PERTEMUAN 9
MENGONTROL ALIRAN PROGRAM PERULANGAN WHILE, DO WHILE

A. TUJUAN PEMBELAJARAN

1. Mahasiswa dapat memahami Konsep Kontrol pengulangan / Looping While Do while
2. Mahasiswa dapat mengerti aturan penulisan Kontrol pengulangan While
3. Mahasiswa dapat mengerti aturan penulisan Kontrol pengulangan Do While
4. Mahasiswa dapat mempraktekan penggunaan Kontrol pengulangan While dan Do While

B. URAIAN MATERI

1. Perulangan While

Perulangan *while* digunakan untuk mengulang bagian program sampai kondisi Boolean yang ditentukan bernilai ***true***. setelah kondisi Boolean bernilai ***false***, pengulangan otomatis berhenti.

Jika jumlah iterasi tidak tetap, disarankan untuk menggunakan perulangan while.

Bentuk umum penulisan perulangan ***while*** adalah **:**

```
while(condition)
{
   statement(s);
   Increment / decrement

}
```

a. ***Condition***/kondisi:

Adalah ekspresi yang diuji. Jika kondisinya benar atau *true*, badan loop atau statement dieksekusi. Ketika kondisi bernilai *false*, maka keluar dari perulangan *while*.

contoh i<=90

b. *Statement*/Pernyataan

badan program yang dieksekusi selama kondisi ***true***

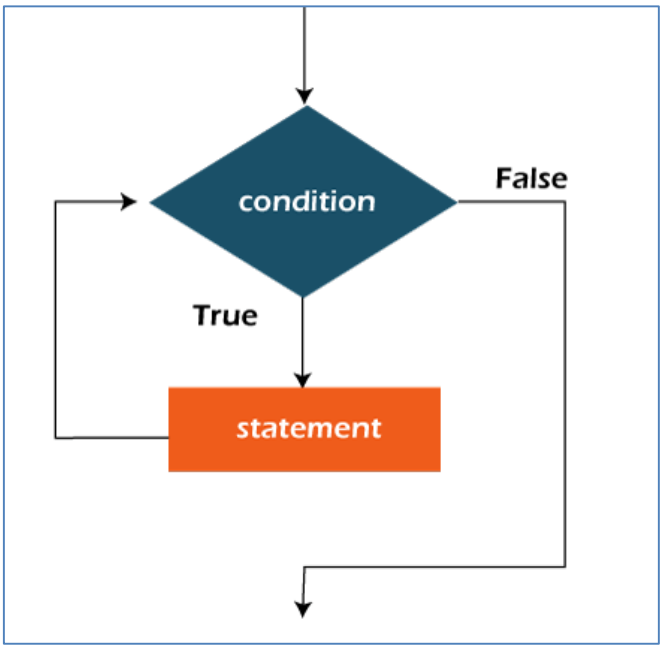
c. *Increment* /decrement

Adalah ekspresi penambahan / pengurangan suatu variable kondisi

Contoh **i++**

2. Gambar diagram flowchart perulangan *While*

Hal penting tentang while loop adalah, terkadang perulangan ini tidak dapat dieksekusi. Jika kondisi yang akan diuji menghasilkan *false*, badan perulangan akan dilewati dari pernyataan pertama setelah perulangan *while* akan dieksekusi.



Gambar 9. 1 diagram flowchart perulangan while

Contoh program perulangan while :

a. Pada contoh di bawah ini, kita akan mencetak nilai bilangan integer dari 10 hingga 100. Jika melewati angka 100 maka perulangan akan berhenti ekspresi variabel menggunakan penambahan (increment)

```
public class while1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    int i=10;
    while(i<=100){
        System.out.println(i);
    i+=10;
    }
    }
}
```

Output :

```
10
20
30
40
50
60
70
80
90
100
```

b. Pada contoh berikutnya, kita akan mencetak nilai bilangan integer dari 15 hingga 5. Jika angka dibawah 5 maka perulangan akan berhenti ekspresi menggunakan pengurangan/penurunan 1 (decrement)

```
public class while2 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    int i=10;
    while(i>=1){
        System.out.println(i);
    i--;
    }
    }
 }
```

Output :

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

3. Infinite while loop

Infinite while loop atau perulangan yang tidak akan pernah berakhir, adalah loop while yang tak terbatas. karena kondisinya selalu ***true*** contoh

program dibawah ini adalah *Infinite while loop* karena i>1 akan selalu benar saat kita menaikkan nilai i di dalam while loop.

Contoh program :

```
public class infinitewhile {
     /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
         int i=10;
         while(i>1)
         {
             System.out.println(i);
              i++;
         }
    }
}
```

Output :

```
10
11
12
13
14
Ctrl+C
```

4. Perulangan while dengan pernyataan Break

Saat bekerja dengan while loop, terkadang diinginkan untuk melewatkan beberapa pernyataan di dalam loop atau segera menghentikan loop tanpa memeriksa ekspresi pengujian.

Dalam kasus seperti itu, pernyataan *break*/ Pernyataan *break* di Java akan mengakhiri loop, dan kontrol program berpindah ke pernyataan berikutnya setelah loop.biasanya digunakan dengan pernyataan pengambilan keputusan (Pernyataan if...else).

Berikut contoh pernyataan break pada while Java:

```
public class whileBreak {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
          // for loop
        for (int i = 1; i <= 10; ++i) {

            // if the value of i is 5 the loop terminates
            if (i == 5) {
                break;
            }
            System.out.println(i);
        }
    }

}
```

Output :

```
1
2
3
4
```

Pada program di atas, kita menggunakan perulangan for untuk mencetak nilai i pada setiap iterasi.

Perhatikan pernyataan dibawah,

```
if (i == 5) {
    break;
}
```

Artinya ketika nilai i sama dengan 5, loop berakhir. Oleh karena hasil output nilai kurang dari 5 saja.

5. Perulangan Do While

Perulangan do-while digunakan untuk mengulangi bagian dari program berulang kali, sampai kondisi yang ditentukan benar atau *true*. setidaknya perulangan dieksekusi satu kali, disarankan untuk menggunakan loop do-while.

Perulangan do-while disebut exit control loop. Oleh karena itu, tidak seperti perulangan *for* dan while, do-while memeriksa kondisi di ujung loop body. Perulangan do-while loop dieksekusi setidaknya sekali karena kondisi diperiksa setelah badan perulangan.

Bentuk umum perulangan ***do-wile***

```
do
{
  statement(s);
} while(condition);
```

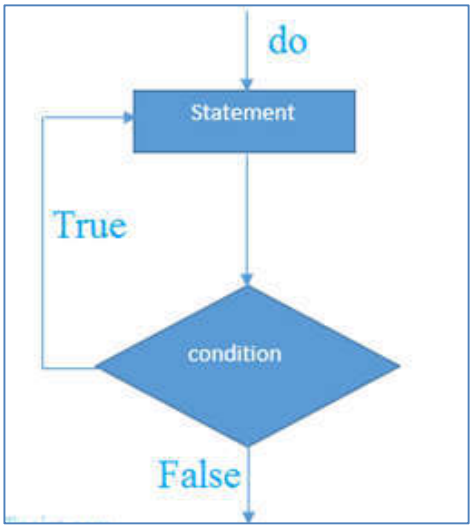
a. *Condition*/kondisi:

adalah ekspresi yang diuji. Jika kondisinya benar atau *true* , badan loop atau statement dieksekusi. Ketika kondisi bernilai *false*, maka keluar dari perulangan

contoh i<=90.

b. *Statement*/Pernyataan

badan program yang dieksekusi selama kondisi ***true***



Gambar 9. 2 diagram flowchart do-while

Contoh 1 program perulangan do-while (increment)

Pada contoh di bawah ini, hasilnya akan mencetak nilai integer dari 1 hingga 10. Berbeda dengan pengulangan for, kita perlu menginisialisasi variabel yang digunakan terlebih dahulu dalam suatu kondisi (contoh disini i). Jika tidak, perulangan akan mengeksekusi tanpa batas.

```
public class dowhileloop {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
         int i=1;
         do{
            System.out.println(i);
          i++;
          }while(i<=10);
         }}
```

Output :

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

Contoh 2 program perulangan do-while (decrement)

```
public class doWhiledecrement {


    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
          int i=10;
         do{
            System.out.println(i);
          i-=3;
          }while(i>=1);
         }
```

```
        }
```

Output :

```
   10
   7
   4
   1
```

C. LATIHAN/TUGAS

1. Modifikasi program pengulangan nested for dibawah ini menggunakan perulangan *while*

```
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 yang diharapkan :

```
*
```

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

2. Buat Program deret bilangan bulat dengan pertambahan 5, variabel nilai awal dan nilai akhir di input menggunakan *scanner* :
   Output yang diharapkan :

```
Masukan bilangan awal :  3
Masukan bilangan akhir : 15
Hasil deret bilangan :
3    , 8, 13
```

3. Lengkapi bidang kosong pada potongan program dibawah ini untuk menampilkan variabel i

```
int i = 1;
            (i < 6) {
  System.out.println(i);
            ;
}
```

D. REFERENSI

Horstmann Cay S., (2011). *Big Java 4t^h 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