Praktikum VI :
Perulangan II

Tujuan Praktikum :

1. Praktikan dapat memahami konsep perulangan while dan do-while.
2. Praktikan dapat memahami perbedaan perulangan while dan do-while.
3. Praktikan dapat memahami konsep nested while dan do-while.
4. Praktikan dapat mengimplementasikannya ke dalam program.

Perulangan while dan do-while

- Perulangan while dan do-while termasuk ke dalam jenis uncounted loop.
- Uncounted loop ini memiliki jumlah pengulangan yang tidak tentu. Tapi, tidak menutup kemungkinan juga, jumlah pengulangannya dapat ditentukan.

Perulangan While

- While bisa kita artikan selama artinya selama kondisi bernilai true maka perulangan akan terus berjalan. Cara kerja perulangan ini seperti percabangan, ia akan melakukan perulangan selama kondisinya bernilai **true**.
- Perulangan while akan berhenti sampai kondisi bernilai false. Tidak menutup kemungkinan juga, perulangan while dapat melakukan counted loop.

Deklarasi Umum While :

```
while (kondisi) {

    statement yang akan diulang;

}
```

- *kondisi* bisa kita isi operasi yang menghasilkan nilai boolean (true/false).

Contoh Program

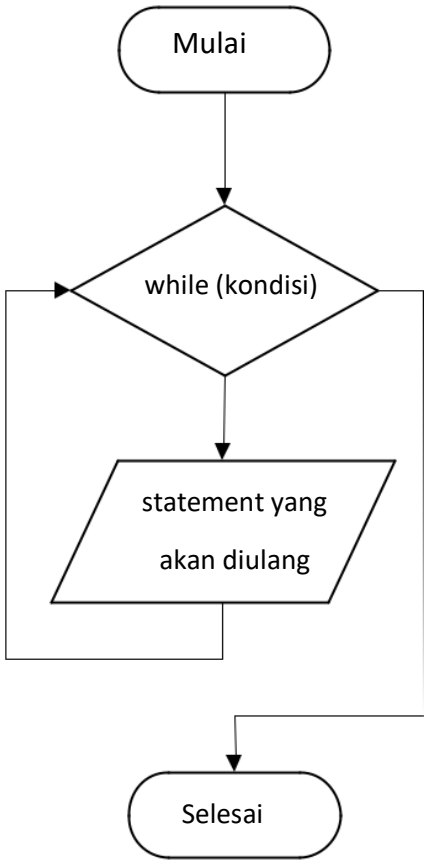
```
public static void main(String[] args) {
        int i = 0;
        while (i <= 5) {
            //blok kode yang akan diulang
            System.out.println("Perulangan ke-" + i);
            //increment nilai i
            i++;
        }
    }
```

Output :

```
run:
Perulangan ke-0
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
Perulangan ke-5
BUILD SUCCESSFUL (total time: 0 seconds)
```

❖ Coba hapuskan sintaks i++ pada source code dan lihat apa perbedaannya

Flowchart perulangan while



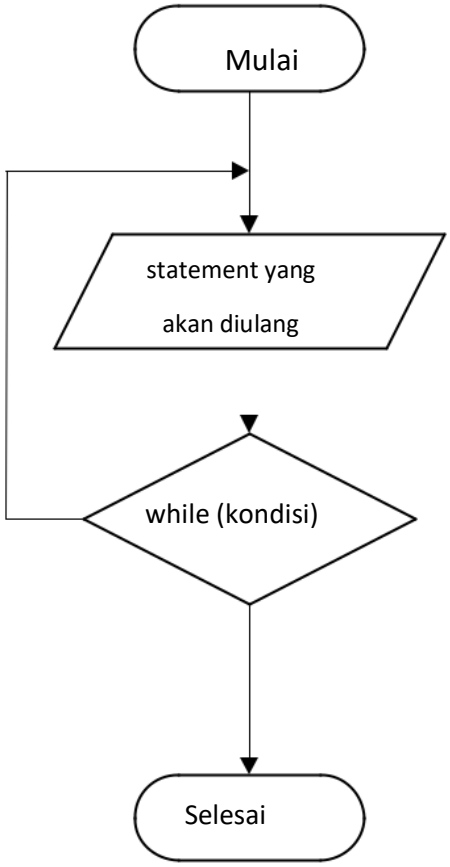
Perulangan do-while

- Cara kerja perulangan do-while sebenarnya sama seperti perulangan *while*.
- Bedanya, *do-while* melakukan eksekusi terlebih dulu. Kemudian mengecek kondisinya. Jadi pada do-while ini akan terulang minimal 1 kali.
- Jadi kerjakan dulu (do), baru dicek dulu kondisinya. Kalau kondisi bernilai true, maka lanjutkan perulangan.

Deklarasi Umum do-while

```
do{
        statement yang akan diulang;
}while(kondisi);
```

Flowchart perulangan do-while :



Contoh Program :

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

        int i = 0;
        do {
            System.out.println("Perulangan ke-" + i);
            i++;
        } while (i < 1);
        System.out.println("Perulangan telah berhenti");
    }
```

Output :

```
run:
Perulangan ke-0
Perulangan telah berhenti
BUILD SUCCESSFUL (total time: 0 seconds)
```

Nested while dan do-while

- Jenis perulangan di dalam perulangan bisa berbeda, misalnya di dalam perulangan while ada do-while.
- Contoh program nested while :

```
public static void main(String[] args) {
        int outer = 1;
        while (outer < 3) {
            int inner = 5;
            while (inner < 8) {
                System.out.println(outer + " " +
inner);
                inner++;
            }
            outer++;
        }
    }
```

Outputnya :

```
run:
1 5
1 6
1 7
2 5
2 6
2 7
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Ubah kondisi inner < 8 menjadi inner < 12 dan lihat apa yang terjadi.
- Ubah inisialisasi dari int outer=1; menjadi int outer=-4; dan lihat apa yang terjadi
- Coba tambahkan lagi perulangan while dalam perulangan while di bagian **inner** dengan menggunakan variabel penghitung bernama **deepest**. Atur **deepest** dengan nilai 10 sampai 14 dan lihat output dari **outer**, **inner** dan **deepest**.

Contoh program nested while :

```
public static void main(String[] args) {
        int i = 1, j = 1;
        int hasilKali;
        System.out.println("Tabel Perkalian");
        while (i <= 2) {
            while (j <= 5) {
                hasilKali = i * j;
                System.out.println(i + " + " + j + "
= " + hasilKali);
                j++;
            }
            i++;
        }
    }
```

Outputnya :

```
run:
Tabel Perkalian
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
BUILD SUCCESSFUL (total time: 0 seconds)
```

❖ Contoh program nested do-while :

```
public static void main(String[] args) {
        int i = 1;
        do {
            int k = 3;
            do {
                System.out.println(" ");
                k--;
            } while (k >= i);
            int j = 1;
            do {
                System.out.println(i + " ");
                j++;
            } while (j <= i);
            System.out.println("");
            i++;
        } while (i <= 5);
    }
```

Outputnya :

```
run:

1


2
2

3
3
3

4
4
4
4

5
5
5
5
5
BUILD SUCCESSFUL (total time: 0 seconds)
```