LAPORAN PRAKTIKUM

AGORITMA PEMROGRAMAN

“PERULANGAN FOR”

DISUSUN OLEH:

NOFRI ILHAM

2511531013

DOSEN PENGAMPU:

Dr. WAHYUDI, S.T, M.T

ASISTEN PRAKTIKUM:

MUHAMMAD ZAKI AL HAFIZ

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur kehadirat Allah SWT karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan praktikum dengan judul "Perulangan For pada Bahasa Pemrograman Java" tepat pada waktunya.

Laporan ini disusun untuk memenuhi salah satu tugas pada mata kuliah Pemrograman Dasar, khususnya dalam memahami konsep struktur perulangan (looping) pada bahasa pemrograman Java. Melalui praktikum ini, penulis mempelajari bagaimana penggunaan perulangan for dapat membantu dalam mengulang blok kode secara efisien, serta bagaimana penerapannya dalam berbagai kasus pemrograman.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta semua pihak yang telah membantu dalam proses pelaksanaan dan penyusunan laporan ini. Semoga laporan ini dapat memberikan manfaat bagi pembaca yang ingin memahami konsep dasar perulangan dalam Java.

Padang, 31 Oktober 2025

Penulis

Nofri Ilham

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman, sering kali dibutuhkan proses yang harus dijalankan secara berulang dengan pola yang sama, seperti menampilkan data, menghitung deret angka, atau mengolah array. Jika dilakukan secara manual dengan menulis perintah berulang, program akan menjadi panjang, tidak efisien, dan sulit dipelihara.

Untuk mengatasi hal tersebut, digunakan struktur perulangan (looping). Salah satu bentuk perulangan yang paling umum dan efisien adalah perulangan for. Perulangan for memungkinkan program menjalankan blok kode berulang kali berdasarkan kondisi tertentu dengan kontrol yang jelas terhadap inisialisasi, kondisi, dan perubahan nilai variabel.

Pemahaman terhadap perulangan for penting karena merupakan dasar dalam pembuatan algoritma yang efisien, terutama untuk pengulangan dengan jumlah iterasi yang sudah diketahui.

1.2 Tujuan

1. Memahami pengertian dan fungsi perulangan for dalam pemrograman.
2. Mempelajari cara menuliskan dan menggunakan struktur for dengan benar.
3. Dapat menerapkan perulangan for untuk menyelesaikan masalah yang membutuhkan proses berulang, seperti menampilkan data atau melakukan perhitungan berurutan.
4. Melatih kemampuan logika dan berpikir sistematis dalam menyusun program yang efisien menggunakan perulangan.

1.3 Manfaat

1. Menambah pemahaman tentang konsep dasar perulangan dalam pemrograman.
2. Membantu mahasiswa memahami cara kerja logika berulang pada suatu program.
3. Memudahkan dalam menyusun program yang efisien untuk proses yang dilakukan berulang kali.
4. Menjadi dasar dalam pembuatan algoritma yang lebih kompleks di tahap pembelajaran berikutnya.
5. Melatih ketelitian dan kemampuan berpikir logis dalam menyelesaikan permasalahan menggunakan kode program.

BAB II
PEMBAHASAN

2.1 Pengertian

Perulangan for adalah salah satu konsep dasar dalam pemrograman yang memungkinkan program untuk mengulangi serangkaian pernyataan selama kondisi tertentu terpenuhi. Dalam bahasa pemrograman Java, perulangan for digunakan untuk mengulangi tugas-tugas dengan jumlah pengulangan yang telah ditentukan sebelumnya.

Perulangan for digunakan ketika Anda ingin menjalankan serangkaian pernyataan dengan jumlah pengulangan yang telah diketahui sebelumnya. Dalam bahasa pemrograman Java, perulangan for memiliki tiga bagian utama:

1. **Inisialisasi**: Bagian pertama menginisialisasi variabel penghitung dan menentukan nilai awalnya.
2. **Kondisi**: Bagian kedua adalah kondisi yang dievaluasi setiap kali perulangan dimulai. Jika kondisi ini bernilai true, perulangan akan berlanjut; jika kondisi ini bernilai false, perulangan akan berhenti.
3. **Iterasi**: Bagian ketiga adalah iterasi, yang mengubah nilai variabel penghitung pada setiap pengulangan. Ini memastikan bahwa perulangan pada akhirnya akan berhenti ketika kondisi menjadi false.

Berikut adalah bentuk dasar dari perulangan for dalam bahasa pemrograman Java:

1. inisialisasi adalah langkah awal yang dilakukan sebelum perulangan dimulai. Biasanya, ini digunakan untuk menginisialisasi variabel penghitung.
2. kondisi adalah ekspresi yang dievaluasi setiap kali perulangan dimulai. Jika kondisi ini bernilai true, perulangan akan berlanjut; jika kondisi ini bernilai false, perulangan akan berhenti.
3. iterasi adalah langkah yang dilakukan setiap kali perulangan selesai. Ini digunakan untuk mengubah nilai variabel penghitung agar tidak terjebak dalam perulangan tak terbatas.

2.2 Kode Program Pekan 5

Pada pekan 5 mahasiswa telah melakukan praktikum mengenai berbagai program perulangan for dalam bahasa Java. Berikut adalah kode program sebagai contoh dari kode program perulangan.

2.2.1 Kode Program Perulangan For1

```
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
12 }
```

Kode Program 2.1

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Membuat perulangan for pada kode program
3. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program di atas menggunakan perulangan for untuk menampilkan angka 1 sampai 10. Variabel i dimulai dari 1 dan setiap kali perulangan dijalankan, nilainya bertambah 1 hingga mencapai 10. Perintah System.out.println(i); digunakan untuk menampilkan nilai i ke layar.

2.2.2 Kode Program Perulangan For2

```
1  package pekan5;
2
3  public class PerulanganFor2 {
4
5      public static void main(String[] args) {
6          for (int i=1; i <= 10; i++) {
7              System.out.print(i+" ");
8          }
9      }
10
11 }
```

Kode program 2.2

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Membuat perulangan for serta perintah yang dijalankan
3. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program di atas merupakan contoh penggunaan perulangan for untuk menampilkan angka dari 1 hingga 10 pada satu baris. Variabel i diinisialisasi dengan nilai awal 1, kemudian nilainya akan terus bertambah satu per satu sampai mencapai angka 10. Perintah System.out.print(i + " "); berfungsi untuk menampilkan nilai i tanpa membuat baris baru, sehingga seluruh angka muncul secara berurutan dengan spasi sebagai pemisah.

2.2.3 Kode Program Perulangan For3

```
1  package pekan5;
2
3  public class PerulanganFor3 {
4
5      public static void main(String[] args) {
6          int jumlah=0;
7          for (int i=1; i<=10; i++) {
8              System.out.print(i);
9              jumlah= jumlah+i;
10             if (i<10) {
11                 System.out.print(" + ");
12             }
13         }
14         System.out.println();
15         System.out.println("jumlah = "+jumlah);
16
17     }
18
19 }
```

Kode Program 2.3

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Membuat perulangan for serta perintah yang dijalankan
3. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program di atas merupakan penggunaan perulangan for yang tidak hanya menampilkan angka, tetapi juga melakukan proses penjumlahan. Pada program ini, variabel jumlah diinisialisasi dengan nilai 0 dan digunakan untuk menyimpan hasil total dari penjumlahan setiap angka dalam perulangan. Perulangan for dimulai dengan nilai i = 1 dan berjalan hingga i <= 10, di mana setiap kali perulangan berlangsung, program akan menampilkan nilai i menggunakan System.out.print(i);. Setelah itu, nilai i ditambahkan ke variabel jumlah melalui perintah jumlah = jumlah + i;. Bagian if (i < 10) berfungsi untuk menampilkan tanda "+" hanya di antara angka-angka, agar tidak ada tanda plus setelah angka terakhir.

2.2.4 Kode Program Perulangan For4

```
1  package pekan5;
2  import java.util.Scanner;
3  public class PerulanganFor4 {
4
5      public static void main(String[] args) {
6          int jumlah=0;
7          int batas;
8          Scanner input= new Scanner(System.in);
9          System.out.print("Masukkan nilai batas = ");
10         batas = input.nextInt();
11         for (int i=1 ; i<=10; i++) {
12             System.out.print(i);
13             jumlah = jumlah+i;
14             if (i<batas) {
15                 System.out.print(" + ");
16             } else {
17                 System.out.print(" = ");
18
19             }
20         }
21         System.out.println(jumlah);
22     }
23
24 }
```

Kode Program 2.4

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Membuat perulangan for serta perintah yang dijalankan
3. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program ini merupakan pengembangan dari program perulangan sebelumnya dengan tambahan input dari pengguna. Di awal program, variabel jumlah diinisialisasi dengan nilai 0

untuk menampung hasil penjumlahan, sedangkan variabel batas digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna melalui objek Scanner.Setelah pengguna memasukkan nilai batas, program menjalankan perulangan for dari angka 1 hingga 10. Pada setiap iterasi, nilai i ditampilkan ke layar dan ditambahkan ke variabel jumlah.

Selanjutnya, digunakan percabangan if–else untuk mengatur simbol yang muncul di antara angka. Jika nilai i lebih kecil dari batas, maka akan ditampilkan tanda "+", sedangkan jika nilai i telah mencapai batas, tanda yang ditampilkan berubah menjadi "=". Setelah perulangan selesai, program menampilkan hasil akhir dari variabel jumlah sebagai total penjumlahan. Program ini menunjukkan penerapan logika perulangan dan percabangan secara bersamaan untuk menghasilkan tampilan keluaran yang teratur dan sesuai dengan kondisi yang diberikan pengguna.

2.2.5 Kode Program Nested For 0

```
1  package pekan5;
2
3  public class nestedFor0 {
4
5      public static void main(String[] args) {
6          for (int line =1; line <=5; line++) {
7              for (int j=1; j<= (-1* line + 5); j++) {
8                  System.out.print(".");
9
10             }
11             System.out.print(line);
12             System.out.println();
13         }
14
15     }
16
17 }
```

Kode Program 2.5

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Membuat perulangan for serta perintah yang dijalankan

3. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program di atas menggunakan perulangan bersarang (nested loop) untuk menampilkan pola tertentu pada layar. Pada program ini terdapat dua buah perulangan for, yaitu perulangan luar (outer loop) yang mengatur jumlah baris (line) dan perulangan dalam (inner loop) yang mengatur jumlah titik (“.”) yang ditampilkan pada setiap baris. Perulangan luar berjalan dari nilai line = 1 hingga line = 5. Untuk setiap iterasi line, perulangan dalam akan mencetak sejumlah titik berdasarkan ekspresi (-1 * line + 5), yang secara matematis menentukan berapa banyak titik yang harus muncul pada baris tersebut. Setelah perulangan dalam selesai, program akan mencetak nilai line di akhir baris, kemudian berpindah ke baris berikutnya menggunakan System.out.println().

Secara keseluruhan, program ini memperlihatkan bagaimana struktur perulangan bersarang dapat digunakan untuk menghasilkan pola yang teratur dan berubah sesuai dengan nilai variabel pengendali.

2.2.6 Kode Program Perulangan Nested For

```java
package pekan5;

public class nestedFor1 {

    public static void main(String[] args) {
        for (int i=1; i<=5; i++) {
            for (int j=1; j<=5; j++) {
                System.out.print("*");
            }
            System.out.println();
            // to end the line
        }
    }

}
```

Kode Program 2.6

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Membuat perulangan for serta perintah yang dijalankan
3. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program di atas menggunakan perulangan bersarang (nested loop) untuk menampilkan pola bintang berbentuk persegi berukuran 5x5. Terdapat dua perulangan for di dalam program, yaitu perulangan luar (outer loop) dan perulangan dalam (inner loop).

Perulangan luar (for (int i = 1; i <= 5; i++)) berfungsi untuk mengatur jumlah baris yang akan dicetak, sedangkan perulangan dalam (for (int j = 1; j <= 5; j++)) digunakan untuk mencetak karakter * pada setiap baris. Setelah perulangan dalam selesai, program menjalankan perintah System.out.println() untuk pindah ke baris berikutnya.

Dengan struktur seperti ini, setiap baris akan berisi lima karakter bintang, dan hasil akhirnya adalah pola bintang berbentuk kotak. Program ini menunjukkan bagaimana nested loop dapat digunakan untuk membuat pola berulang yang terstruktur dalam pemrograman Java.

2.2.7 Kode Program Nested For2

```java
package pekan5;

public class nestedFor2 {

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

    }

}
```

Kode Program 2.7

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Membuat perulangan for serta perintah yang dijalankan
3. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program ini menggunakan nested loop (perulangan bersarang) untuk menampilkan pola angka berdasarkan hasil penjumlahan dua variabel indeks, yaitu i dan j. Perulangan luar (for (int i = 0; i <= 5; i++)) mengatur jumlah baris yang akan dicetak, sedangkan perulangan dalam (for (int j = 0; j <= 5; j++)) bertugas menampilkan nilai hasil dari i + j pada setiap baris. Setelah satu baris selesai dicetak, perintah System.out.println() digunakan untuk pindah ke baris berikutnya.

Melalui program ini, setiap baris menampilkan pola angka yang berubah sesuai kombinasi nilai i dan j. Program ini menunjukkan bagaimana dua tingkat perulangan for dapat digunakan untuk membuat tampilan data yang berpola dan terstruktur dalam Java.

BAB III
PENUTUP

3.1 Kesimpulan

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa perulangan for merupakan salah satu struktur kontrol yang sangat penting dalam pemrograman Java. Struktur ini digunakan ketika jumlah pengulangan sudah diketahui sebelumnya dan dapat membantu menjalankan blok kode secara berulang dengan cara yang efisien serta mudah dipahami.

Melalui berbagai contoh program yang telah diuji, seperti perulangan sederhana, perulangan dengan operasi aritmatika, hingga perulangan bersarang (nested loop), mahasiswa dapat memahami bagaimana logika perulangan bekerja serta bagaimana cara menggabungkannya dengan kondisi dan input pengguna. Pemahaman ini menjadi dasar penting dalam membangun algoritma yang terstruktur dan efisien pada berbagai kasus pemrograman.

3.2 Saran

Dalam mempelajari konsep perulangan for, disarankan agar mahasiswa lebih banyak berlatih membuat variasi program dengan logika yang berbeda, seperti penggunaan kondisi if–else, input dinamis, maupun kombinasi antar perulangan.

Selain itu, mahasiswa juga perlu memahami hubungan antara variabel penghitung, kondisi perulangan, dan perubahan nilai pada setiap iterasi agar terhindar dari kesalahan logika seperti infinite loop. Dengan latihan dan pemahaman yang baik, konsep perulangan dapat diterapkan secara efektif pada pembuatan program yang lebih kompleks di masa mendatang.

DAFTAR PUSTAKA

[1] Minarsih, “Belajar Bahasa Pemrograman Java #33: Perulangan For Bahasa Java.” [Daring]. Tersedia pada: https://www.minarsih.com/artikel/belajar-bahasa-pemrogaman-java-33-perulangan-for-bahasa-java. [Diakses: 31-Okt-2025].