



Java Fundamental | Bootcamp

Looping

Subjects : Looping

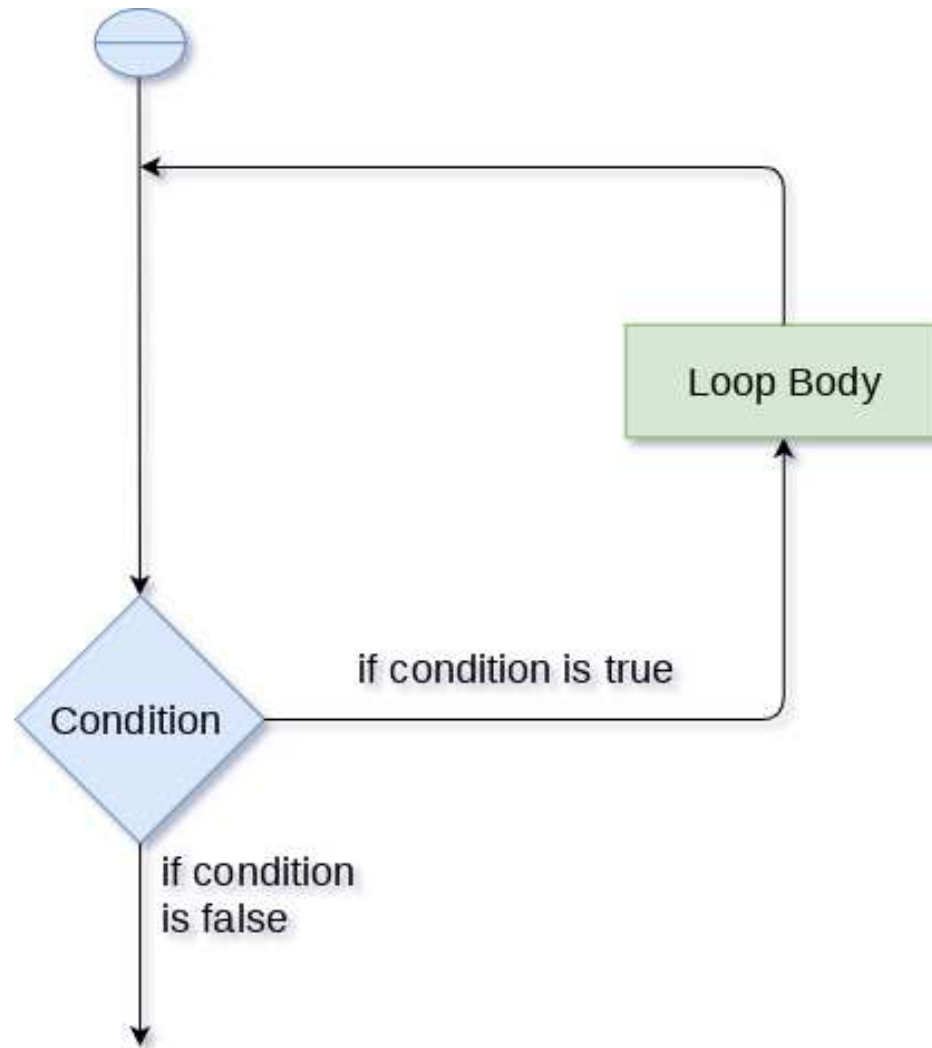
1. Operator Pembanding
2. Loop
3. While
4. Do While
5. For

Operator Perbandingan

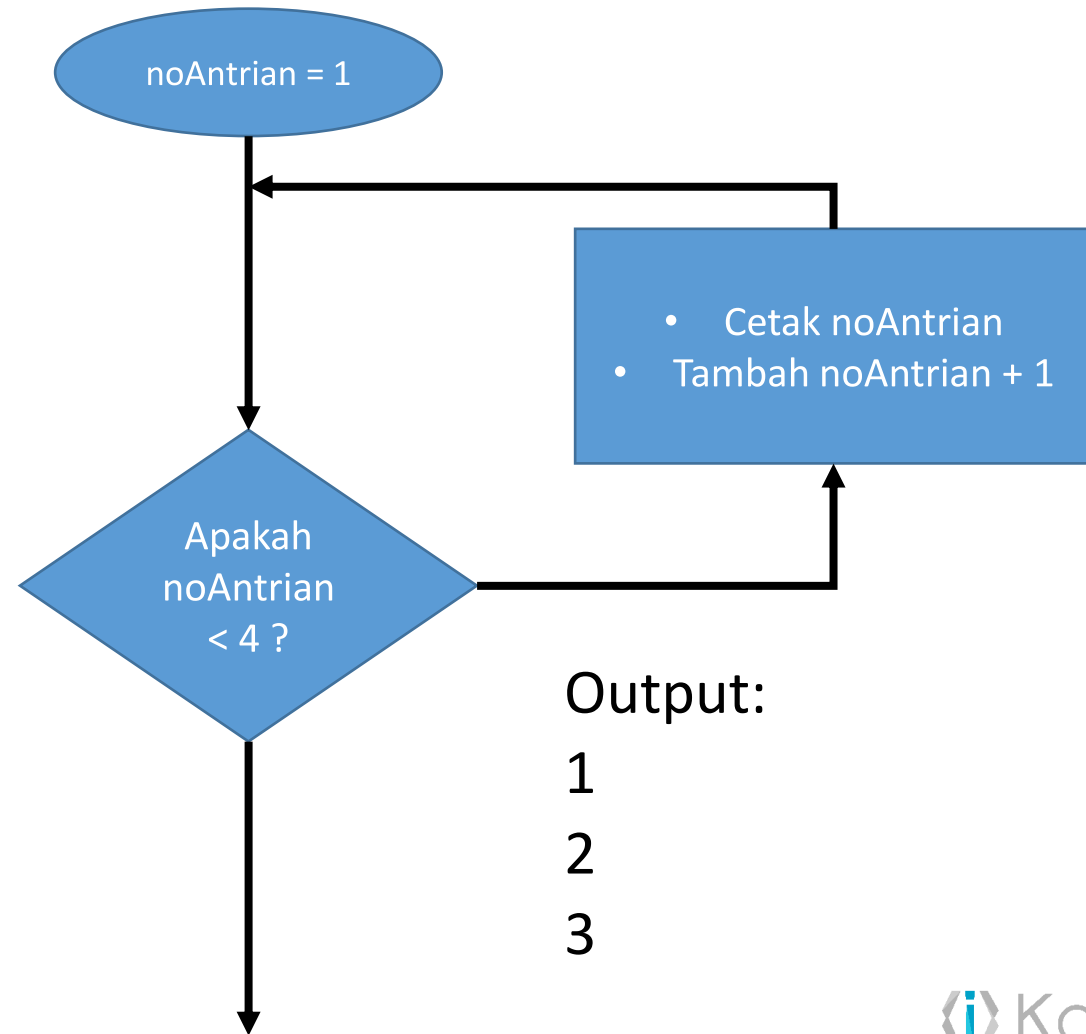
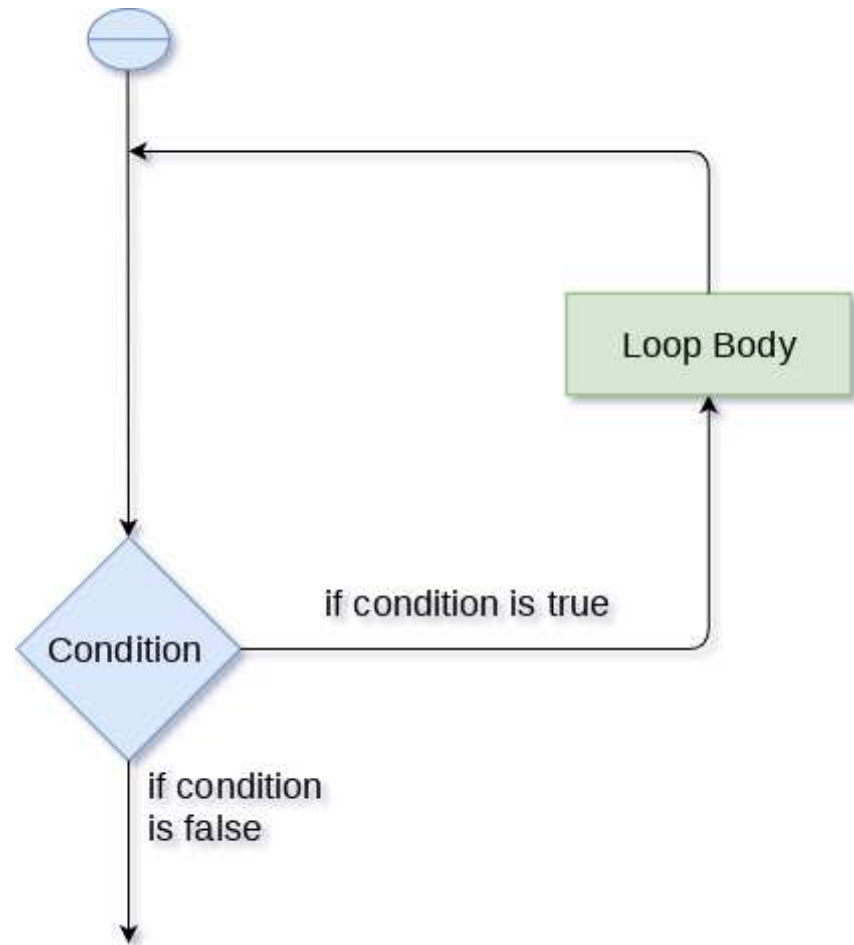
No	Operasi	Operator
1	Sama dengan	==
2	Lebih besar dari	>
3	Lebih kecil dari	<
4	Tidak sama dengan	!=
5	Lebih besar sama dengan	>=
6	Lebih kecil sama dengan	<=

Table disamping adalah operator perbandingan yang biasa digunakan dalam pembuatan suatu kondisi.

Loop



Loop



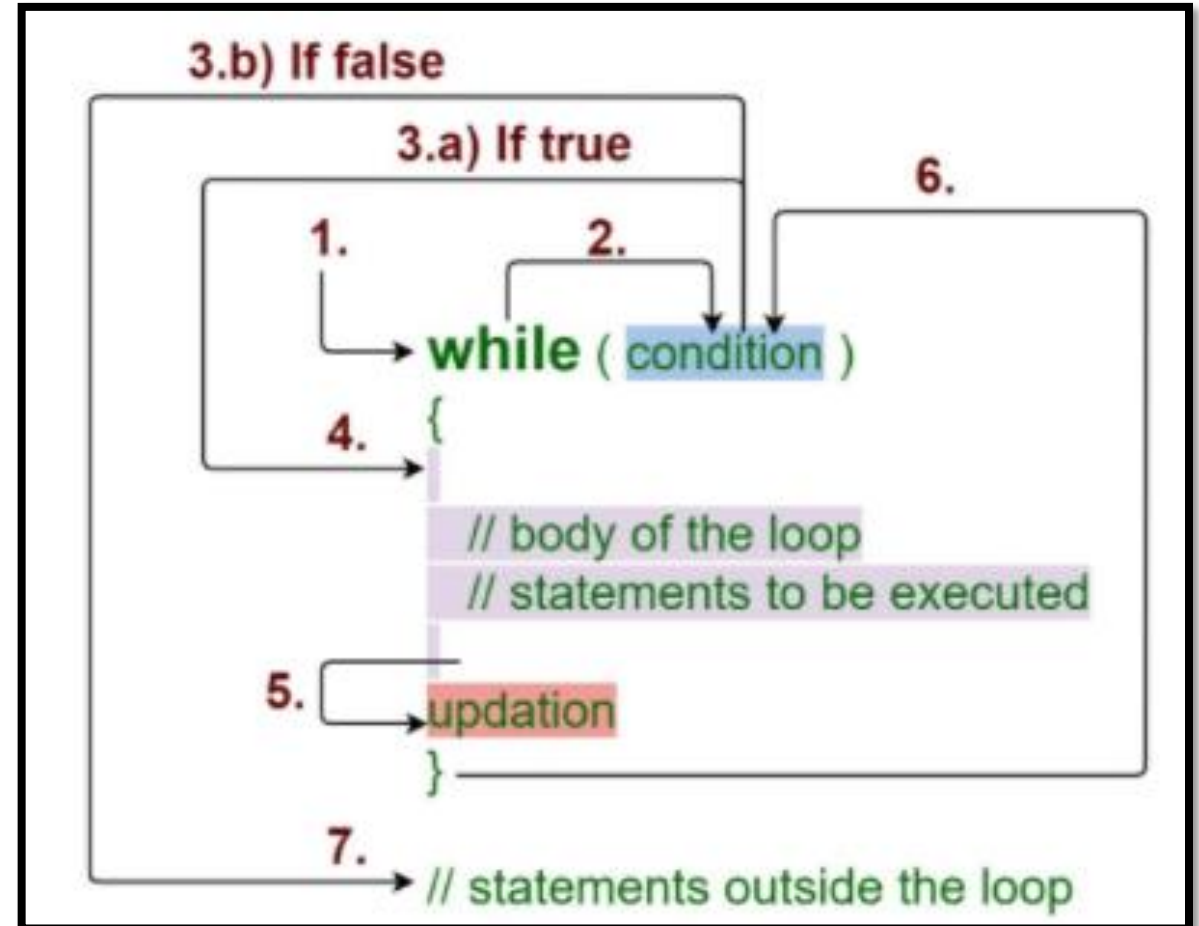
While

Perulangan while loop melalui blok kode selama kondisi yang ditentukan benar:

```
while (condition) {  
    // code block to be executed  
}
```

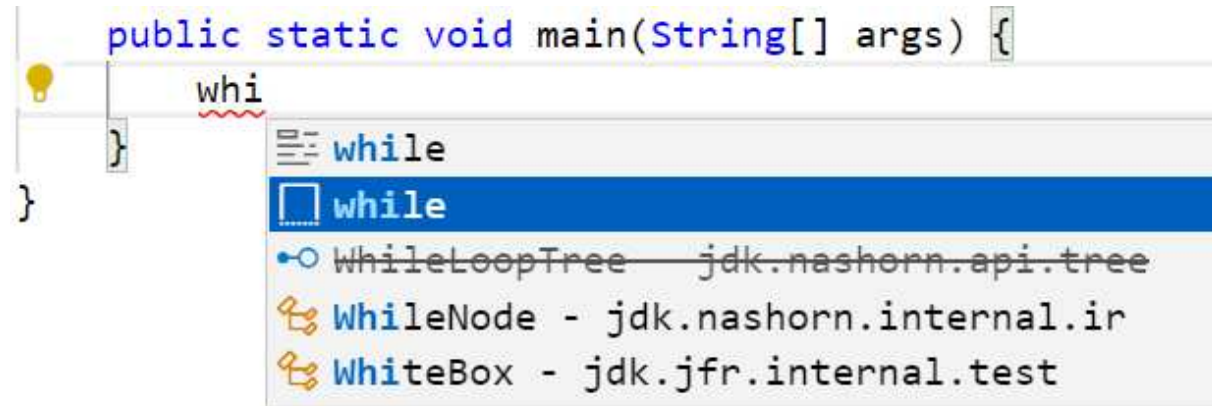
While - Flowchart

```
while (condition) {  
    // code block to be executed  
}
```



While – Pilihan Sintax Java

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



- Ketik while lalu ctrl + space
- Pilih yang while dengan icon kotak putih

While - Contoh

```
int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}
```

Blok kode yang ada didalamnya terdiri dari 2 kode yaitu

1. Cetak variable i
2. Tambah nilai variable i menjadi 1

While

Output:

0
1
2
3
4

```
public class KelasWhileSyntax {  
    Run | Debug  
    public static void main(String[] args) {  
        int i = 0;  
        while (i < 5) {  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

While

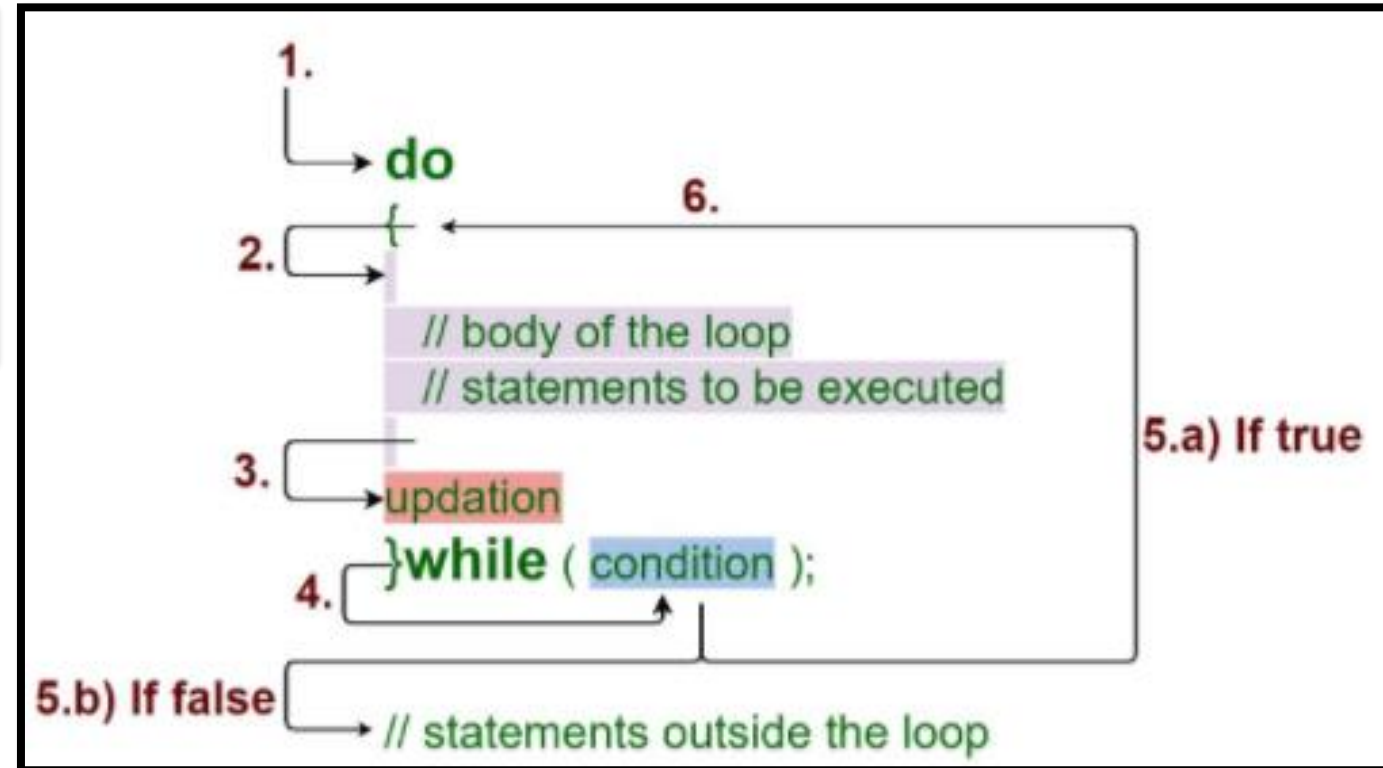
```
public class KelasWhileDasar2 {  
    Run | Debug  
    public static void main(String[] args) {  
        int noAntrian = 1;  
        while (noAntrian < 4) {  
            System.out.println(noAntrian);  
            noAntrian = noAntrian + 1;  
        }  
    }  
}
```

Do While

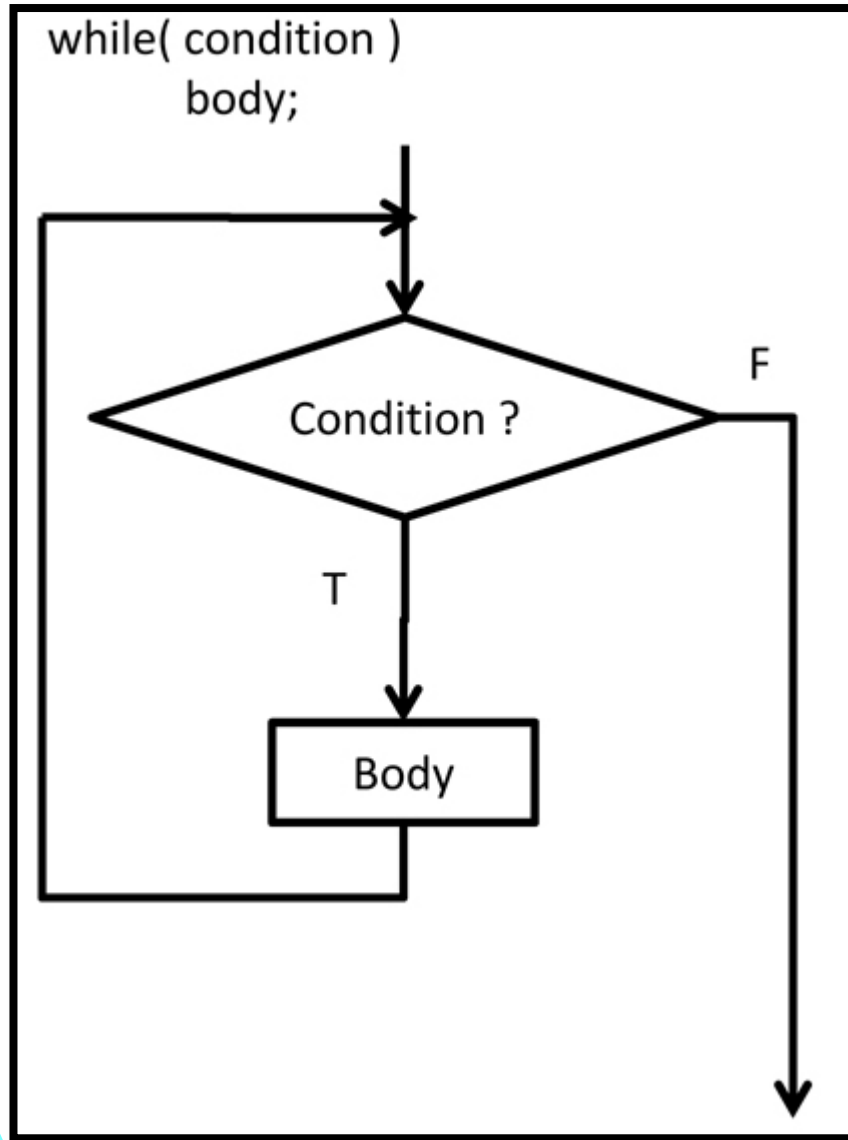
```
do {  
    // code block to be executed  
}  
while (condition);
```

Do While - Flowchart

```
do {  
    // code block to be executed  
}  
while (condition);
```

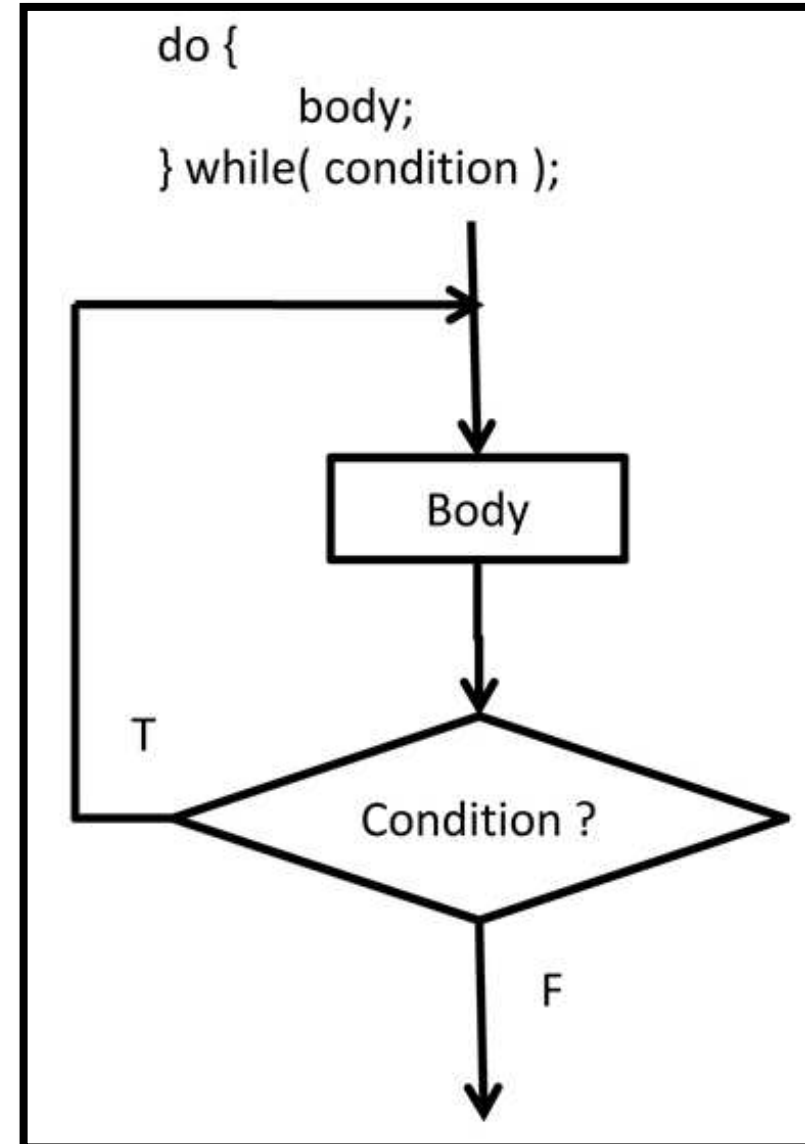


While



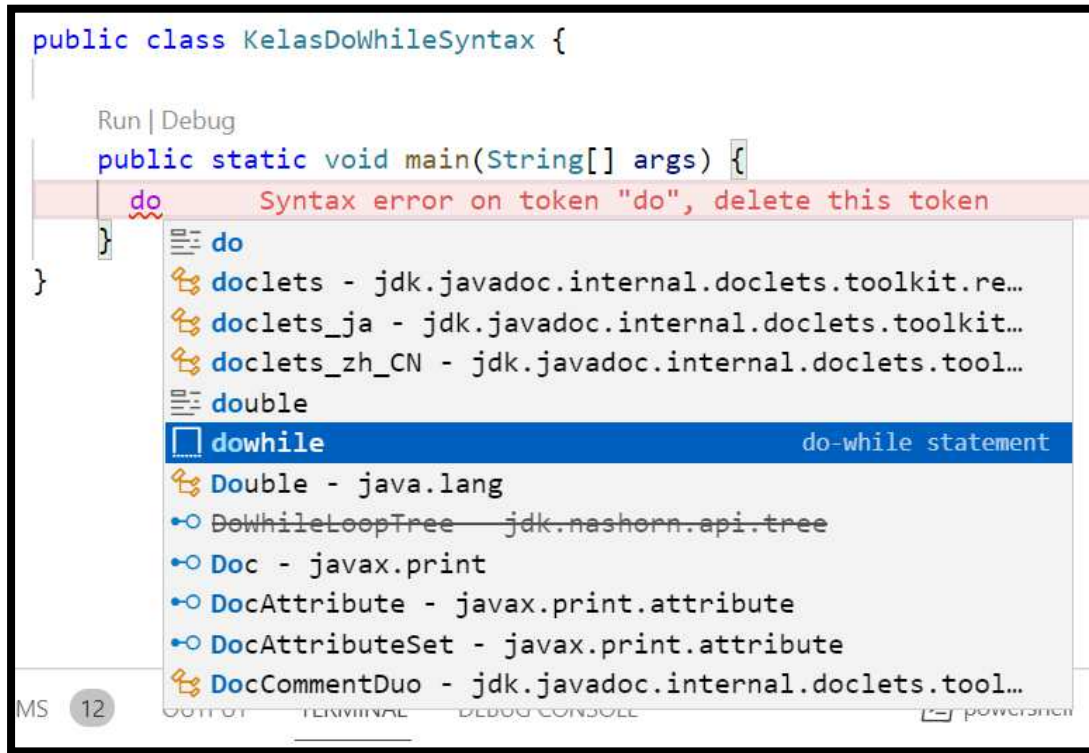
vs

Do While



Do While – Sintax Java

```
public class KelasDoWhileSyntax {  
    Run | Debug  
    public static void main(String[] args) {  
        do  
        Syntax error on token "do", delete this token  
    }  
}
```



The screenshot shows an IDE window with a Java class named 'KelasDoWhileSyntax'. The code is as follows:

```
public class KelasDoWhileSyntax {  
    Run | Debug  
    public static void main(String[] args) {  
        do  
        Syntax error on token "do", delete this token  
    }  
}
```

A red error message is displayed: "Syntax error on token 'do', delete this token". A dropdown menu is open, showing a list of suggestions:

- do
- doclets - jdk.javadoc.internal.doclets.toolkit.re...
- doclets_ja - jdk.javadoc.internal.doclets.toolkit...
- doclets_zh_CN - jdk.javadoc.internal.doclets.tool...
- double
- dowhile** do-while statement
- Double - java.lang
- DoWhileLoopTree - jdk.nashorn.api.tree
- Doc - javax.print
- DocAttribute - javax.print.attribute
- DocAttributeSet - javax.print.attribute
- DocCommentDuo - jdk.javadoc.internal.doclets.tool...

The 'dowhile' option is highlighted in blue. The IDE interface also shows 'MS 12' in the bottom left corner and 'powerShell' in the bottom right corner.

- Ketik Do lalu ctrl + space
- Klik do while statement

Do While – Sintax Java

```
public class KelasDoWhileSyntax {  
    Run | Debug  
    public static void main(String[] args) {  
        do {  
            } while (condition);  
    }  
}
```

Do While - Contoh

```
public class KelasDoWhileDasar {  
    Run | Debug  
    public static void main(String[] args) {  
        int i = 4;  
        do {  
            System.out.println(i);  
            i++;  
        } while (i < 7);  
    }  
}
```

Do While - Contoh

```
public class KelasDoWhileDasar2 {  
    Run | Debug  
    public static void main(String[] args) {  
        int noAntrian = 1;  
        do {  
            System.out.println(noAntrian);  
            noAntrian++;  
        } while (noAntrian < 4);  
    }  
}
```

For

- Saat kita tahu persis berapa kali kita ingin mengulang block code lebih direkomendasikan untuk menggunakan **for**.

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

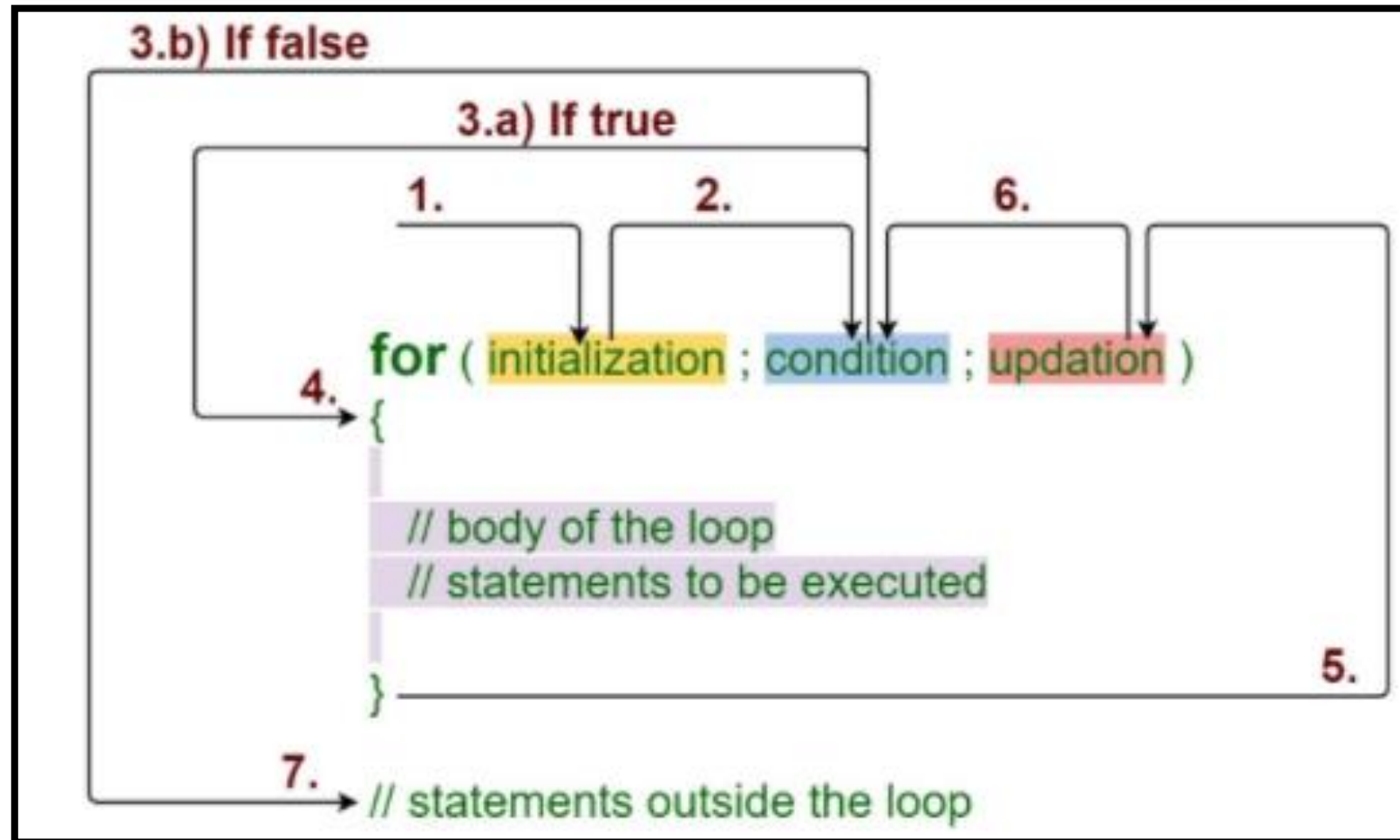
For

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

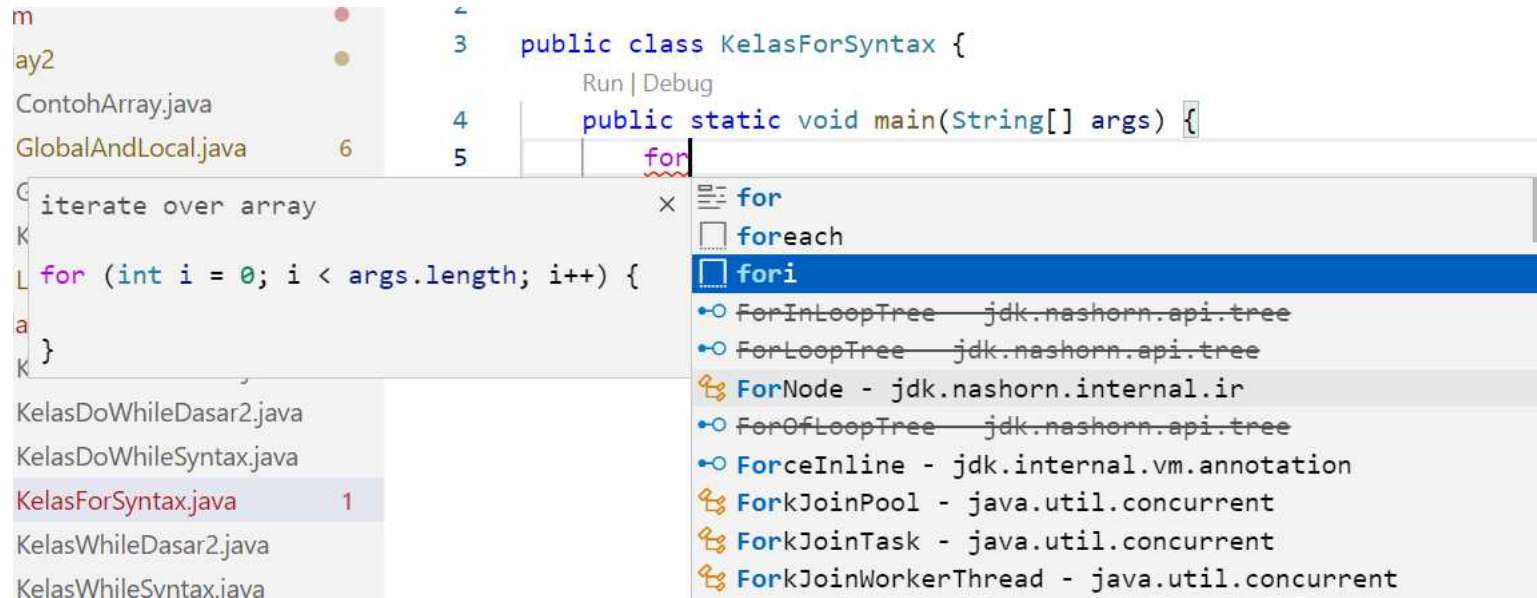
- Statement1 dieksekusi (satu kali) sebelum eksekusi blok kode.
- Statement2 mendefinisikan kondisi untuk mengeksekusi blok kode.
- Statement3 dieksekusi (setiap kali) setelah blok kode dijalankan.

For

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```



For – Sintax Java



```
3 public class KelasForSyntax {
4     public static void main(String[] args) {
5         for
```

Iterate over array

```
for (int i = 0; i < args.length; i++) {
}
```

- for
- foreach
- fori
- ForInLoopTree - jdk.nashorn.api.tree
- ForLoopTree - jdk.nashorn.api.tree
- ForNode - jdk.nashorn.internal.ir
- ForOfLoopTree - jdk.nashorn.api.tree
- ForceInline - jdk.internal.vm.annotation
- ForkJoinPool - java.util.concurrent
- ForkJoinTask - java.util.concurrent
- ForkJoinWorkerThread - java.util.concurrent

- Ketik for lalu ctrl + space
- Pilih for iterate over array

For – Sintax Java

```
public class KelasForSyntax {  
    Run | Debug  
    public static void main(String[] args) {  
        for (int i = 0; i < args.length; i++) {  
        }  
    }  
}
```

For – Sintax Java

```
for (int i = 0; i < args.length; i++) {  
  
}
```

```
// statement1;          statement2;          statement3  
for (int i = 0; i < args.length; i++) {  
    // blok kode untuk dieksekusi  
}
```

For – Sintax Java

```
// statement1;          statement2;          statement3
for (int i = 0;  i < args.length;          i++) {
    // blok kode untuk dieksekusi
}
```

- Statement1 → `int i = 0;` artinya nilai awal looping for dari angka 0
- Statement2 → `i < args.length;` artinya nilai akhir looping for hingga nilai `args.length` (`args.length` sendiri belum ditentukan)
- Statement3 → `i++;` artinya nilai looping akan terus bertambah 1

For - Contoh

```
public class KelasForSyntax {  
    Run | Debug  
    public static void main(String[] args) {  
        for (int i = 0; i < 4; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

Output :

0
1
2
3

For – Contoh 3

```
public class KelasForDasar2 {  
    Run | Debug  
    public static void main(String[] args) {  
        int antrian = 1;  
        for (int i = 0; i < 4; i++) {  
            System.out.println(antrian);  
            antrian +=1;  
        }  
    }  
}
```

Coba jelaskan mengapa outputnya seperti itu?

1
2
3
4

For – Contoh 4

```
public class KelasForDasar4 {  
    Run | Debug  
    public static void main(String[] args) {  
        for (int noAntrian = 1; noAntrian < 4; noAntrian++) {  
            System.out.println(noAntrian);  
        }  
    }  
}
```

Output :

1
2
3

Coba jelaskan mengapa outputnya seperti itu?

For – Contoh 5

```
public class KelasForDasar5 {  
    Run | Debug  
    public static void main(String[] args) {  
        int noAntrian = 1;  
  
        for (int i = noAntrian; i < 4; i++) {  
            System.out.println(noAntrian);  
            noAntrian+=2;  
        }  
    }  
}
```

Coba jelaskan
mengapa outputnya
seperti itu?

1
3
5

For – Contoh 6

```
public class KelasForDasar5 {  
    Run | Debug  
    public static void main(String[] args) {  
        int noAntrian = 1;  
  
        for (int i = noAntrian; i < 4; i++) {  
            System.out.println(noAntrian);  
            noAntrian++;  
        }  
    }  
}
```

Output :

1
2
3

Coba jelaskan
mengapa outputnya
seperti itu?

Input Console - Scanner

```
public class KelasInputConsole {
```

Run | Debug

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

scan

- ScanEnvironment - jdk.nashorn.internal.runtime.re...
- Scanner - java.util** java.util.Scanner
- Scanner - jdk.nashorn.internal.parser
- ScrollPaneBorder - javax.swing.plaf.metal.MetalBo...
- ScrollPaneConstants - javax.swing
- ScrollPaneLayout - javax.swing
- ScrollPaneUI - javax.swing.plaf
- ScatteringByteChannel - java.nio.channels
- SaslClient - javax.security.sasl
- SaslClientFactory - javax.security.sasl
- SourceCodeAnalysis - jdk.jshell
- SchedulingAction - jdk.internal.net.http.common.S...

Input Console - Scanner

```
public class KelasInputConsole {
```

Run | Debug

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

```
Scanner scan = new Scanner(System.in);
```

Object Scanner

```
System.out.println("PENDAFTARAN PESERTA KODEHIVE ACADEMY");
```

```
System.out.print("Masukkan Nama : ");
```

```
String nama = scan.nextLine();
```

Baca Inputan String

```
System.out.print("Masukkan Alamat : ");
```

```
String alamat = scan.nextLine();
```

```
System.out.print("Masukkan Usia : ");
```

```
Integer usia = scan.nextInt();
```

Baca Inputan Int

```
System.out.println();
```

```
System.out.println("BIODATA PESERTA");
```

```
System.out.println("-----");
```

```
System.out.println("Nama Peserta : "+ nama);
```

```
System.out.println("Alamat Peserta : "+ alamat);
```

```
System.out.println("Usia Pesera : "+ usia);
```

Output

PENDAFTARAN PESERTA KODEHIVE ACADEMY

Masukkan Nama : Gusta

Masukkan Alamat : Jatim

Masukkan Usia : 26

Input

BIODATA PESERTA

Nama Peserta : Gusta

Alamat Peserta : Jatim

Usia Pesera : 26

Output