

BAHASA PEMROGRAMAN JAVA

Pertemuan IV

STRUKTUR KONTROL PERULANGAN

Oleh
Achmad Arrosyidi



TUJUAN PEMBELAJARAN

Mahasiswa dapat mendemonstrasikan pembuatan struktur kontrol perulangan tunggal.



Perulangan Tunggal:

1. Struktur Kontrol Perulangan Tunggal:
 1. For
 2. While
 3. Do while
2. Pernyataan Khusus:
 1. Break
 2. Continue
 3. Return
3. Latihan



1. STRUKTUR KONTROL PERULANGAN

- Struktur kontrol percabangan khusus, salah satu alurnya kembali ke posisi semula → berulang menurut kondisi tertentu.
- Perulangan tunggal:
 1. For
 2. While
 3. Do-while.
- Kategorisasi Perulangan
 1. Tunggal
 2. Majemuk / bersarang (Nested)



1.1. PERULANGAN FOR (1)

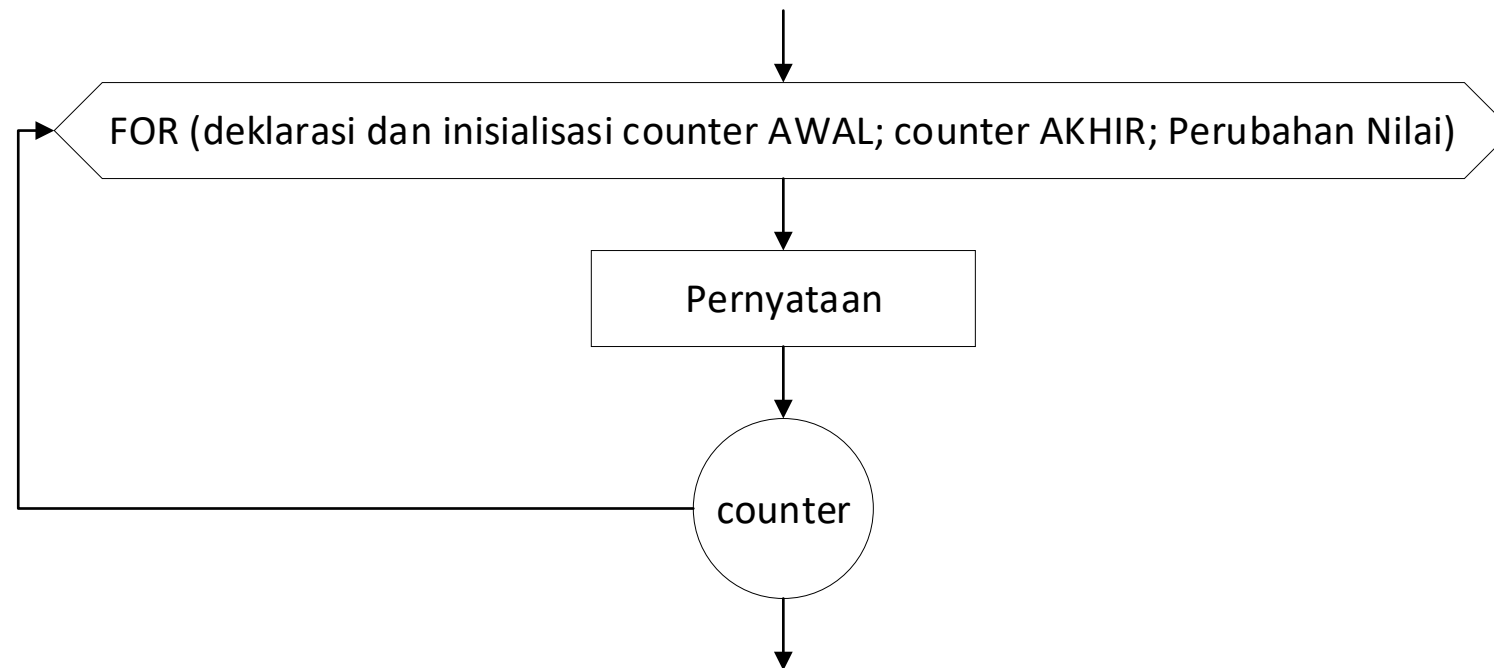
- Variabel counter → Deklarasi → inisialisasi → menentukan nilai **AWAL** s/d **AKHIR**
- **Penentu** perulangan → selisih **counter** (counter awal s/d counter akhir), dan **jangka** perubahan nilai → di**AWAL** perulangan.
- Variabel perubahan nilai → bilangan **bulat**.



1.1. PERULANGAN FOR (2)

Syntax:

```
for (deklarasi dan inisialisasi AWAL; AKHIR; Perubahan Nilai) {  
    pernyataan  
}
```



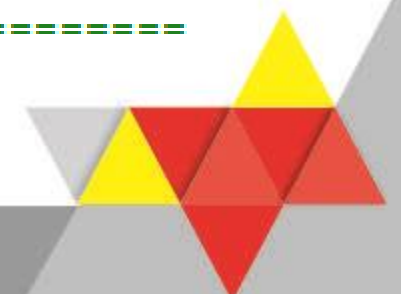
1.1. PERULANGAN FOR (3)

- Contoh perulangan FOR dengan Increment Maju:

```
1 //File: Program_3_6.java
2 import java.io.*;
3 public class Program_3_6 {
4     public static void main (String[] args) {
5         int qty=0;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ")
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             for(int i=1; i<=qty; i++){
12                 System.out.println("cetak halo ke-" + i);
13             }
14         }
15         catch (Exception e){
16             System.out.println("Masukan harus angka");
17         }
18     }
19 }
20
21
22 }
```

Console

```
CD: D:\Buku Java\2 - Program\Baru\Bab 3
Current directory: D:\Buku Java\2 - Program\Baru\Bab 3
java Program_3_6
Process started (PID=1820) >>>
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-1
cetak halo ke-2
cetak halo ke-3
cetak halo ke-4
<<< Process finished (PID=1820). (Exit code 0)
===== READY =====
```



1.1. PERULANGAN FOR (4)

- Contoh perulangan FOR dengan Decrement Mundur:

```
1 //File: Program_3_6_Decrement.java
2 import java.io.*;
3 public class Program_3_6_Decrement {
4     public static void main (String[] args) {
5         int qty=0;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             for(int i=qty; i>=1; i--){
12                 System.out.println("cetak halo ke-" + i);
13             }
14         }
15         catch (Exception e){
16             System.out.println("Masukan harus angka");
17         }
18     }
19 }
20
21
22
23
```

Console

```
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-4
cetak halo ke-3
cetak halo ke-2
cetak halo ke-1
<<< Process finished (PID=2712). (Exit code 0)
===== READY =====
```



1.1. PERULANGAN FOR STEP CUSTOM (5)

- Contoh perulangan FOR dengan step custom Maju:

```
1 //Perulangan_For_step_custom.java
2 import java.io.*;
3 public class Perulangan_For_step_custom {
4     public static void main (String[] args) {
5         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
6         System.out.print("Masukkan angka untuk menentukan step dalam looping: ");
7
8         try {
9             int step = Integer.parseInt(tampung.readLine());
10
11             for (int i=1; i<=10; i=i+step) {
12                 System.out.println("Cetak nilai i: " + i);
13             }
14
15             catch (Exception e){
16                 System.out.println("Masukan harus angka");
17             }
18
19     }
20 }
```

Console

Masukkan angka untuk menentukan step dalam looping: 5

Cetak nilai i: 1

Cetak nilai i: 6

<<< Process finished (PID=14988). (Exit code 0)

===== READY =====



1.1. PERULANGAN FOR STEP CUSTOM (6)

- Contoh perulangan FOR dengan step custom Mundur:

```
1 //Perulangan_For_step_custom_mundur.java
2 import java.io.*;
3 public class Perulangan_For_step_custom_mundur {
4     public static void main (String[] args) {
5         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
6         System.out.print("Masukkan angka untuk menentukan step dalam looping: ");
7
8         try {
9             int step = Integer.parseInt(tampung.readLine());
10
11             for (int i=10; i>=1; i=i-step) {
12                 System.out.println("Cetak nilai i: " + i);
13             }
14
15         } catch (Exception e){
16             System.out.println("Masukan harus angka");
17         }
18     }
19 }
20 }
```

Console

```
Process started (PID=13036) >>>
Masukkan angka untuk menentukan step dalam looping: 3
Cetak nilai i: 10
Cetak nilai i: 7
Cetak nilai i: 4
Cetak nilai i: 1
<<< Process finished (PID=13036). (Exit code 0)
===== READY =====
```



1.2. PERULANGAN WHILE (1)

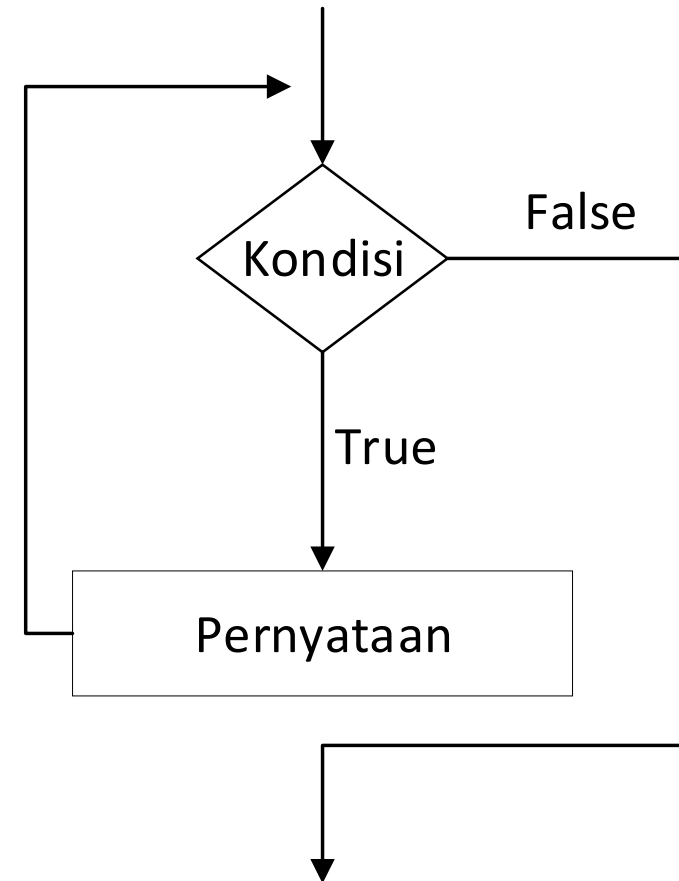
- Terdapat **ekspresi logika**.
- Penentuan perulangan → hasil ekspresi/kondisi logika.
- Perubahan **counter** → **dalam** perulangan.
- Perulangan → memeriksa ekspresi di **AWAL** pernyataan.
- **Tidak menjamin** perulangan akan dilakukan walaupun hanya **satu** kali
- **Jika Pertama** kali hasil pemeriksaan logika = **false** → **tidak** menjalankan pernyataan.



1.2. PERULANGAN WHILE (2)

Syntax:

```
while (kondisi logika) {  
    pernyataan  
}
```



1.2. PERULANGAN WHILE (3)

- Contoh perulangan WHILE:

```
1 //File: Program_3_7.java
2 import java.io.*;
3 public class Program_3_7 {
4     public static void main (String[] args) {
5         int qty=0, counter=1;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             while (counter<qty+1){
12                 System.out.println("cetak halo ke-" + counter);
13                 counter++;
14             }
15         }
16         catch (Exception e){
17             System.out.println("Masukan harus angka");
18         }
19     }
20 }
21 }
```

Console

```
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-1
cetak halo ke-2
cetak halo ke-3
cetak halo ke-4
<<< Process finished (PID=9908). (Exit code 0)
===== READY =====
```



1.3. PERULANGAN DO WHILE (1)

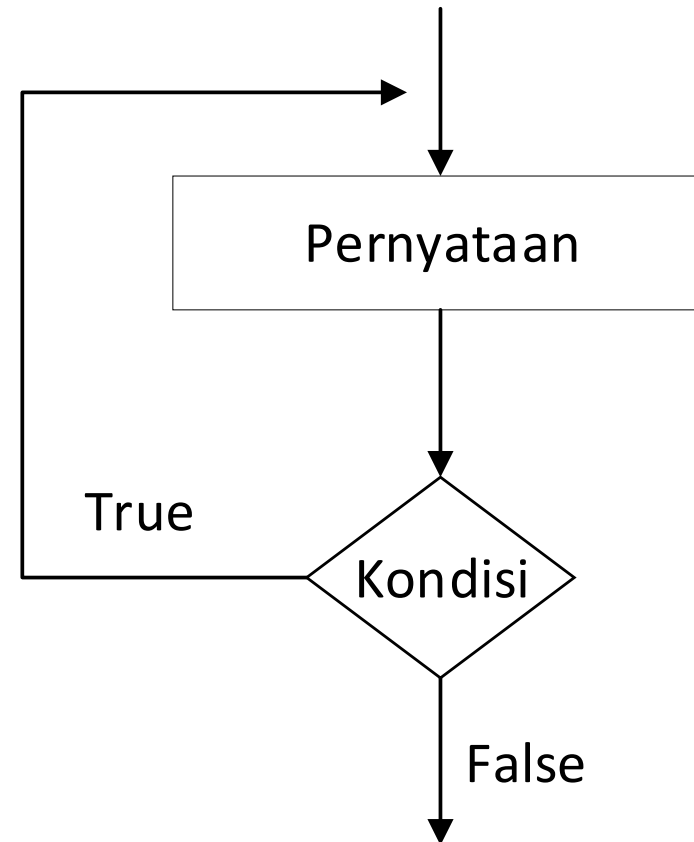
- Terdapat **ekspresi logika**.
- **Penentuan** perulangan → hasil ekspresi/**kondisi** logika.
- Perubahan **counter** → **dalam** perulangan.
- Perulangan → **memeriksa** ekspresi di**AKHIR** pernyataan.
- **Menjamin** perulangan akan dilakukan walaupun hanya **satu** kali
- **Pertama** kali **tidak** melakukan pemeriksaan logika → menjalankan pernyataan.



1.3. PERULANGAN DO WHILE (2)

Syntax:

```
do {  
    pernyataan  
} while (kondisi logika)
```



1.3. PERULANGAN DO WHILE (3)

- Contoh perulangan DO WHILE:

```
1 //File: Program_3_8.java
2 import java.io.*;
3 public class Program_3_8 {
4     public static void main (String[] args) {
5         int qty=0, counter=1;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             do {
12                 System.out.println("cetak halo ke-" + counter);
13                 counter++;
14             } while (counter<qty+1);
15         }
16         catch (Exception e){
17             System.out.println("Masukan harus angka");
18         }
19     }
20 }
21
22
23 }
```

Console

```
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-1
cetak halo ke-2
cetak halo ke-3
cetak halo ke-4
<<< Process finished (PID=6360). (Exit code 0)
===== READY =====
```



2. PERNYATAAN KHUSUS

- Sebuah pernyataan yang didalam terdapat kata kunci / keyword khusus.
- Keyword mempunyai kegunaan untuk menentukan alur kontrol perulangan.
- Terdapat pernyataan khusus:
 1. Break
 2. Continue
 3. Return



2.1. PERNYATAAN KHUSUS: BREAK (1)

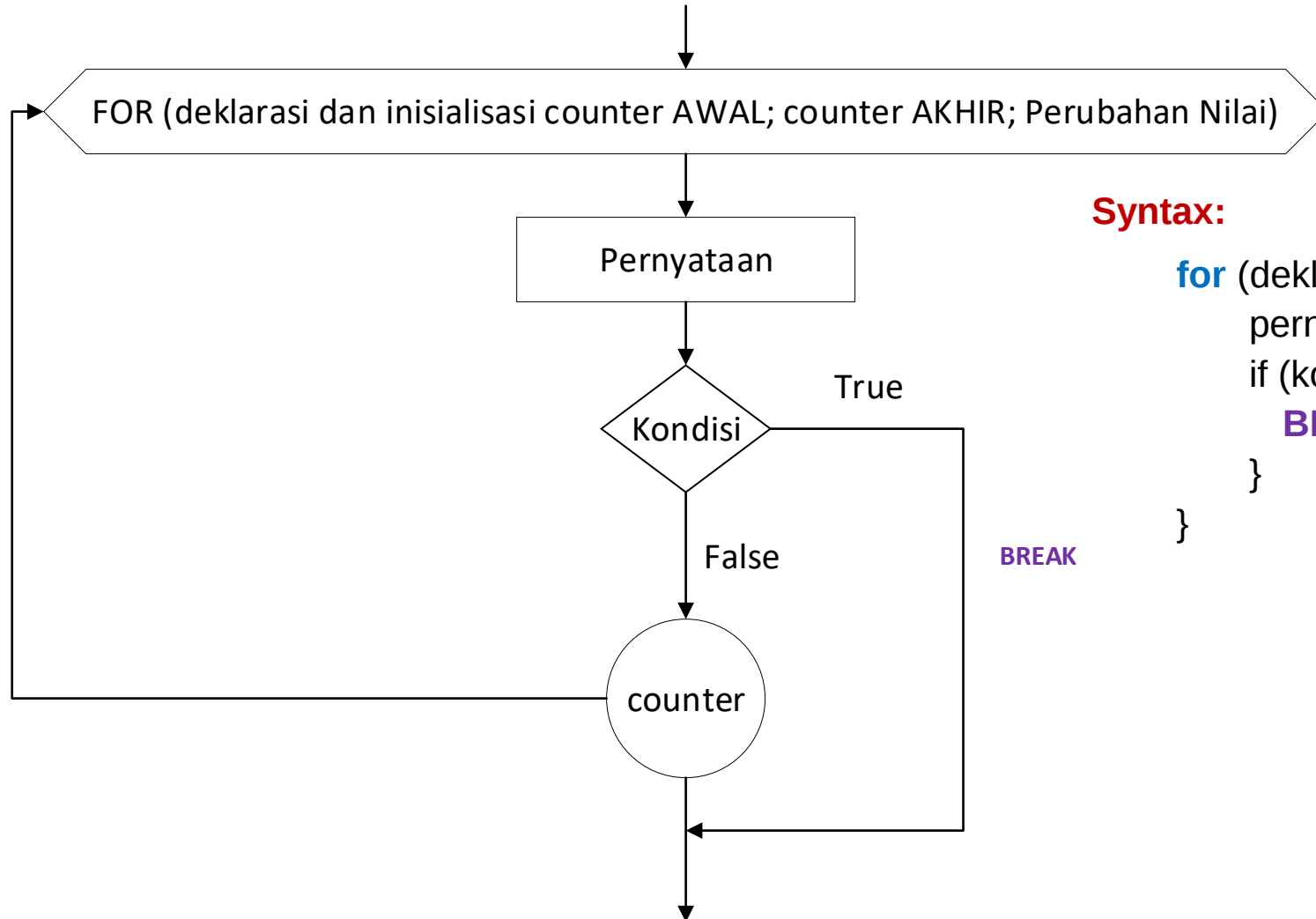
- Pernyataan Break → menghentikan sengaja proses perulangan → kondisi tertentu → perulangan tidak sepenuhnya dikerjakan.
- 2 jenis Break:
 1. Break Unlabeled
 2. Break Labeled



2.1.1. PERNYATAAN BREAK UNLABELED

- Struktur pernyataan break unlabeled:
 1. **Bawaan** struktur program (built-in) → **Switch-Case**
 2. **Tambahan** struktur program → **For, While dan Do-While.**
- Ciri dari Break unlabeled adalah kode **Break** → **Tanpa** diikuti oleh kode program atau **script Java**.

2.1.1.A. PERNYATAAN BREAK UNLABELED DALAM FOR (1)



Syntax:

```
for (deklarasi dan inisialisasi AWAL; AKHIR; Perubahan Nilai) {  
    pernyataan  
    if (kondisi logika) {  
        BREAK  
    }  
}
```



2.1.1.A. PERNYATAAN BREAK UNLABELED DALAM FOR (2)

- Contoh BREAK UNLABELED dalam FOR:

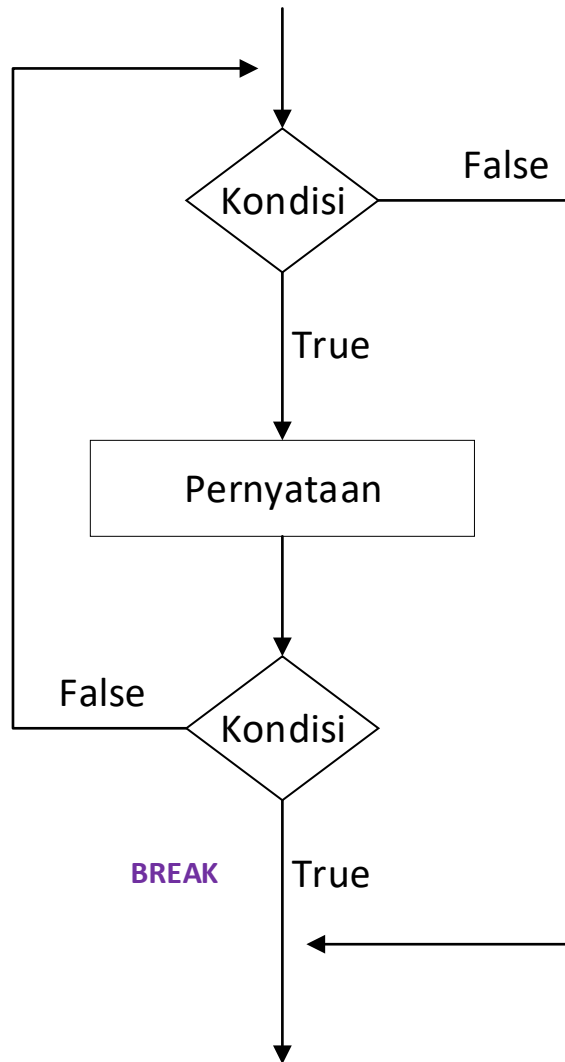
```
1 //File: Program_3_12.java
2 import java.io.*;
3 public class Program_3_12 {
4     public static void main (String[] args) {
5         int qty=0;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             for(int i=1; i<=qty; i++){
12                 System.out.println("cetak halo ke-" + i);
13                 if (i==2) {
14                     break;
15                 }
16             }
17         }
18         catch (Exception e){
19             System.out.println("Masukan harus angka");
20         }
21     }
22 }
23 }
```

Console

```
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-1
cetak halo ke-2
<<< Process finished (PID=2372). (Exit code 0)|
===== READY =====
```



2.1.1.B. PERNYATAAN BREAK UNLABELED DALAM WHILE (1)



Syntax:

```
while (kondisi logika) {  
    pernyataan  
    if (kondisi logika) {  
        BREAK  
    }  
}
```



2.1.1.B. PERNYATAAN BREAK UNLABELED DALAM WHILE (2)

- Contoh BREAK UNLABELED dalam WHILE:

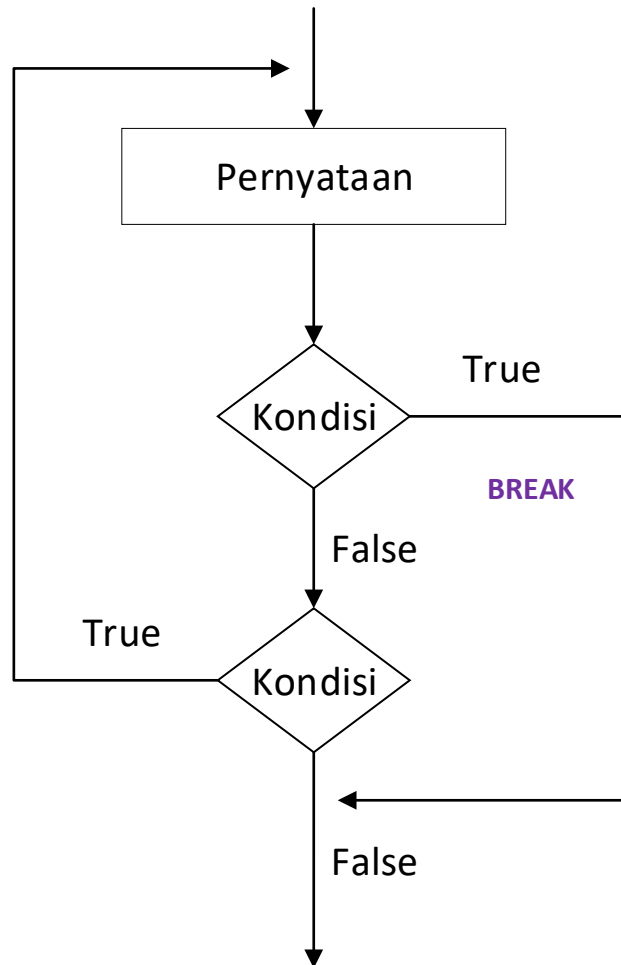
```
1 //File: Program_3_13.java
2 import java.io.*;
3 public class Program_3_13 {
4     public static void main (String[] args) {
5         int qty=0, counter=1;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             while (counter<qty+1){
12                 System.out.println("cetak halo ke-" + counter);
13                 counter++;
14                 if (counter == 2) {
15                     break;
16                 }
17             }
18         }
19         catch (Exception e){
20             System.out.println("Masukan harus angka");
21         }
22     }
23 }
24 }
```

Console

```
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-1
<<< Process finished (PID=4080). (Exit code 0)
===== READY =====
```



2.1.1.C. PERNYATAAN BREAK UNLABELED DALAM DO WHILE (1)



Syntax:

```
do {  
    pernyataan  
} while (kondisi logika)
```



2.1.1.C. PERNYATAAN BREAK UNLABELED DALAM DO WHILE (2)

- Contoh BREAK UNLABELED dalam DO WHILE:

```
1 //File: Program_3_14.java
2 import java.io.*;
3 public class Program_3_14 {
4     public static void main (String[] args) {
5         int qty=0, counter=1;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             do {
12                 System.out.println("cetak halo ke-" + counter);
13                 counter++;
14                 if (counter == 2);
15                 break;
16             } while (counter<qty+1);
17
18         }
19         catch (Exception e){
20             System.out.println("Masukan harus angka");
21         }
22     }
23 }
```

Console

```
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-1
<<< Process finished (PID=12648). (Exit code 0)
===== READY =====
```



2.1.2. PERNYATAAN BREAK LABELED

- Struktur pernyataan break labeled:
 1. **Bukan Bawaan** struktur program
 2. **Tambahan** struktur program → **For, While dan Do-While.**
- Ciri dari Break labeled adalah kode **Break** → **diikuti** oleh kode program atau **script Java.**



2.1.2. PERNYATAAN BREAK LABELED DALAM FOR

- Contoh BREAK LABELED dalam FOR:

```
1 //File: Program_3_15.java
2 class Program_3_15 {
3     public static void main(String[] args) {
4
5         int[][] arrayOfInts = {
6             { 32, 87, 3, 589 },
7             { 12, 1076, 2000, 8 },
8             { 622, 127, 77, 955 }
9         };
10        int searchfor = 12;
11
12        int i;
13        int j = 0;
14        boolean foundIt = false;
15
```

Console

Found 12 at 1, 0

<<< Process finished (PID=3180). (Exit code 0)

===== READY =====

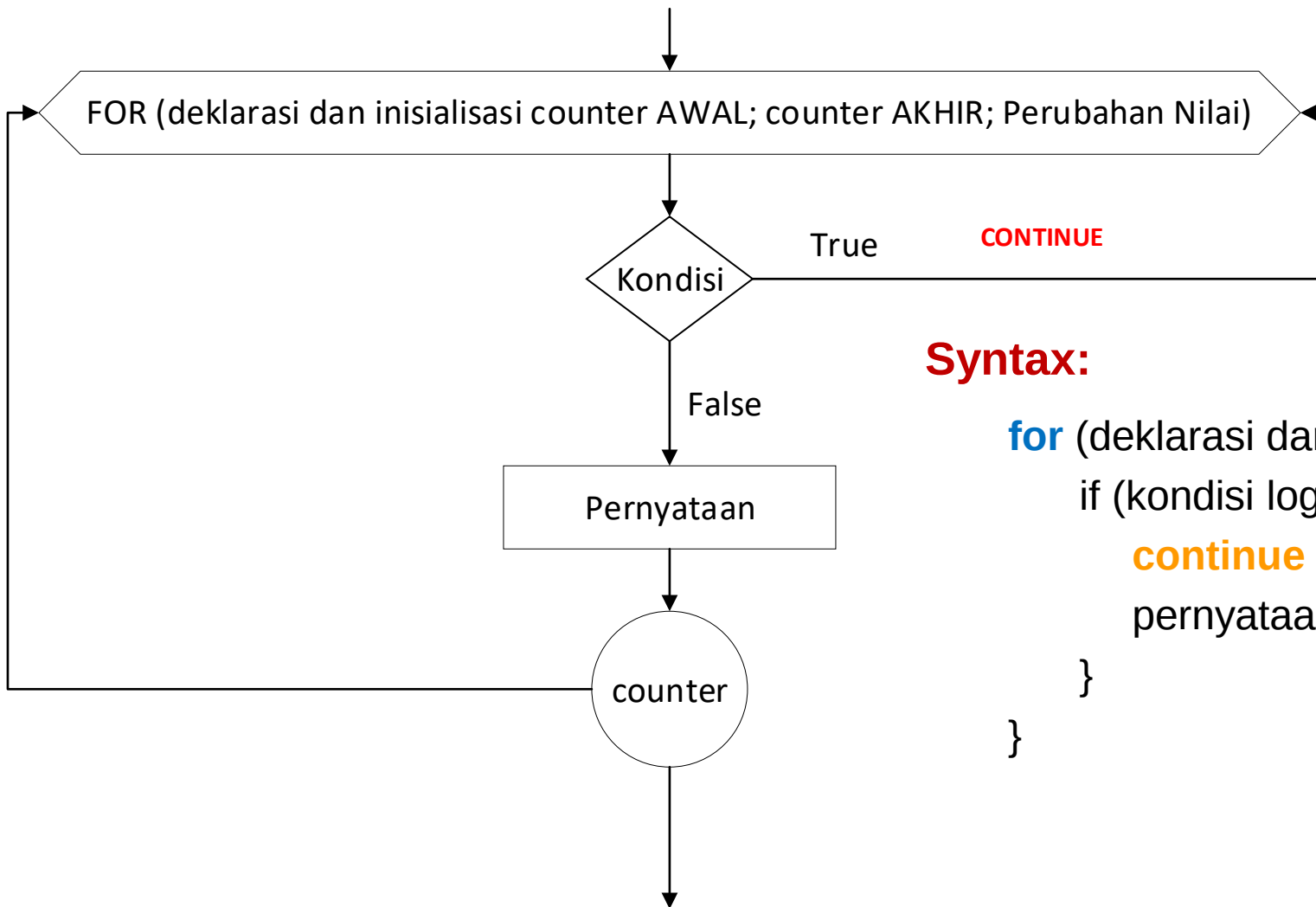
```
15
16        search:
17        for (i = 0; i < arrayOfInts.length; i++) {
18            for (j = 0; j < arrayOfInts[i].length;
19                j++) {
20                if (arrayOfInts[i][j] == searchfor) {
21                    foundIt = true;
22                    break search;
23                }
24            }
25        }
26
27        if (foundIt) {
28            System.out.println("Found " + searchfor + " at " + i + ", " + j);
29        } else {
30            System.out.println(searchfor + " not in the array");
31        }
32    }
33 }
```

2.2. PERNYATAAN CONTINUE

- Jika hasil **ekspresi** bernilai **True** → **Mengabaikan** proses perulangan
- Jika hasil **ekspresi** bernilai **False** akan **melanjutkan** proses perulangan.



2.2.A. PERNYATAAN CONTINUE DALAM FOR (1)



Syntax:

```
for (deklarasi dan inisialisasi AWAL; AKHIR; Perubahan Nilai) {  
    if (kondisi logika) {  
        continue  
    }  
    pernyataan  
}
```



2.2.A. PERNYATAAN CONTINUE DALAM FOR (2)

- Contoh CONTINUE dalam FOR:

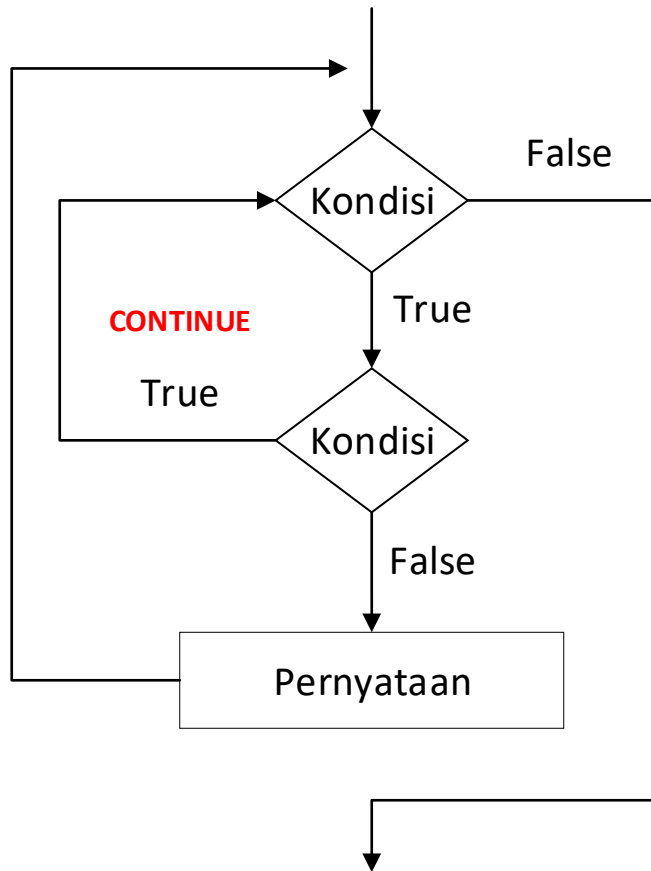
```
1 //File: Program_3_16.java
2 import java.io.*;
3 public class Program_3_16 {
4     public static void main (String[] args) {
5         int qty=0;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             for(int i=1; i<=qty; i++){
12                 if (i==2)
13                     continue;
14                 System.out.println("cetak halo ke-" + i);
15             }
16         }
17         catch (Exception e){
18             System.out.println("Masukan harus angka");
19         }
20     }
21 }
22 }
```

Console

```
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-1
cetak halo ke-3
cetak halo ke-4
<<< Process finished (PID=7164). (Exit code 0)
===== READY =====
```



2.2.B. PERNYATAAN CONTINUE DALAM WHILE (1)



Syntax:

```
while (kondisi logika) {  
    if (kondisi logika) {  
        continue  
    }  
    pernyataan  
}
```



2.2.B. PERNYATAAN CONTINUE DALAM WHILE (2)

- Contoh CONTINUE dalam WHILE:

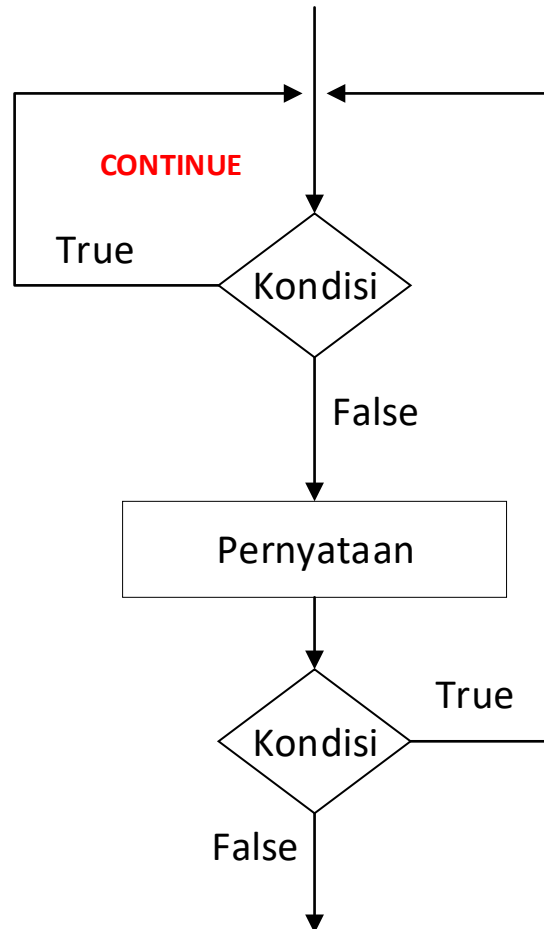
```
1 //File: Program_3_17.java
2 import java.io.*;
3 public class Program_3_17 {
4     public static void main (String[] args) {
5         int qty=0, counter=1;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             while (counter<qty+1){
12                 counter++;
13                 if (counter==2)
14                     continue;
15                 System.out.println("cetak halo ke-" + counter);
16             }
17         }
18         catch (Exception e){
19             System.out.println("Masukan harus angka");
20         }
21     }
22 }
23 }
```

Console

```
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-3
cetak halo ke-4
cetak halo ke-5
<<< Process finished (PID=13004). (Exit code 0)
===== READY =====
```



2.2.C. PERNYATAAN CONTINUE DALAM DO WHILE (1)



Syntax:

```
do {  
    if (kondisi logika) {  
        continue  
        pernyataan  
    }  
} while (kondisi logika)
```



2.2.C. PERNYATAAN CONTINUE DALAM DO WHILE (2)

- Contoh CONTINUE dalam DO WHILE:

```
1 //File: Program_3_18.java
2 import java.io.*;
3 public class Program_3_18 {
4     public static void main (String[] args) {
5         int qty=0, counter=1;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan angka untuk mencetak tulisan halo secara berulang: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             do {
12                 counter++;
13                 if (counter==2)
14                     continue;
15                 System.out.println("cetak halo ke-" + counter);
16             } while (counter<qty+1);
17
18         }
19         catch (Exception e){
20             System.out.println("Masukan harus angka");
21         }
22     }
23 }
```

Console

```
Masukkan angka untuk mencetak tulisan halo secara berulang: 4
cetak halo ke-3
cetak halo ke-4
cetak halo ke-5
<<< Process finished (PID=13704). (Exit code 0)
===== READY =====
```

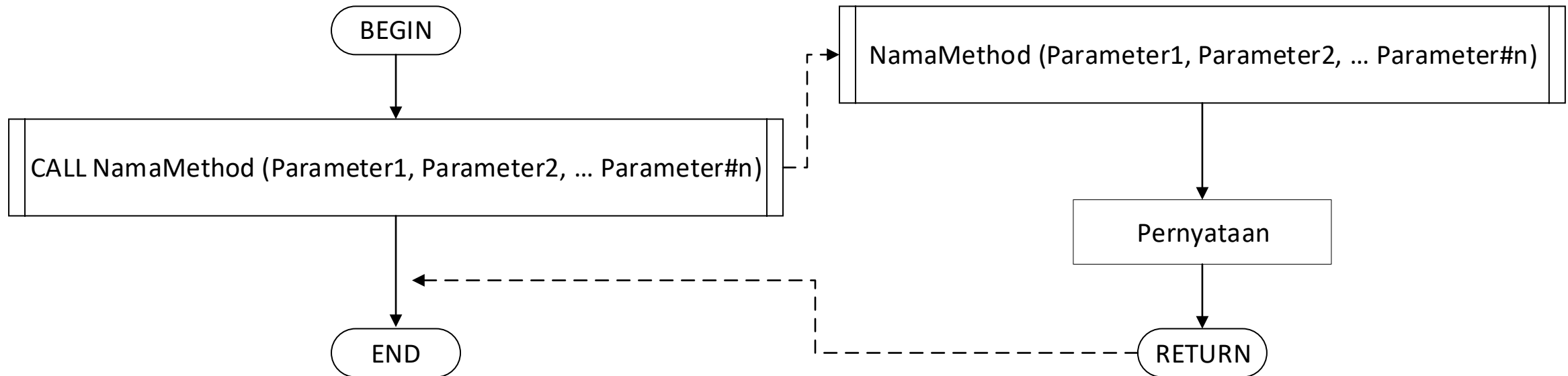


2.3. PERNYATAAN RETURN

- Untuk **kembali** kepada posisi instruksi **pemanggil**.
- Biasanya **Return** → pemanggilan sub proses atau **Method** dari program utama.
- 2 macam Pernyataan Return:
 1. Without Return Value
 2. With Return Value



2.3.1. PERNYATAAN RETURN WITHOUT RETURN VALUE (1)



Syntax:

Return()



2.3.1. PERNYATAAN RETURN WITHOUT RETURN VALUE (2)

- Contoh tanpa RETURN VALUE:

```
1 //File: Program 3_19.java
2 import java.io.*;
3 class Program_3_19 {
4     public static void main(String[] args) {
5         int qty = 0;
6         BufferedReader tampung = new BufferedReader (new InputStreamReader(System.in));
7         System.out.print("Masukkan sebuah angka: ");
8
9         try {
10             qty = Integer.parseInt(tampung.readLine());
11             cetakhallo(qty);
12         }
13         catch (Exception e){
14             System.out.println("Masukan harus angka");
15         }
16     }
17
18     // Mendefinisikan method cetakhallo()
19     private static void cetakhallo(int n) {
20         if (n < 2) {
21             System.out.println("angka tidak boleh lebih kecil dari 2");
22             return; // keluar dari method tulis dan kembali ke pemanggil()
23         }
24
25         // statement ini dilakukan jika qty > 1
26         for (int i=1; i<n; i++) {
27             System.out.println("cetak hallo ke-" + i);
28         }
29     }
30 }
```

Console

Masukkan sebuah angka: 4

cetak hallo ke-1

cetak hallo ke-2

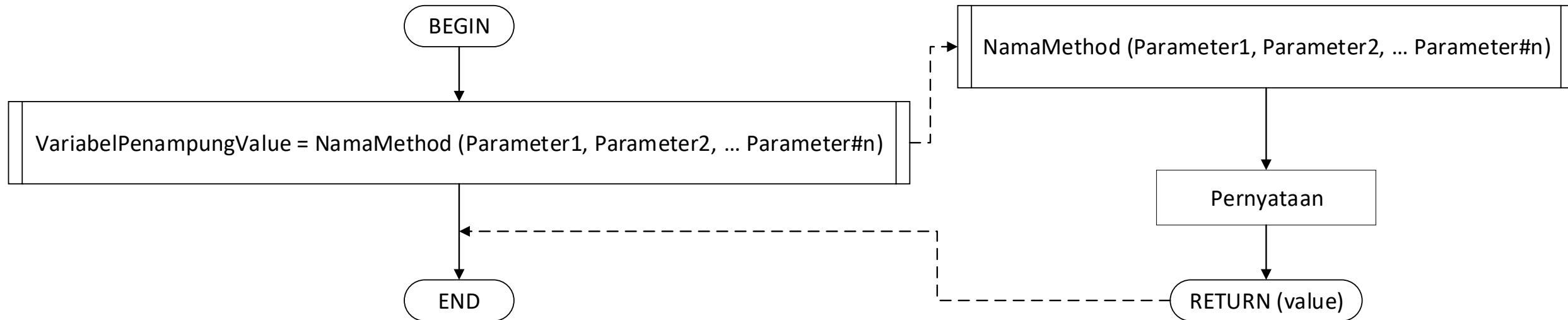
cetak hallo ke-3

<<< Process finished (PID=7512). (Exit code 0)

===== READY =====



2.3.2. PERNYATAAN RETURN WITH RETURN VALUE (1)



Syntax:

Return(value)



2.3.2. PERNYATAAN RETURN WITH RETURN VALUE (2)

- Contoh dengan RETURN VALUE:

```
1 //File: Program_3_20.java
2 import java.io.*;
3 class Program_3_20{
4     public static void main(String[] args) {
5         int pembilang, penyebut;
6         double hasilbagi;
7
8         BufferedReader tampung = new
9         BufferedReader (new InputStreamReader(
10         System.in));
```

Console

```
Masukkan angka pembilang: 4
Masukkan angka penyebut: 2
Hasil pembagian adalah: 2.0<<< Process finished (PID=4836). (Exit code 0)
===== READY =====
```

```
10     try {
11         System.out.print("Masukkan angka pembilang: ");
12         pembilang = Integer.parseInt(tampung.readLine());
13
14         System.out.print("Masukkan angka penyebut: ");
15         penyebut = Integer.parseInt(tampung.readLine());
16
17         if (penyebut==0) {
18             System.out.print("Penyebut tidak boleh bernilai Nol");
19         }
20         else {
21             hasilbagi=hitung(pembilang, penyebut);
22             System.out.print("Hasil pembagian adalah: " + hasilbagi);
23         }
24     }
25     catch (Exception e){
26         System.out.println("Masukan harus angka");
27     }
28 }
29
30 // Mendefinisikan method pembagian()
31 public static double hitung(int bilang, int sebut) {
32     double hasilhitung = (double) bilang/sebut;
33     return hasilhitung;
34 }
35 }
```

3. LATIHAN (1)

- Pada sebuah toko swalayan, terdapat promo baru yang ditawarkan untuk para ayam.
- Promo tersebut diberlakukan untuk pembelian shampoo dengan jumlah genap, minimal pembelian 2 botol shampoo akan diberikan gratis 1 botol shampoo.
- Promo ini berlaku untuk kelipatan.
- Buatlah daftar untuk mencatat jumlah shampoo gratis yang harus diberikan kepada ayam jika terdapat batasan maksimal pembelian shampoo.

2 botol	4 botol	6 botol	8 botol	10 botol	12 botol	14 botol	16 botol	18 botol	20 botol
1 botol	2 botol	3 botol	4 botol	5 botol	6 botol	7 botol	8 botol	9 botol	10 botol



3. LATIHAN (2)

- Contoh Tampilan Program

Jumlah Pembelian Shampoo : 10 [inputan]

2 shampoo gratis 1 shampoo.

4 shampoo gratis 2 shampoo.

6 shampoo gratis 3 shampoo.

8 shampoo gratis 4 shampoo.

10 shampoo gratis 5 shampoo.

Ayam tersebut mendapatkan gratis 5 shampoo tambahan.

- Catatan:

Kerjakan menggunakan
3 macam perulangan:

1. for
2. while
3. do while



**SELESAI
TERIMA KASIH**

