



Percabangan, Perulangan dan Nested Class



Percabangan

- Menggunakan statement **if** - **else** atau **switch**.
- *Syntax* penulisan **if** - **else** :

```
if (kondisi)  
    statement1  
else  
    statement2
```



if - else dapat berada lagi di dalam
if - else (nested if - else) :

if (*kondisi*)

```
    if (kondisi)  
        statement1  
  
    else  
        statement2
```

else

statement3



Atau seperti di bawah ini :

```
if (kondisi)  
    statement1
```

```
else
```

```
    if (kondisi)  
        statement2  
    else  
        statement3
```

Contoh

```
void hitung()  
{  
    nAngka=(0.25*tgs)+(0.35*mid)+(0.4*uas);  
    System.out.println("Nilai Akhir : " + nAngka);  
    if(nAngka>85)  
        {nHuruf='A';}  
    else if ((nAngka>70) && (nAngka<=85))  
    {  
        nHuruf='B';  
    }  
    else if ((nAngka>55) && (nAngka<=70))  
        {nHuruf='C';}  
    else if ((nAngka>39) && (nAngka<=55))  
        {nHuruf='D';}  
    else {nHuruf='E';}  
    System.out.println("Nilai Huruf : " + nHuruf);  
}
```

Contoh

```
1 public class Branching{
2     public static void main(String args[]){
3         char nilai='a';
4         String predikat;
5         if(nilai=='a'){
6             predikat="Excellent";
7         }else if(nilai=='b'){
8             predikat="Good";
9         }else if(nilai=='c'){
10            predikat="Fair";
11        }else if(nilai=='d'){
12            predikat="Dafuq";
13        }else if(nilai=='e'){
14            predikat="Fail";
15        }else{
16            predikat="No Such Grade";
17        }
18        System.out.println("Nilai : "+nilai);
19        System.out.println("Predikat: "+predikat);
20    }
```


- Syntax penulisan **switch** :

```
switch (ekspresi) {  
    case Constant1 :  
        statement1; break;  
    case Constant2 :  
        statement2; break;  
    ...  
    default :  
        defaultStatement  
}
```

Contoh

```
1 public class Branching2{
2     public static void main(String args[]){
3         char nilai='a';
4         String predikat;
5         switch(nilai){
6             case 'a':predikat="Excellent";break;
7             case 'b':predikat="Good";break;
8             case 'c':predikat="Fair";break;
9             case 'd':predikat="Dafuq";break;
10            case 'e':predikat="Fail";break;
11            default:predikat="No Such Grade";
12        }
13        System.out.println("Nilai : "+nilai);
14        System.out.println("Predikat: "+predikat);
15    }
16 }
```


Contoh

```
2 public class Branching3{
3     public static void main(String args[]){
4         Scanner in=new Scanner(System.in);
5         String nilai;
6         String predikat;
7         System.out.print("Nilai : ");nilai=in.nextLine();
8         switch(nilai){ //type char, integer, String
9             case "a":predikat="Excellent";break;
10            case "b":predikat="Good";break;
11            case "c":predikat="Fair";break;
12            case "d":predikat="Dafuq";break;
13            case "e":predikat="Fail";break;
14            default:predikat="No Such Grade";
15        }
16        System.out.println("Nilai : "+nilai);
17        System.out.println("Predikat: "+predikat);
18    }}
19 //default dpt diletakkan dimanapun
20 //break harus ada
```

- *Syntax* penulisan **switch** di atas, dapat di-ekuivalen-kan dengan penulisan **if-else** yang *nested*, menjadi :

```
nilaiSwitch = ekspresi;  
if (nilaiSwitch == nilai_1)  
    statement_1  
else if (nilaiSwitch == nilai_2)  
    statement_2  
  
...  
else if (nilaiSwitch == nilai_i)  
    statement_i  
else  
    statement_(i+1)
```

Perulangan dengan `while`

- Mengulang badan *looping* selama kondisi adalah benar.
- Badan *looping* adalah *statement-statement* yang berada di antara tanda kurung kurawal.
- *Syntax* :

```
while (kondisi) {  
    statement1;  
    statement2;  
}
```

Perulangan dengan while

```
class turun1 {  
    static int max = 10, i = 1;  
    public static void main(String[] args) {  
        while (max > 0) {  
            System.out.println("Angka "+i+": "+max);  
            i = i + 1;  
            max = max - 1  
        }  
    }  
}
```

Contoh while

```
1 public class MyWhile{
2     public static void main(String args[]) {
3         int c=0;
4         while(c<10) {
5             System.out.println("while - "+c);
6             c++;
7         }
8     }
9 }
```

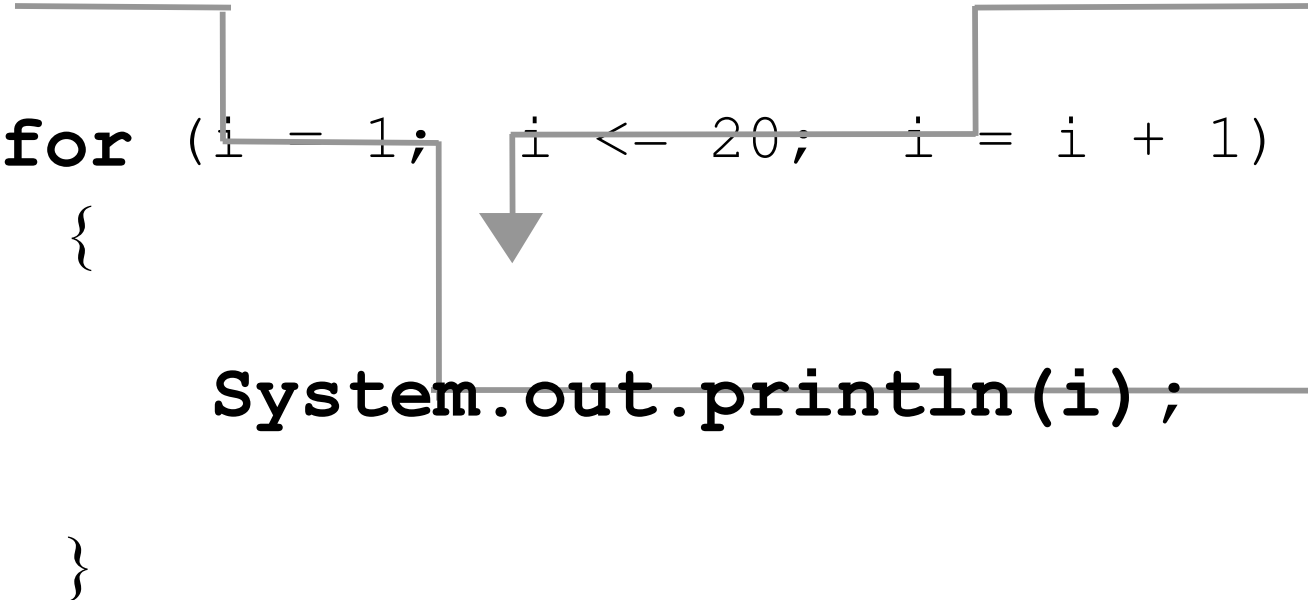

Perulangan dengan for

```
for (statement1; kondisi;  
statement2) {  
    statement3;  
}
```

Hal ini sama dengan perintah dengan **while** :

```
statement1;  
while (kondisi) {  
    statement3;  
    statement2;  
}
```

Perulangan dengan for



```
for (i = 1; i <= 20; i = i + 1)
{
    System.out.println(i);
}
```

The diagram illustrates the execution flow of a for loop. A horizontal line at the top represents the loop header. A vertical line descends from the left side of the header to the opening curly brace of the loop body. Another vertical line descends from the right side of the header to the closing curly brace. A horizontal line connects the bottom of these two vertical lines, passing through the loop body. An arrow points from the middle of the header line down to the loop body, indicating the start of the loop iteration.

Perulangan dengan for

```
class turun1for {  
    static int max = 10, i = 1;  
    public static void main (String[] args) {  
        for (i = 1; max > 0; i = i + 1) {  
            System.out.println("Angka"+i+": "+max);  
            max = max - 1;  
        }  
    }  
}
```

Contoh for

```
1 public class ForLoop{
2     public static void main(String args[]){
3         for(int i=0;i<10;i++){
4             System.out.println("for-"+i);
5         }
6         whileLoop();
7         loopWhile();
8     }
9     static void whileLoop(){
10         int i=0;
11         while(i<10){
12             System.out.println("while - "+i);
13             i++;
14         }
15     }
16     static void loopWhile(){
17         int i=0;
18         do{
19             System.out.println("dowhile - "+i+"->"+
20                 getKeterangan(i));
21             i++;
22         }while(i<10);
23     }
24     static String getKeterangan(int x)
25     {
26         if(x>5) {return "Joss - "+x;}
27         else {return "Biasa - "+x;}
28     }
29 }
```

Perulangan dengan do-while

```
do {  
    statement1;  
    statement2;  
} while (kondisi);
```

ekivalen

```
statement1;  
statement2;  
while (kondisi) {  
    statement1;  
    statement2;  
}
```

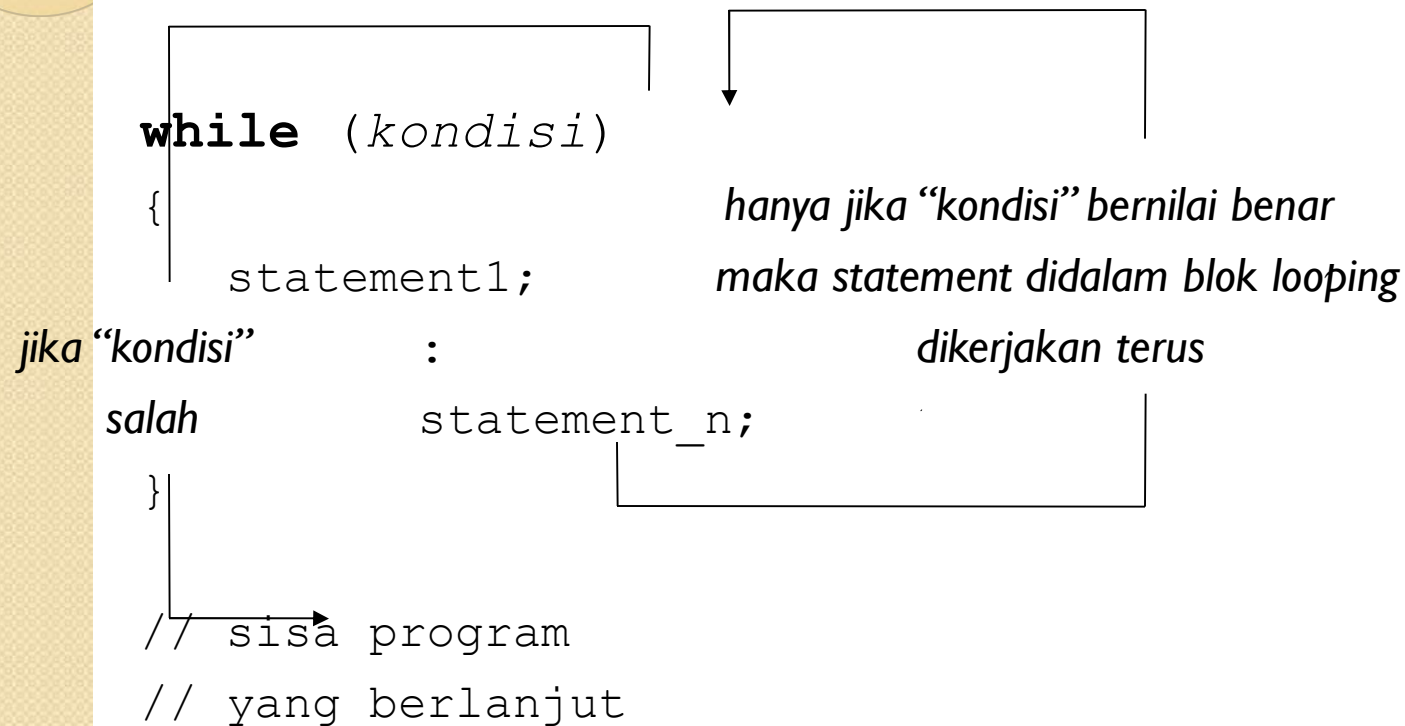

Perulangan dengan do-while

```
class turun1do {  
    static int max = 10, i = 1;  
    public static void main (String[] args) {  
        do {  
            System.out.println("Angka"+i+": "+max);  
            i = i + 1;  
            max = max - 1;  
        }  
        while (max > 0);  
    }  
}
```

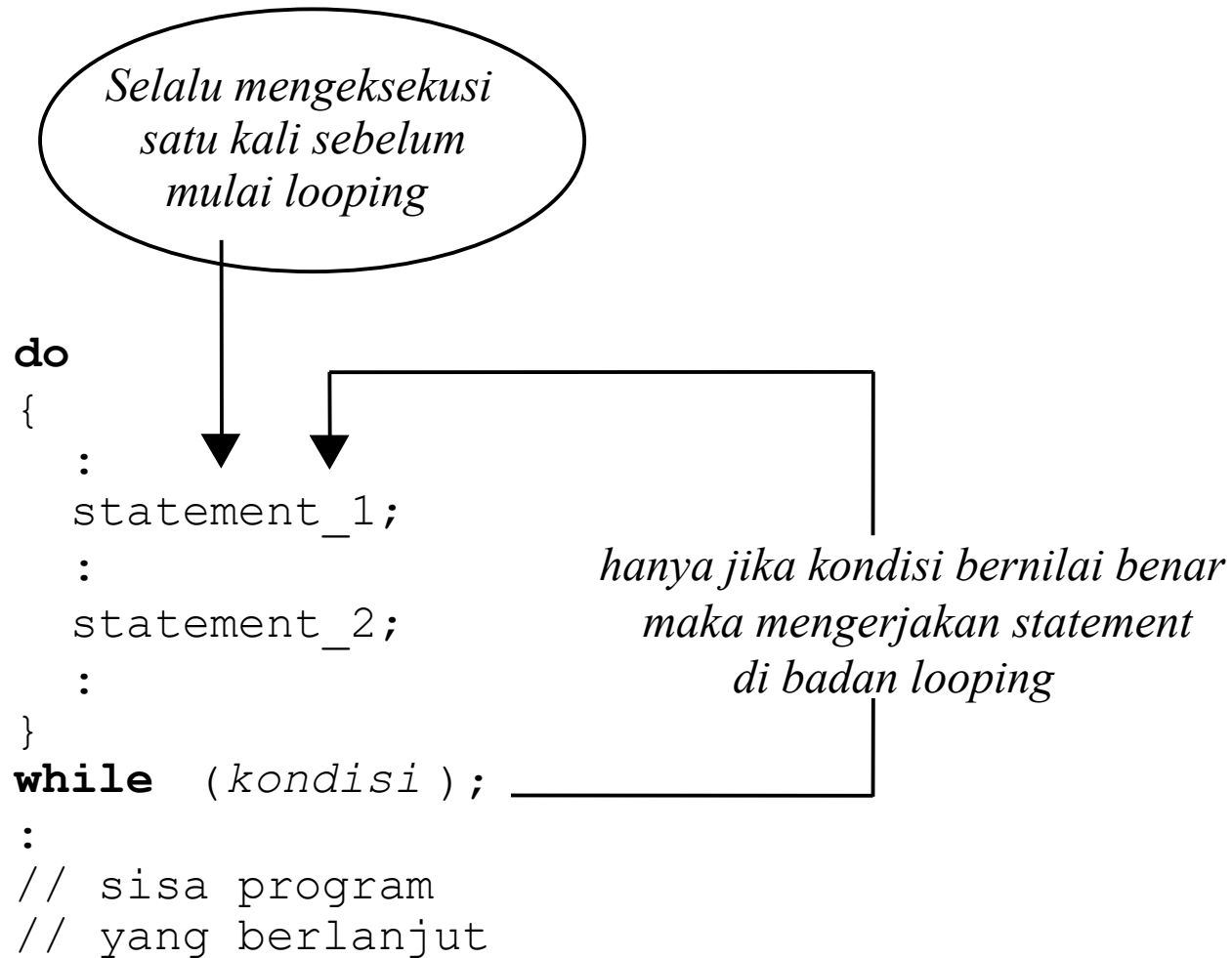
Contoh do-while

```
1 public class MyDoWhile{
2     public static void main(String args[]) {
3         int c=0;
4         do{
5             System.out.println("while - "+c);
6             c++;
7         }while (c<10) ;
8     }
9 }
```

Cara kerja while

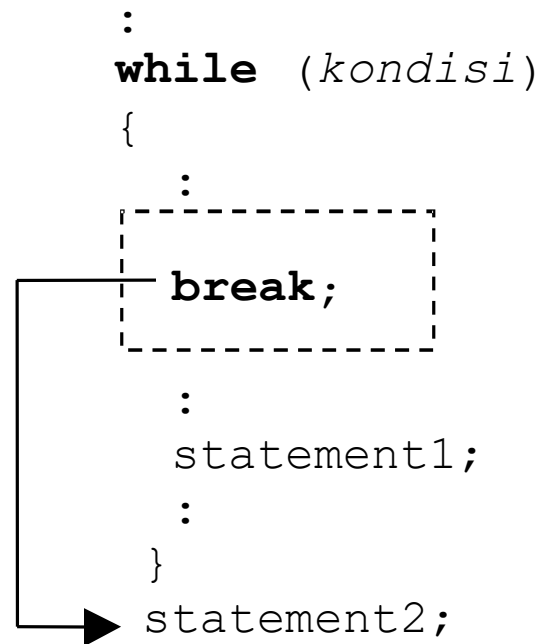


Cara kerja do-while



Break

Menghentikan *looping* secara paksa; artinya sebelum kondisi *looping* terpenuhi, *looping* dapat dihentikan.

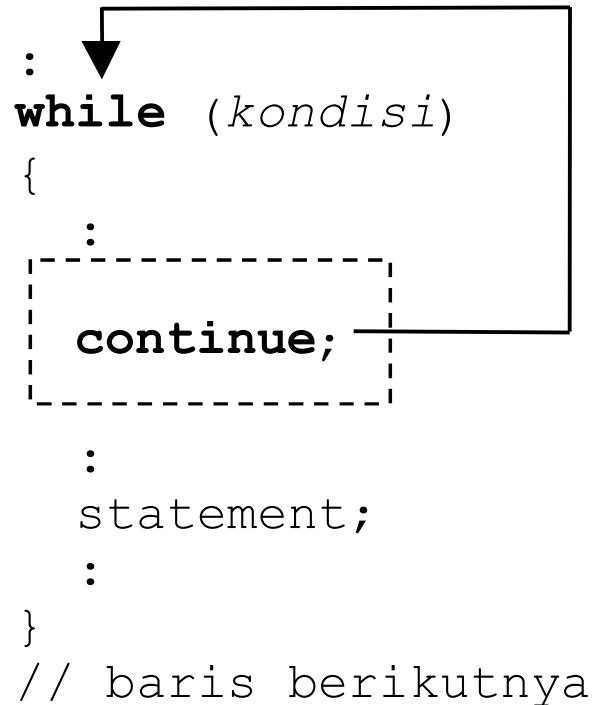


Contoh Break

```
1 public class MyWhileBreak{
2     public static void main(String args[]){
3         int c=0;
4         while(c<10){
5             c++;
6             if(c==7) {break;} else {
7                 System.out.println("while - "+c) ;}
8         }
9     }
10 }
```

Continue

Menghentikan *looping* yang terjadi saat itu dan kembali ke *looping* selanjutnya (dari awal).



```
:  
while (kondisi)  
{  
:  
    continue;  
:  
    statement;  
:  
}  
// baris berikutnya
```

Contoh continue

```
1 public class MyWhileContinue{
2     public static void main(String args[]){
3         int c=0;
4         while (c<10) {
5             c++;
6             if(c==7) {continue;} else {
7                 System.out.println("while - "+c) ;}
8         }
9     }
10 }
11 }
```

Nested class

- Merupakan **class** di dalam **class**; artinya dalam satu *source code* terdiri dari banyak **class**.
- Contohnya :

```
class hewan {  
    class herbivora {  
        ... }  
    class karnivora {  
        ... }  
}
```

Static class

- Hanya kelas bersarang yang bisa statis (*Only nested classes can be static*).
- Kelas yang ada kelas bersarang dikenal dengan kelas luar.
- Ada 2 :
 - Static class
 - Non-Static Class

Static & Non-Static class

- Static class bersarang tidak memerlukan referensi kelas luar, tapi kelas nested non-static class memerlukan referensi kelas luar.
- Kelas dalam (kelas bersarang non-statis) dapat mengakses anggota kelas Outer statis dan non-statis. Kelas statis tidak dapat mengakses anggota non-statis dari kelas luar. Hanya bisa mengakses anggota statis kelas luar.
- Sebuah instance dari *inner class* tidak dapat dibuat tanpa sebuah instance dari *outer class* dan *inner class* dapat merujuk data dan metode yang didefinisikan di *outer class* dimana ia bersarang, jadi tidak perlu melewati referensi objek ke konstruktor dari kelas dalam

Nested class

```
1 class Luar
2 { void cetakLuar()
3 { System.out.println("cetak Luar.....");
4 //cetakTengah(); --> error
5 }
6 class Tengah
7 { void cetakTengah()
8 { cetakLuar();
9 System.out.println("cetak Tengah.....");
10 //cetakDalam();-->error
11 }
12 class Dalam
13 { void cetakDalam()
14 { cetakLuar();
15 cetakTengah();
16 System.out.println("cetak Dalam.....");
17 }
18 }
19 }
20 }
21 //class luar tdk bisa akses dalam/inner
22 //class dalam/inner bisa akses kelas di luarnya
```

Nested class

```
1 public class TestLuar
2 {
3     public static void main(String[] a)
4     {
5         Luar l=new Luar();
6         Luar.Tengah t=l.new Tengah();
7         Luar.Tengah.Dalam d=t.new Dalam();
8         l.cetakLuar();
9         t.cetakTengah();
10        d.cetakDalam();
11    }
12 }
```

Contoh Nested class

```
1  /* Java program to demonstrate how to implement static and non-static
2     classes in a java program. */
3  class OuterClass{
4      private static String msg = "GeeksForGeeks";
5      // Static nested class
6      public static class NestedStaticClass{
7          // Only static members of Outer class is directly accessible in nested
8          // static class
9          public void printMessage() {
10             // Try making 'message' a non-static variable, there will be
11             // compiler error
12             System.out.println("Message from nested static class: " + msg);
13         }
14     }
15     // non-static nested class - also called Inner class
16     public class InnerClass{
17         //Both static and non-static members of Outer class are accessible in this Inner class
18         public void display(){
19             System.out.println("Message from non-static nested class: " + msg);
20         }
21     }
22 }
```


Contoh Nested class

```
23 public class Main
24 {    // How to create instance of static and non static nested class?
25     public static void main(String args[]){
26         // create instance of nested Static class
27         OuterClass.NestedStaticClass printer = new OuterClass.NestedStaticClass();
28         // call non static method of nested static class
29         printer.printMessage();
30         // In order to create instance of Inner class we need an Outer class
31         // instance. Let us create Outer class instance for creating
32         // non-static nested class
33         OuterClass outer = new OuterClass();
34         OuterClass.InnerClass inner = outer.new InnerClass();
35         // calling non-static method of Inner class
36         inner.display();
37         // we can also combine above steps in one step to create instance of
38         // Inner class
39         OuterClass.InnerClass innerObject = new OuterClass().new InnerClass();
40         // similarly we can now call Inner class method
41         innerObject.display();
42     }
43 }
```

D:\JavaMagelang>javac Main.java

```
D:\JavaMagelang>java Main
Message from nested static class: GeeksForGeeks
Message from non-static nested class: GeeksForGeeks
Message from non-static nested class: GeeksForGeeks
```



Ada pertanyaan



Latihan

1. Buatlah program untuk menghitung nilai PBO anda !
2. Buatlah program untuk menyelesaikan rumus ABC, $D=b^2-4ac$

Rehat Sejenak

- Sushi
- Hi Cewek