loop*). Menurut Savitch (2014), *nested loop* memungkinkan pengulangan di dalam pengulangan lain untuk menghasilkan pola atau struktur data yang lebih kompleks. Hal ini terlihat pada hasil keluaran berupa pola titik, bintang, dan tabel angka, di mana setiap baris dikontrol oleh perulangan luar (*i*), sedangkan kolom dikontrol oleh perulangan dalam (*j*). Selain itu, penggunaan *System.out.print* dan *System.out.println* memperlihatkan perbedaan dalam pencetakan hasil di layar, di mana *print* menampilkan data pada baris yang sama, sedangkan *println* berpindah ke baris berikutnya.

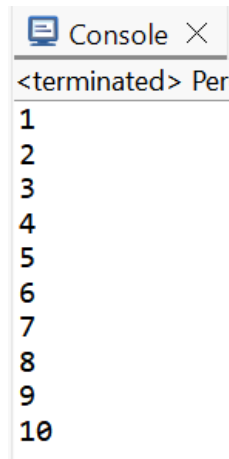
Secara teoritis, perulangan juga menjadi bagian penting dalam algoritma efisiensi dan otomatisasi proses komputasi, sebagaimana dijelaskan oleh Sebesta (2016) bahwa *looping* merupakan dasar untuk pengulangan instruksi yang berkontribusi terhadap penghematan waktu eksekusi dan jumlah baris kode.

Berdasarkan implementasi ketujuh program tersebut, terlihat bahwa struktur perulangan dapat digunakan untuk berbagai tujuan, mulai dari menampilkan data sederhana, menghitung nilai aritmatika, hingga membentuk pola atau tabel. Oleh karena itu, pemahaman terhadap konsep dasar perulangan sangat penting bagi programmer dalam membangun program yang efisien, fleksibel, dan terstruktur.

BAB III KESIMPULAN

3.1 Hasil Praktikum

3.1.1 PerulanganFor1

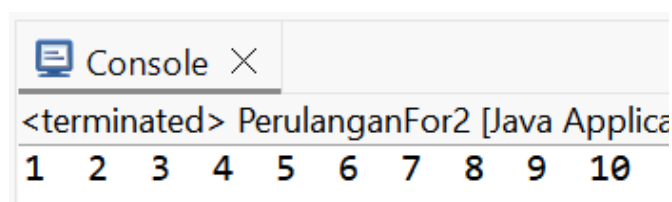


```
<terminated> Per
1
2
3
4
5
6
7
8
9
10
```

Gambar 3.1 (Hasil Praktikum PerulanganFor1)

Output menampilkan angka dari 1 sampai 10 yang muncul secara berurutan ke bawah di konsol. Ini berarti programnya dibuat untuk menampilkan deretan angka tersebut satu per satu. Program menggunakan perulangan *for* agar angka bisa muncul berturut-turut tanpa perlu diketik manual. Setiap angka dicetak di baris baru karena menggunakan perintah *println*, sehingga hasilnya terlihat rapi dan mudah dibaca serta menampilkan hasil sesuai logika yang telah ditentukan.

3.1.2 PerulanganFor2

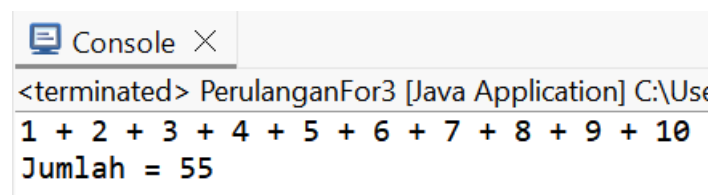


```
<terminated> PerulanganFor2 [Java Applica
1 2 3 4 5 6 7 8 9 10
```

Gambar 3.2 (Hasil Praktikum PerulanganFor2)

Output menampilkan angka dari 1 sampai 10 yang muncul secara berurutan ke samping di konsol. Ini berarti programnya dibuat untuk menampilkan deretan angka tersebut satu per satu. Program menggunakan perulangan *for* agar angka bisa muncul berturut-turut tanpa perlu diketik manual. Setiap angka dicetak berurutan dalam satu baris, sehingga hasilnya terlihat sesuai kondisi dan logika yang telah ditentukan.

3.1.3 PerulanganFor3

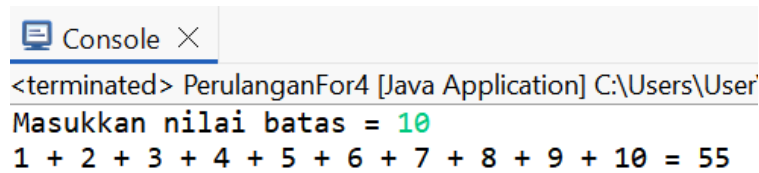


```
<terminated> PerulanganFor3 [Java Application] C:\Use
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah = 55
```

Gambar 3.3 (Hasil Praktikum PerulanganFor3)

Output di konsol ini menampilkan proses dan hasil dari operasi penjumlahan deret angka. Baris pertama, yaitu $1 + 2 + 3 + \dots$ sampai $+ 10$, berfungsi untuk menampilkan operasi yang sedang dilakukan, yaitu menjumlahkan semua angka berurutan dari 1 hingga 10. Baris kedua menunjukkan hasil akhir dari perhitungan tersebut, yaitu Jumlah = 55. Ini berarti program menggunakan perulangan *'for'* untuk menjumlahkan deret bilangan secara otomatis, dan hasil akhirnya yang berupa total keseluruhan (55) dicetak setelah proses perulangan selesai.

3.1.4 PerulanganFor4



```

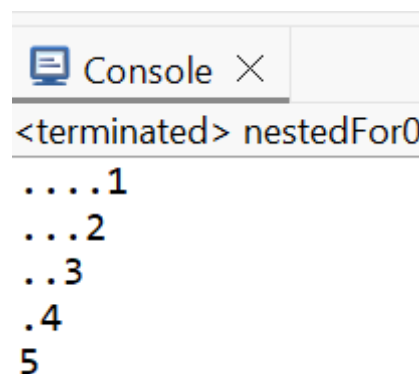
Console X
<terminated> PerulanganFor4 [Java Application] C:\Users\User
Masukkan nilai batas = 10
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55

```

Gambar 3.4 (Hasil Praktikum PerulanganFor4)

Output di atas menunjukkan hasil dari sebuah program Java yang meminta pengguna memasukkan nilai batas, yaitu angka 10. Program kemudian menampilkan deret penjumlahan dari angka 1 sampai 10, yaitu $1 + 2 + 3 + \dots + 10$, dan menghitung totalnya menjadi 55. Dengan kata lain, program ini menghitung jumlah semua bilangan bulat dari 1 hingga nilai batas yang dimasukkan pengguna dan menampilkan proses perhitungannya secara terperinci sehingga mudah dipahami.

3.1.5 nestedFor0



```

Console X
<terminated> nestedFor0
....1
...2
..3
.4
5

```

Gambar 3.5 (Hasil Praktikum nestedFor0)

Program diatas mencetak angka dari 1 sampai 5, di mana setiap angka didahului oleh sejumlah tanda titik (.) yang jumlahnya berkurang

satu setiap baris. Pola ini menunjukkan bahwa semakin besar angkanya, semakin sedikit jumlah tanda titik di depannya, sehingga tampilan yang dihasilkan membentuk baris yang unik dan teratur.

3.1.6 nestedFor1

```

Console X
<terminated> nestedFor1
*****
*****
*****
*****
*****

```

Gambar 3.6 (Hasil Praktikum nestedFor1)

Output di atas berasal dari program yang menggunakan perulangan bertingkat untuk mencetak pola bintang. Setiap baris terdiri dari lima tanda bintang (*) dan jumlah barisnya juga ada 5, sehingga membentuk persegi atau kotak bintang yang rapi. Pola ini biasa digunakan untuk latihan dasar pemrograman untuk memahami konsep *nested loop* atau perulangan di dalam perulangan.

3.1.5 nestedFor2

```

Console X
<terminated> nestedFor2
0 1 2 3 4 5
1 2 3 4 5 6
2 3 4 5 6 7
3 4 5 6 7 8
4 5 6 7 8 9
5 6 7 8 9 10

```

Gambar 3.7 (Hasil Praktikum nestedFor2)

Output konsol ini menampilkan sebuah tabel angka berukuran 6x6 yang dihasilkan oleh sebuah program menggunakan struktur perulangan bersarang (*nested loops*). Pola angka tersebut sangat terstruktur, setiap angka di dalam tabel merupakan hasil penjumlahan antara indeks baris dan indeks kolom. Setiap baris adalah urutan angka yang dimulai satu angka lebih besar dari baris sebelumnya, dan setiap kolom menunjukkan angka yang bertambah secara seragam.

3.2 Saran Pengembangan

Berdasarkan hasil analisis dari ketujuh kode program perulangan yang telah dijalankan, terdapat beberapa saran pengembangan agar program menjadi lebih interaktif, efisien, dan bermanfaat dalam penerapan nyata. Pada program pertama hingga keempat, yang berfokus pada perulangan tunggal, program dapat dikembangkan dengan menambahkan validasi input, agar pengguna tidak memasukkan nilai yang tidak sesuai (misalnya angka negatif atau huruf).

Selain itu, hasil keluaran dapat dibuat lebih informatif, seperti menampilkan langkah-langkah perhitungan penjumlahan secara otomatis atau menambahkan tampilan berbasis grafik sederhana. Program keempat yang menggunakan input pengguna juga bisa diperluas dengan menambahkan fitur perhitungan rata-rata atau faktorial, sehingga pengguna dapat memilih jenis operasi yang diinginkan, bukan hanya penjumlahan saja.

Untuk program kelima hingga ketujuh, yang menerapkan perulangan bersarang, pengembangan dapat dilakukan dengan menambahkan opsi bentuk pola yang berbeda sesuai pilihan pengguna, misalnya segitiga, piramida, atau pola terbalik. Program keenam (pola bintang) bisa dikembangkan menjadi program pembentuk pola dinamis di mana pengguna dapat menentukan ukuran pola

melalui input. Sementara program ketujuh (tabel penjumlahan) dapat dikembangkan menjadi tabel perkalian atau tabel kombinasi matematika agar lebih bervariasi dan edukatif.

Secara keseluruhan, pengembangan terhadap program-program perulangan ini bertujuan agar tidak hanya berfungsi sebagai latihan dasar pemrograman, tetapi juga memiliki nilai praktis yang lebih tinggi, mudah digunakan oleh pengguna, dan memperlihatkan kemampuan logika serta kreativitas dalam membangun algoritma yang efisien dan adaptif.

DAFTAR PUSTAKA

- [1] J. Lewis dan W. Loftus, *Java Software Solutions: Foundations of Program Design*, 9th ed., Pearson, 2018. [Online]. Available: <https://www.pearson.com>
- [2] H. Schildt, *Java: The Complete Reference*, 11th ed., McGraw-Hill Education, 2018. [Online]. Available: <https://www.mheducation.com>
- [3] W. Savitch, *Absolute Java*, 6th ed., Pearson, 2014. [Online]. Available: <https://www.pearson.com>
- [4] R. W. Sebesta, *Concepts of Programming Languages*, 12th ed., Pearson, 2016. [Online]. Available: <https://www.pearson.com>