

PERTEMUAN 8

MENGONTROL ALIRAN PROGRAM PERULANGAN FOR (for loop)

A. TUJUAN PEMBELAJARAN

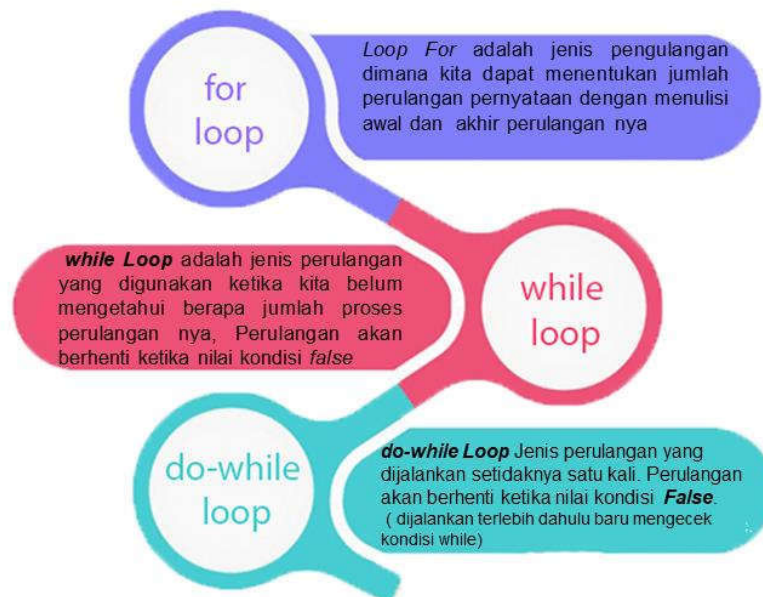
1. Mahasiswa dapat memahami Konsep Kontrol pengulangan / Looping
2. Mahasiswa dapat mengerti aturan kode Kontrol pengulangan For
3. Mahasiswa dapat mempraktekan penggunaan Kontrol pengulangan For

B. URAIAN MATERI

1. Perulangan For

Adalah jenis perulangan dimana kita dapat menentukan jumlah perulangan pernyataan dengan menulisi kondisi nilai awal, nilai akhir perulangan nya, serta kenaikan dan penurunannya.

Perulangan for adalah bagian perintah perulangan dari 3 jenis perulangan seperti pada gambar dibawah ini (pada pertemuan selanjutnya kita akan membahas perulangan while, do while)



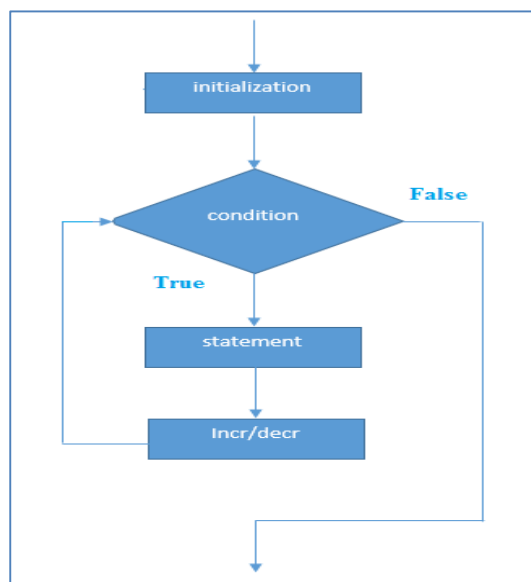
Gambar 8. 1 Jenis Perulangan

2. Bentuk Umum Penulisan For

```
for(initialization; condition ; increment/decrement)
{
    statement(s) ;
}
```

- Initialitaton** : adalah kondisi awal atau nilai awal yang dieksekusi sekali ketika loop/perulangan dimulai. kita dapat menginisialisasi variabel, atau kita dapat menggunakan variabel yang sudah diinisialisasi(opsional).
- Condition** : adalah kondisi kedua yang dijalankan setiap kali untuk menguji kondisi loop. Ini melanjutkan eksekusi sampai kondisinya **false** atau batasan nilai akhir yang ditentukan.
- Increment /Decrement**: adalah menambah atau mengurangi nilai variabel yang ditentukan pada kondisi awal.
- Statement** : adalah pernyataan loop dieksekusi beberapa kali sampai kondisi berikutnya **False**

Berikut diagram flowchart dari perulangan **for**



Gambar 8. 2 diagram Flowchart Perulangan For

Contoh code sederhana perulangan for

```
public class For1 {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
  
        for(int i=1; i<=10; i++){  
            System.out.println("Nilai Variabel i adalah  
: "+i);  
        }  
    }  
}
```

Output :

```
Nilai Variabel i adalah : 1  
Nilai Variabel i adalah : 2  
Nilai Variabel i adalah : 3  
Nilai Variabel i adalah : 4  
Nilai Variabel i adalah : 5  
Nilai Variabel i adalah : 6  
Nilai Variabel i adalah : 7  
Nilai Variabel i adalah : 8  
Nilai Variabel i adalah : 9  
Nilai Variabel i adalah : 10
```

3. Perulangan For bersarang (Nested For)

Perulangan For bersarang /*nested for* adalah perulangan for terdapat perulangan for lainnya, ini merupakan pengembangan dari perulangan For. Pada perulangan For bersarang ini yang terlebih dahulu dieksekusi yaitu For yang ada di dalam terdahulu.

Bentuk Umum penulisan :

```
for(statement; condition check; increment/decrement) {  
    // Code inside outer loop  
    for(statement; condition check; increment/decrement) {  
        // Code inside inner loop  
    }  
    // Code inside outer loop }
```

Contoh program nested for

```
public class nestedFor {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        //loop of i  
        for(int i=1;i<=4;i++){  
            //loop of j  
            for(int j=1;j<=3;j++){  
                System.out.println(i+" "+j);  
            }  
        }  
    }  
}
```

Output :

```
1 1  
1 2  
1 3  
2 1
```

```
2  2
2  3
3  1
3  2
3  3
4  1
4  2
4  3
```

Contoh 2 program membuat piramid nested for

```
public class piramidnested {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        for(int i=1;i<=5;i++){
            for(int j=1;j<=i;j++){
                System.out.print("* ");
            }

            System.out.println();//baris baru
        }
    }
}
```

Output :

```
*
* *
* * *
* * * *
* * * * *
```

4. Perulangan For each

Perulangan for-each digunakan untuk melintasi array atau koleksi di Java. Lebih mudah digunakan daripada loop for sederhana karena kita tidak perlu menambah nilai dan menggunakan notasi subscript.

For each bekerja atas dasar elemen dan bukan indeks. Ini akan mengembalikan elemen satu per satu dalam variabel yang ditentukan.

Bentuk Umum penulisan For each :

```
for(data_type variable : array_name){  
    //code to be executed  
}
```

Catatan : keyword *array* akan dibahas pada pertemuan selanjutnya

Contoh program perulangan *for each*.

```
public class foreachDemo {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        int arr[]={12,23,44,56,78};  
        //Printing array using for-each loop  
        for(int i:arr){  
            System.out.println(i);  
        }  
    }  
}
```

Output :

```
12
23
44
56
78
```

5. Label For Loop

Kita dapat memiliki nama masing-masing pada Java for loop dengan pelabelan. Contoh menggunakan label sebelum perulangan for. Ini berguna saat menggunakan loop for bersarang karena kita dapat memutus atau melanjutkan loop for tertentu.

Bentuk umum penulisan pelabelan ***For***

```
labelname:
for(initialization; condition; increment/decrement){
//code to be executed
}
```

Contoh program label *for*

```
public class labeledFor {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        aa:
        for(int i=1;i<=3;i++){
            bb:
            for(int j=1;j<=3;j++){
```

```
                if (i==2&& j==2) {  
                    break aa;  
                }  
                System.out.println(i+" "+j);  
            }  
        }  
    }  
}
```

Output :

```
1 1  
1 2  
1 3  
2 1
```

Contoh lain program pelabelan for

```
public class labeledfor2 {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        aa:  
        for(int i=1;i<=3;i++){  
            bb:  
            for(int j=1;j<=3;j++){  
                if (i==2&& j==2) {  
                    break bb;  
                }  
            }  
        }  
    }  
}
```



```
        System.out.println(i+" "+j);
    }
}
}
```

Output :

```
1 1
1 2
1 3
2 1
3 1
3 2
3 3
```

6. Infinite For Loop

infinite loop adalah perulangan terus menerus atau kondisinya tidak akan pernah mengembalikan *false*. Contoh inisialisasi awal nilai variabel *i* ke 1, selanjutnya : *i>1*, dan menaikkan *i++* (increment) sehingga nilai *i* tidak akan pernah mengembalikan *false*. Pada akhirnya akan mengarah pada kondisi loop terus menerus. *Infinite loop* bisa dilakukan juga jika penulisan setelah perintah *for* tidak diberikan nilai variable awal, akhir dan penaikan dan penurun nya ,

Contoh sintaks penulisan :

```
for(;;){
    //kode untuk di eksekusi :
}
```

Contoh program *infinite loop*

```
public class ForExample {
```

```
public static void main(String[] args) {  
    //Gunakan tanpa kondisi pada for loop  
    for(;;){  
        System.out.println("infinitive loop");  
    }  
}  
}
```

Output :

```
infinitive loop  
infinitive loop  
infinitive loop  
infinitive loop  
infinitive loop  
ctrl+c
```

7. Enhanced for loop

Enhanced for loop berguna ketika kita ingin mengulangi perintah pada Array/larik(Array akan dibahas lebih lanjut pertemuan berikutnya), *Enhanced for loop* melakukan iterasi secara otomatis berdasarkan jumlah elemen array sehingga tidak perlu lagi menentukan batasan awal dan batasan akhir melalui variabel control control nya. Keuntungannya ketika menggunakan *enhanced for statement* adalah menghindari proses iterasi jumlah elemen array diluar yang seharusnya karena kesalahan memberi nilai pada control variabel.

Bentuk Umum penulisan *enhanced for* :

```
for (parameter : nama-array)  
    pernyataan tunggal;
```

atau

```
for (parameter : nama-array ) {
```

```
    blok pernyataan;  
}
```

contoh :

```
class ForLoopExample3 {  
    public static void main(String args[]){  
        int arr[]={2,11,45,9};  
        for (int num : arr) {  
            System.out.println(num);  
        }  
    }  
}
```

Output :

```
2  
11  
45  
9
```

C. LATIHAN /TUGAS

1. Buat program pengulangan for untuk menampilkan bilangan pertambahan 5. Nilai awal dan nilai akhir ditentukan menggunakan input menggunakan keyboard (*Scanner*)

Contoh output :

```
Nilai Awal ?: 5  
Nilai Akhir ?: 100
```

Hasil nya :

- 1. 5**
- 2. 10**
- 3. 15**
- 4. 20**
- 5. 25**
- 6. 30**
- 7. 35**
- 8. 40**
- 9. 45**
- 10. 50**

2. Buat program pengulangan for untuk menampilkan bilangan tahun, Nilai awal dan nilai akhir ditentukan menggunakan input menggunakan keyboard (*Scanner*)

Contoh output :

Tahun Akhir ?: 2015

Tahun Awal ?: 1950

Hasil nya :

2015

2014

2013

2012

.

.

1950

D. REFERENSI

Horstmann Cay S., (2011). *Big Java 4th Edition*, san jose university , united state Of America. RRD jefferson city publishing.

Deitel Paul , Deitel Harvey, (2012) Java how to program eighth edition, pearson education, Boston Massachusetts , USA, *publishing as prentice hall*.

Rose Cristhoper, (2017), Java Succinctly Part 2, Morrisville, NC 27560, USA, Syncfusion, Inc.

Downey Allen B. , Mayfield Chris, (2017), Think Java, Needham, Massachusetts, USA, Green Tea Press

Hayes Helen, (2021), BeginnersBook.com, <https://beginnersbook.com/java-tutorial-for-beginners-with-examples/>, di akses pada tanggal 21 November 2021.

Sonoo Jaiswal, (2021) JavaTpoint offers college campus training , <https://www.javatpoint.com/java-tutorial>, diakses pada tanggal 1 Desember 2021