

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
LAPORAN PRAKTIKUM PEKAN 5 (PERULANGAN)

Disusun Oleh:

Syasya Halwa Gazwani

(2511531018)

Dosen Pengampu:

Dr. Wahyudi, S.T, M.T

Asisten Praktikum:

Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2025

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh,

Segala puji bagi Allah SWT yang telah memberikan kesempatan serta kemudahan kepada penulis untuk dapat menyelesaikan Laporan Praktikum pada mata kuliah Algoritma Pemrograman, sehingga Laporan Praktikum ini dapat dikumpulkan dengan tepat waktu. Atas rahmat dan karunianya Laporan Praktikum dapat terselesaikan dengan baik. Sholawat serta salam juga penulis sampaikan kepada baginda tercinta yaitu Nabi Muhammad SAW. yang kita nantikan syafa'atnya di akhirat nanti. Laporan Praktikum ini bertujuan untuk menambah wawasan para pembaca untuk lebih memperdalam ilmu yang ada pada makalah ini.

Dalam penyusunan Laporan Praktikum ini, penulis mengalami banyak kesulitan dan penulis menyadari bahwa Laporan Praktikum ini masih jauh dari kesempurnaan. Untuk itu, penulis sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan pada Laporan Praktikum ini. Penulis mengucapkan terima kasih kepada bapak Dr. Wahyudi, S.T, M.T. selaku dosen mata kuliah Algoritma Pemrograman yang telah memberikan tugas ini sehingga dapat menambah pengetahuan dan wawasan sesuai dengan mata kuliah yang penulis tekuni. Penulis juga mengucapkan terima kasih kepada semua pihak yang telah membagi sebagian pengetahuannya sehingga penulis dapat menyelesaikan Laporan Praktikum ini.

Padang, September 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	i
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	1
BAB II PEMBAHASAN	3
2.1 PerulanganFor1	3
2.1.1 Kode Program	3
2.1.2 Langkah Kerja	3
2.1.3 Analisis Hasil	3
2.2 PerulanganFor2	4
2.2.1 Kode Program	4
2.2.2 Langkah Kerja	4
2.2.3 Analisis Hasil	4
2.3 PerulanganFor3	5
2.3.1 Kode Program	5
2.3.2 Langkah Kerja	5
2.3.3 Analisis Hasil	6
2.4 PerulanganFor4	7
2.4.1 Kode Program	7
2.4.2 Langkah Kerja	7
2.4.3 Analisis Hasil	8
2.5 NestedFor0	8
2.5.1 Kode Program	8
2.5.2 Langkah Kerja	9
2.5.3 Analisis Hasil	9
2.6 NestedFor1	10
2.6.1 Kode Program	10
2.6.2 Langkah Kerja	10
2.6.3 Analisis Hasil	10

2.7	NestedFor2	11
2.7.1	Kode Program	11
2.7.2	Langkah Kerja	11
2.7.3	Analisis Hasil	12
KESIMPULAN.....		13
DAFTAR PUSTAKA.....		14

BAB I PENDAHULUAN

1.1 Latar Belakang

Pemrograman atau *programming* merupakan sebuah proses menulis, menguji dan memperbaiki (*debug*), serta memelihara kode yang dapat membangun suatu program komputer. Kode ini ditulis dalam berbagai bahasa pemrograman atau sering disebut juga bahasa komputer. Tujuan dari pemrograman adalah untuk memuat suatu program yang dapat melakukan suatu perhitungan atau “pekerjaan” sesuai dengan keinginan pemrograman. Untuk melakukan pemrograman, diperlukan keterampilan dalam algoritma, logika, bahasa pemrograman, dan pada banyak kasus, pengetahuan-pengetahuan lain seperti matematika. Bahasa Pemrograman, atau biasanya disebut bahasa komputer atau *computer language programming*, yaitu sebuah instruksi standar untuk mengendalikan sebuah komputer. Seperangkat aturan *syntax* dan *semantic* yang digunakan untuk memberikan sebuah definisi pada program komputer dikenal sebagai bahasa pemrograman. Dengan menggunakan bahasa ini, seorang *programmer* dapat dengan tepat menentukan data yang akan diproses oleh komputer, bagaimana penyimpanan dan transfernya, dan tindakan apa yang harus diambil dalam berbagai keadaan.

1.2 Tujuan

- 1.2.1 Memahami konsep dasar tipe data dalam bahasa pemrograman Java serta penerapannya dalam pembuatan program sederhana.
- 1.2.2 Mengembangkan keterampilan analisis terhadap hasil keluaran program sesuai dengan logika yang diterapkan.

1.3 Manfaat Praktikum

- 1.3.1 Menambah wawasan dan keterampilan mahasiswa dalam mengelola data menggunakan bahasa pemrograman Java sebagai dasar untuk pemrograman yang lebih lanjut.

- 1.3.2 Memberikan pemahaman dasar kepada mahasiswa mengenai penggunaan tipe data dalam bahasa pemrograman Java, sehingga dapat menjadi bekal dalam mengembangkan program yang lebih kompleks di tahap berikutnya.

BAB II PEMBAHASAN

2.1 PerulanganFor1

2.1.1 Kode Program

```

1 package pekan5;
2
3 public class PerulanganFor1 {
4
5     public static void main(String[] args) {
6         for (int i = 1; i <= 10; i++) {
7             System.out.println(i);
8         }
9     }
10
11 }

```

Gambar 2.1 Kode Program PerulanganFor1.

2.1.2 Langkah Kerja

2.1.2.1 Menyatakan bahwa *class* PerulanganFor1 berada dalam *package* pekan 5.

```
1 package pekan5;
```

Gambar 2.1.2.1 *Package* pekan 5.

2.1.2.2 Mendefinisikan *class* utama bernama PerulanganFor1.

```
3 public class PerulanganFor1 {
```

Gambar 2.1.2.2 *public class* PerulanganFor1.

2.1.2.3 Metode utama yaitu titik awal eksekusi program Java, perulangan *for* (*for loop*), dan menampilkan hasil *output*.

```

5     public static void main(String[] args) {
6         for (int i = 1; i <= 10; i++) {
7             System.out.println(i);

```

Gambar 2.1.2.3 *Method main, for loop, dan output*.

2.1.3 Analisis Hasil

```

1
2
3
4
5
6
7
8
9
10

```

Gambar 2.1.3 Hasil *output* kode program PerulanganFor1.

Hasil *output* dari program tersebut adalah deretan angka dari 1 sampai 10 yang ditampilkan secara berurutan ke bawah pada layar. Program menggunakan perulangan *for* yang dimulai dari nilai $i = 1$ dan terus berjalan selama nilai i kurang dari atau sama dengan 10. Setiap kali perulangan dijalankan, nilai i akan bertambah satu dan ditampilkan. Karena perintah `println` mencetak hasil di baris baru, maka setiap angka muncul di baris yang terpisah.

2.2 PerulanganFor2

2.2.1 Kode Program

```
1 package pekan5;
2
3 public class perulanganfor2 {
4     public static void main(String[] args) {
5         for (int i = 1; i <= 10; i++) {
6             System.out.print(i+" ");
7         }
8     }
9 }
10 }
```

Gambar 2.2.1 Kode program PerulanganFor2.

2.2.2 Langkah Kerja

2.2.2.1 Program dimulai dari *method main*.

2.2.2.2 Inisialisasi variabel perulangan dan menghasilkan *output*.

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

Gambar 2.2.2.2 Inisialisasi dan *output*.

2.2.3 Analisis Hasil

```
1 2 3 4 5 6 7 8 9 10
```

Gambar 2.2.3 Hasil *output* kode program PengulanganFor2.

Hasil *output* dari program tersebut adalah deretan angka dari 1 sampai 10 yang ditampilkan secara berurutan dalam satu baris dengan spasi di antara setiap angka. Program menggunakan perulangan *for* yang dimulai dari nilai $i = 1$ hingga $i = 10$. Setiap kali perulangan dijalankan, nilai i dicetak ke layar. Perintah *print* di sini tidak membuat baris baru setelah menampilkan angka, sehingga semua angka ditampilkan dalam satu baris saja. Tanda " " setelah i berfungsi untuk memberikan jarak antarangka agar tidak menempel satu sama lain.

2.3 PerulanganFor3

2.3.1 Kode Program

```

1 package pekan5;
2
3 public class perulanganfor3 {
4     public static void main(String[] args) {
5         int jumlah=0;
6         for (int i=1; i<=10; i++) {
7             System.out.print(i);
8             jumlah= jumlah+1;
9             if (i<10) {
10                 System.out.print(" + ");
11             }
12         }
13         System.out.println();
14         System.out.println("Jumlah = "+jumlah);
15     }
16 }

```

Gambar 2.3.1 Kode program PerulanganFor3.

2.3.2 Langkah Kerja

2.3.2.1 Program dimulai dari *method main*.

2.3.2.2 Membuat variabel bertipe *int* dengan nilai awal 0.

```

5         int jumlah=0;

```

Gambar 2.3.2.2 Deklarasi dan inisialisasi variabel jumlah.

2.3.2.3 Masuk ke perulangan *for*, menampilkan nilai i ke layar, menambahkan nilai i ke variabel jumlah, dan menampilkan tanda *plus* (+) di antara angka.

```

6         for (int i=1; i<=10; i++) {
7             System.out.print(i);
8             jumlah= jumlah+1;
9             if (i<10) {
10                 System.out.print(" + ");

```

Gambar 2.3.2.3 Perulangan *for*, *print* nilai *i* ke layar, menampilkan tanda *plus* (+).

2.3.2.4 Membuat baris baru setelah perulangan selesai dan menampilkan hasil akhir penjumlahan.

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

Gambar 2.3.2.4 Baris baru dan hasil akhir.

2.3.3 Analisis Hasil

```
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah = 10
```

Gambar 2.3.3 Hasil *output* kode program PerulanganFor3.

Hasil *output* dari program tersebut menampilkan deretan angka dari 1 sampai 10 yang dipisahkan oleh tanda tambah (+), kemudian di bawahnya menampilkan hasil penjumlahan dari semua angka tersebut. Pada saat program dijalankan, angka 1 hingga 10 dicetak secara berurutan menggunakan perulangan *for*. Setiap kali angka ditampilkan, program juga menambahkan nilainya ke variabel *jumlah* yang berfungsi untuk menghitung total keseluruhan. Tanda “+” hanya muncul di antara angka, bukan setelah angka terakhir, karena ada kondisi *if* ($i < 10$) yang memastikan tanda *plus* tidak dicetak ketika *i* sudah bernilai 10. Setelah seluruh angka ditampilkan, program membuat baris baru dan menuliskan hasil penjumlahannya.

2.4 PerulanganFor4

2.4.1 Kode Program

```

1 package pekan5;
2
3 import java.util.Scanner;
4
5 public class perulanganfor4 {
6     public static void main(String[] args) {
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 }

```

Gambar 2.4.1 Kode program PerulanganFor4.

2.4.2 Langkah Kerja

2.4.2.1 Program dimulai dari *method main*.

2.4.2.2 Mendefinisikan dua variabel, jumlah digunakan untuk menyimpan hasil penjumlahan, dan batas akan diisi dengan angka terakhir yang ditentukan oleh pengguna.

```

7         int jumlah=0;
8         int batas;

```

Gambar 2.4.2.2 Medefinisikan dua variabel.

2.4.2.3 Membuat objek *Scanner* untuk *input*.

```

9         Scanner input= new Scanner(System.in);

```

Gambar 2.4.2.3 Objek *Scanner*.

2.4.2.4 Meminta pengguna memasukkan batas angka dan menutup *Scanner*.

```

10        System.out.print("Masukkan nilai batas = ");
11        batas= input.nextInt();
12        input.close();

```

Gambar 2.4.2.4 Menutup *Scanner*.

2.4.2.5 Masuk perulangan *for*, menampilkan nilai *i*, menambahkan nilai *i* ke variabel jumlah, menampilkan tanda “+” atau “=”, dan menampilkan hasil penjumlahan akhir.

```

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);

```

Gambar 2.4.2.5 Perulangan *for* dan menampilkan hasil.

2.4.3 Analisis Hasil

```

Masukkan nilai batas = 5
1 + 2 + 3 + 4 + 5 = 15

```

Gambar 2.4.3 Hasil *output* kode program PerulanganFor4.

Jika nilai batas yang dimasukkan adalah 5, maka program akan menampilkan hasil berupa deretan angka dari 1 sampai 5 yang dijumlahkan, kemudian diikuti dengan tanda sama dengan dan hasil totalnya. Ketika pengguna mengetik angka 5 setelah pesan “Masukkan nilai batas =”, program akan menjalankan perulangan *for* dari angka 1 hingga 5. Setiap kali angka ditampilkan, program juga menambahkan nilainya ke variabel jumlah. Selain itu, program menampilkan tanda “+” di antara setiap angka, kecuali setelah angka terakhir, di mana tanda “=” akan ditampilkan. Setelah perulangan selesai, program menampilkan nilai total dari hasil penjumlahan semua angka tersebut.

2.5 NestedFor0

2.5.1 Kode Program

```

1 package pekan5;
2
3 public class nestedFor0 {
4     public static void main(String[] args) {
5         for (int line = 1; line <= 5; line++) {
6             for (int j = 1; j <= (-1 * line + 5); j++) {
7                 System.out.print(".");
8             }
9             System.out.print(line);
10            System.out.println();
11        }
12    }
13 }

```

Gambar 2.5.1 Kode program NestedFor0.

2.5.2 Langkah Kerja

2.5.1 Program dimulai dari *method main*.

2.5.2 Perulangan luar mengatur baris (*line*) yang akan dicetak, perulangan dalam digunakan untuk mencetak spasi di depan angka, menampilkan angka *line* setelah spasi, pindah ke baris baru, mengulangi hingga *line*=5, dan mencetak hasil.

```

5     for (int line = 1; line <= 5; line++) {
6         for (int j = 1; j <= (-1 * line + 5); j++) {
7             System.out.print(".");
8         }
9         System.out.print(line);
10        System.out.println();

```

Gambar 2.5.2.2 Perulangan *for* dan mencetak hasil.

2.5.3 Analisis Hasil

```

....1
...2
..3
.4
5

```

Gambar 2.5.3 Hasil *output* kode program NestedFor0.

Hasil *output* dari program tersebut menampilkan angka dari 1 sampai 5 yang tersusun ke bawah dengan posisi rata kanan. Program menggunakan dua perulangan, yaitu perulangan luar untuk mengatur baris dan perulangan dalam untuk mencetak spasi sebelum angka. Pada baris pertama, program mencetak empat spasi lalu angka 1, pada baris kedua mencetak tiga spasi lalu angka 2, dan begitu seterusnya hingga baris terakhir yang tidak memiliki spasi di depannya dan langsung

menampilkan angka 5. Setiap kali angka dicetak, program menambahkan perintah *println()* agar kursor berpindah ke baris baru sebelum mencetak angka berikutnya.

2.6 NestedFor1

2.6.1 Kode Program

```

1 package pekan5;
2
3 public class nestedfor1 {
4     public static void main(String[] args) {
5         for (int i = 1; i <= 5; i++) {
6             for (int j = 1; j <= 5; j++) {
7                 System.out.print("*");
8             }
9             System.out.println();
10            // to end the line
11        }
12    }
13 }
14 }

```

Gambar 2.6.1 Kode Program NestedFor1.

2.6.2 Langkah Kerja

2.6.2.1 Program mulai dieksekusi dari *method main()*.

2.6.2.2 Perulangan luar mengatur jumlah baris yang akan dicetak, untuk setiap iterasi pada *loop* luar, perulangan dalam *for* dijalankan, dan setiap iterasi mencetak satu tanda bintang tanpa berpindah baris.

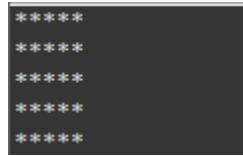
```

5     for (int i = 1; i <= 5; i++) {
6         for (int j = 1; j <= 5; j++) {
7             System.out.print("*");
8         }
9         System.out.println();
10        // to end the line

```

Gambar 2.6.2.2 Perulangan dan mencetak hasil.

2.6.3 Analisis Hasil



Gambar 2.6.3 Hasil output kode program NestedFor1.

Hasil *output* dari program tersebut adalah tampilan berbentuk persegi yang tersusun dari tanda bintang (*). Program menggunakan dua perulangan bersarang (*nested loop*), di mana perulangan luar mengatur jumlah baris, dan perulangan dalam mengatur jumlah bintang pada setiap baris. Pada setiap baris, perulangan dalam mencetak lima tanda bintang secara berurutan tanpa berpindah ke baris baru. Setelah lima bintang selesai dicetak, perintah `System.out.println();` dijalankan untuk membuat baris baru sebelum mencetak baris berikutnya. Proses ini berulang sebanyak lima kali karena perulangan luar berjalan dari $i = 1$ hingga $i = 5$.

2.7 NestedFor2

2.7.1 Kode Program

```

1 package pekan5;
2
3 public class nestedfor2 {
4     public static void main(String[] args) {
5         for (int i = 0; i <= 5; i++) {
6             for (int j = 0; j <= 5; j++) {
7                 System.out.print(i+j+ " ");
8             }
9             System.out.println();
10            // to end the line
11        }
12    }
13 }

```

Gambar 2.7.1 Kode program NestedFor2.

2.7.2 Langkah Kerja

2.7.2.1 Program dimulai dari *method main()*.

2.7.2.2 Variabel i digunakan sebagai penghitung untuk perulangan luar.

```
5 for (int i = 0; i <= 5; i++) {
```

Gambar 2.7.2.2 Perulangan luar.

- 2.7.2.3 Di dalam perulangan luar, terdapat perulangan dalam *for* yang akan dijalankan setiap kali satu iterasi perulangan luar berlangsung.

```
6 for (int j = 0; j <= 5; j++) {
```

Gambar 2.7.2.3 Perulangan dalam.

- 2.7.2.4 Mencetak hasil.

```
7     System.out.print(i+j+ " ");
8 }
9 System.out.println();
```

Gambar 2.7.2.4 Mencetak hasil *output*.

2.7.3 Analisis Hasil

```
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 2.7.3 Hasil kode program NestedFor2.

Hasil *output* dari program tersebut berupa tabel angka berukuran 6 baris dan 6 kolom. Setiap angka yang ditampilkan merupakan hasil penjumlahan antara variabel *i* (dari perulangan luar) dan *j* (dari perulangan dalam).

Perulangan luar (*i*) menentukan baris, sedangkan perulangan dalam (*j*) menentukan kolom pada setiap baris. Pada setiap iterasi, program mencetak hasil dari *i + j* diikuti dengan satu spasi agar angka tidak menempel. Setelah satu baris penuh selesai dicetak (saat *j* mencapai 5), program menulis *println()* agar kursor berpindah ke baris berikutnya dan memulai pencetakan baris baru.

KESIMPULAN

Dari semua program yang telah diberikan dapat dipahami bahwa seluruhnya membahas tentang konsep perulangan (*looping*) dalam bahasa pemrograman Java. Perulangan digunakan agar komputer bisa melakukan suatu proses secara berulang tanpa perlu menulis perintah yang sama berkali-kali.

Pada program awal, perulangan digunakan untuk menampilkan angka dari 1 sampai 10, baik secara vertikal maupun horizontal, sehingga terlihat bagaimana perbedaan penggunaan *println* dan *print* dalam mencetak hasil. Pada program berikutnya, konsep perulangan dikembangkan dengan menambahkan operasi penjumlahan serta penerimaan *input* dari pengguna melalui *Scanner*, sehingga hasil yang ditampilkan bisa menyesuaikan dengan nilai batas yang dimasukkan.

Selanjutnya, pada program yang menggunakan *nested for* atau perulangan bersarang, diperlihatkan bagaimana satu perulangan bisa ditempatkan di dalam perulangan lain untuk menghasilkan pola tertentu, seperti pola bintang atau pola angka yang membentuk kotak dan segitiga.

Secara keseluruhan, semua program tersebut menunjukkan bahwa perulangan adalah salah satu konsep dasar penting dalam pemrograman, karena membantu membuat proses yang berulang menjadi lebih efisien, rapi, dan mudah dikendalikan dengan logika yang jelas.

DAFTAR PUSTAKA

- [1] H. Schildt, *Java: Referensi Lengkap*, Edisi ke-12. New York, AS: McGraw-Hill Education, 2021.
- [2] Oracle, “Pernyataan for (Tutorial Java™ > Mempelajari Bahasa Java > Dasar Bahasa),” Oracle, 2024.
- [3] W. Savitch, *Pemrograman Java Dasar (Absolute Java)*, Edisi ke-7. Boston, AS: Pearson Education, 2019.
- [4] J. Bloch, *Java yang Efektif (Effective Java)*, Edisi ke-3. Boston, AS: Addison-Wesley, 2018.
- [5] Oracle, “Dokumentasi Java Platform, Standard Edition,” Oracle, 2024. [Daring].

