



# PERINTAH PERULANGAN

**Team Teaching JTI**  
POLINEMA 2019

# TUJUAN

Di akhir pertemuan, mahasiswa diharapkan :

1. Memahami konsep perulangan pada pemrograman
2. Mengenal sintaks perulangan di Java
3. Mampu menerapkan perulangan dalam persoalan

# OUTLINE

- Perulangan FOR
- Perulangan WHILE
- Perulangan DO-WHILE

# PENDAHULUAN

- Perintah perulangan/loop statement adalah perintah untuk mengulang satu atau lebih statement sebanyak beberapa kali.
- Loop statement digunakan agar kita tidak perlu menuliskan satu/sekumpulan statement berulang-ulang.
  - Dengan begitu maka kesalahan pengetikan bisa dikurang
- Dalam bahasa Java, ada 3 macam perintah perulangan yang umum digunakan yaitu:
  - Perintah for()
  - Perintah while()
  - Perintah do while()

# PERINTAH : FOR

Umumnya digunakan pada pengulangan yang jumlah iterasinya sudah pasti atau sudah diketahui sebelumnya.

Formula :

```
for (inisialisasi; kondisi; iterasi) {  
    //statement yang akan diulang  
}
```

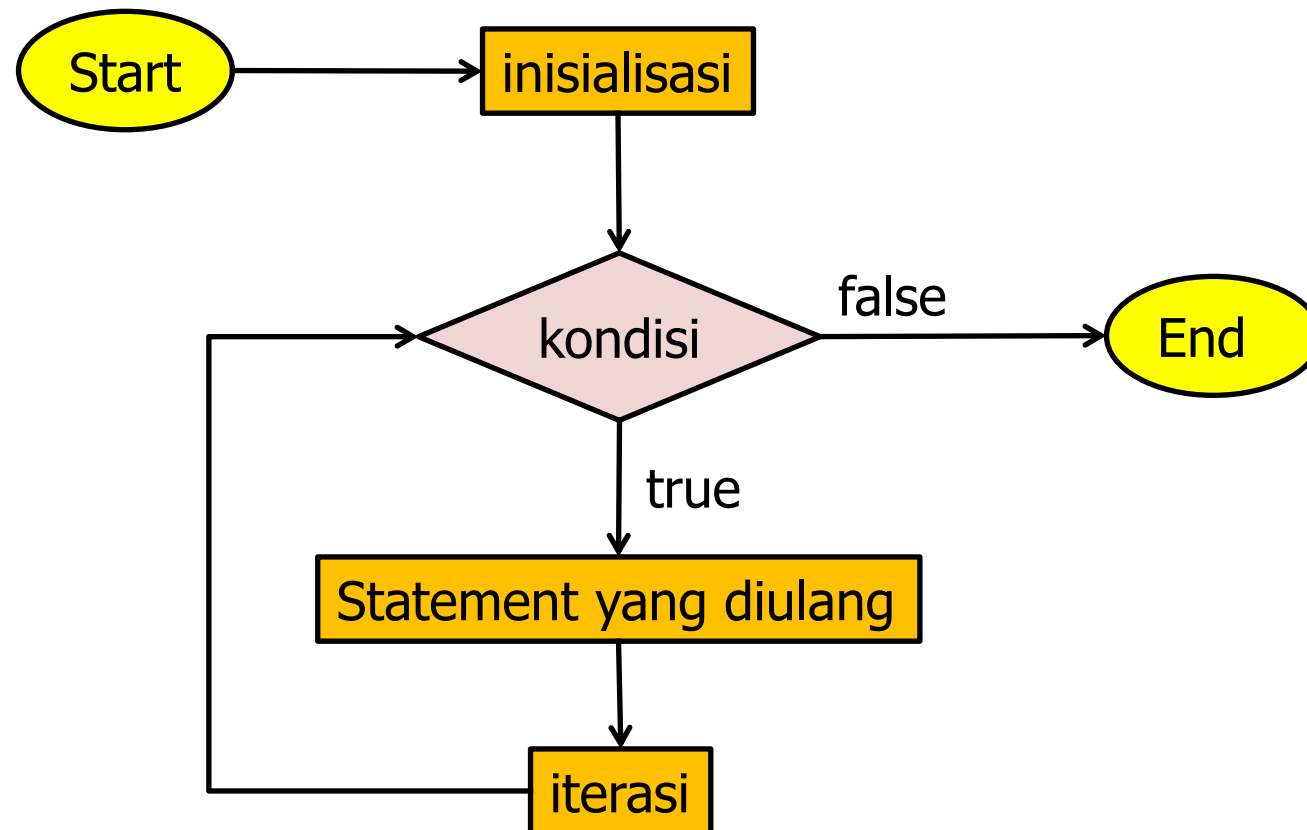
# PERINTAH : FOR

- Syntax dari perintah for():

```
for(inisialisasi nilai; syarat perulangan; perubahan nilai)
{
    ... // perintah-perintah yang akan diulang
}
```

- **Inisialisasi nilai** adalah tempat dimana kita akan memberikan nilai awal pada variabel counter (variabel yang digunakan untuk menghitung jumlah perulangan).
- **Syarat perulangan** adalah syarat yang harus dipenuhi agar perulangan tetap dilakukan.
- **Perubahan nilai** adalah perubahan yang akan dilakukan pada tiap putaran untuk menjamin bahwa perulangan tersebut tidak akan berlangsung terus menerus.

# ALUR KERJA FOR



# PERINTAH : FOR

- Perintah for() biasa digunakan untuk melakukan perulangan sebanyak jumlah yang telah diketahui.
- Contoh: Saya suka pemrograman sebanyak 10 kali

## Output

Saya suka pemrograman  
Saya suka pemrograman  
Saya suka pemrograman  
Saya suka pemrograman  
Saya suka pemrograman  
Saya suka pemrograman  
Saya suka pemrograman  
Saya suka pemrograman  
Saya suka pemrograman  
Saya suka pemrograman

BUILD SUCCESSFUL (total time: 0 seconds)

```
int i;  
  
for (i=1; i<=10; i++)  
{  
    System.out.print("Saya suka pemrograman");  
}
```

Sebelum dilakukan perulangan, variabel i diberi nilai 1

Perulangan akan dilakukan selama  $i \leq 10$

Pada tiap putaran, variabel i akan ditambah dengan 1



# CONTOH PERINTAH : FOR

- Buatlah program untuk mencetak ke layar angka 1 sampai 10.

```
int i;  
  
for (i=1; i<=10; i++)  
{  
    System.out.println(i);  
}
```

Output

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

BUILD SUCCESSFUL

## CONTOH PERINTAH : FOR

- Buatlah program untuk mencetak keterangan bilangan genap dan ganjil dari 1 sampai 10.

```
int i;

for (i=1; i<=10; i++){
    if (i%2==0){
        System.out.println("Angka " +i+ " adalah bilangan genap");
    }
    else {
        System.out.println("Angka " +i+ " adalah bilangan ganjil");
    }
}
```

### Output

```
Angka 1 adalah bilangan ganjil
Angka 2 adalah bilangan genap
Angka 3 adalah bilangan ganjil
Angka 4 adalah bilangan genap
Angka 5 adalah bilangan ganjil
Angka 6 adalah bilangan genap
Angka 7 adalah bilangan ganjil
Angka 8 adalah bilangan genap
Angka 9 adalah bilangan ganjil
Angka 10 adalah bilangan genap
```

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

# CONTOH-CONTOH “FOR”

1. Program untuk menuliskan teks “Java Programming” sebanyak 10 kali.

```
class DemoFor1 {  
    public static void main (String[] args) {  
        for (int i=0; i<10; i++) {  
            System.out.println("Java");  
        }  
    }  
}
```

```
class DemoFor1 {  
    public static void main (String[] args) {  
        for (int i=10; i>0; i--) {  
            System.out.println("Java");  
        }  
    }  
}
```

## 2. Program untuk menjumlahkan 5 bil. positif pertama

```
class DemoFor3 {  
    public static void main(String[] args) {  
        //int n = 5; // 5 bilangan positif pertama  
  
        int hasil = 0;  
        for (int i=1; i<=n; i++) {  
            hasil = hasil + i;  
            if (i != n) {  
                System.out.print(i + " + ");  
            } else {  
                System.out.print("= ");  
            }  
        }  
        System.out.println(hasil);  
    }  
}
```

### 3. Program untuk menentukan apakah sebuah integer termasuk prima atau tidak.

```
class DemoFor4 {  
    public static void main(String[] args) {  
        int bilangan = 7;  
        boolean prima = true;  
  
        for (int i=2; i <= (bilangan/2); i++) {  
            if ((bilangan % i) == 0) {  
                prima = false;  
                break; // menghentikan pengulangan  
            }  
        }  
        if (prima) {  
            System.out.println(bilangan + " merupakan bilangan prima");  
        } else {  
            System.out.println(bilangan + " bukan bilangan prima");  
        }  
    }  
}
```

# PENGUNAAN KOMA DALAM “FOR”

Perhatikan program berikut :

```
class DemoFor5 {  
    public static void main(String[] args) {  
        int j=4;  
  
        for (int i=0; i < 5; i++) {  
            System.out.println("Nilai i: " + i);  
            System.out.println("Nilai j: " + j);  
            System.out.println();  
            j--;  
        }  
    }  
}
```

Dapat disederhanakan sbb :

```
class DemoFor6 {  
    public static void main(String[] args) {  
        int i,j;  
  
        for (i=0, j=4; i < 5; i++, j--) {  
  
            System.out.println("Nilai i: " + i);  
            System.out.println("Nilai j: " + j);  
            System.out.println();  
        }  
    }  
}
```

# VARIASI “FOR”

For dengan  
kondisi berhenti  
sebuah variabel  
boolean

```
import java.util.Scanner;
public class ForBoolean {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int angka, i;
        boolean stop=false;

        for (i=0; !stop; i++){
            System.out.print("Masukkan angka : ");
            angka = input.nextInt();
            System.out.print(angka+" ");
            if (angka == 0)
                stop = true;
        }
        System.out.println("Selesai.");
    }
}
```



Variasi program sebelumnya :

```
import java.util.Scanner;

public class ForBoolean {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int angka, i;
        boolean stop=false;
        i = 0;
        for (; !stop; ){
            System.out.print("Masukkan angka : ");
            angka = input.nextInt();
            if (angka == 0)
                stop = true;
            i++;
        }
        System.out.println("Selesai.");
    }
}
```

# PERINTAH WHILE()

- Syntax dari perintah while():

```
while(syarat perulangan)
{
    ... // perintah-perintah yang akan diulang
}
```

- Syarat perulangan adalah syarat yang harus dipenuhi agar perulangan tetap dilakukan
- Perulangan while akan terus dijalankan selama syarat perulangan bernilai TRUE

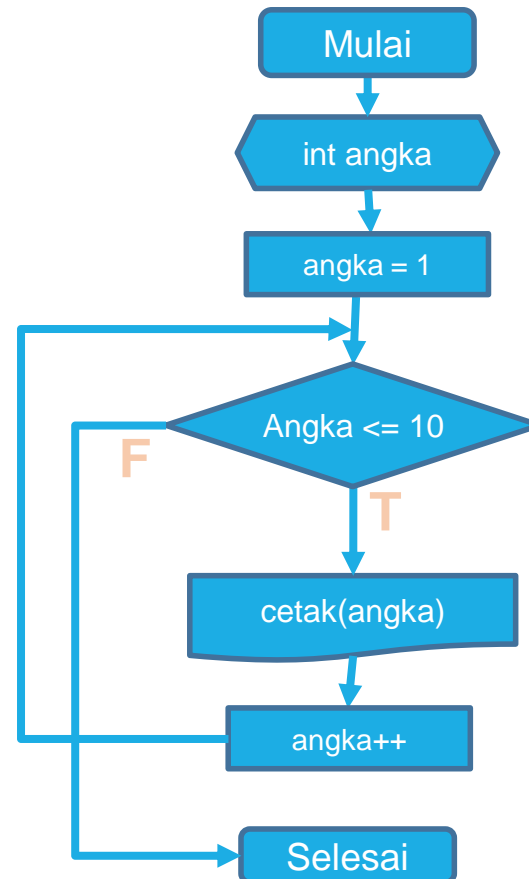
# PERINTAH : WHILE

Formula :

```
inisialisasi  
while (kondisi) {  
    //statement yang diulang  
    iterasi  
}
```

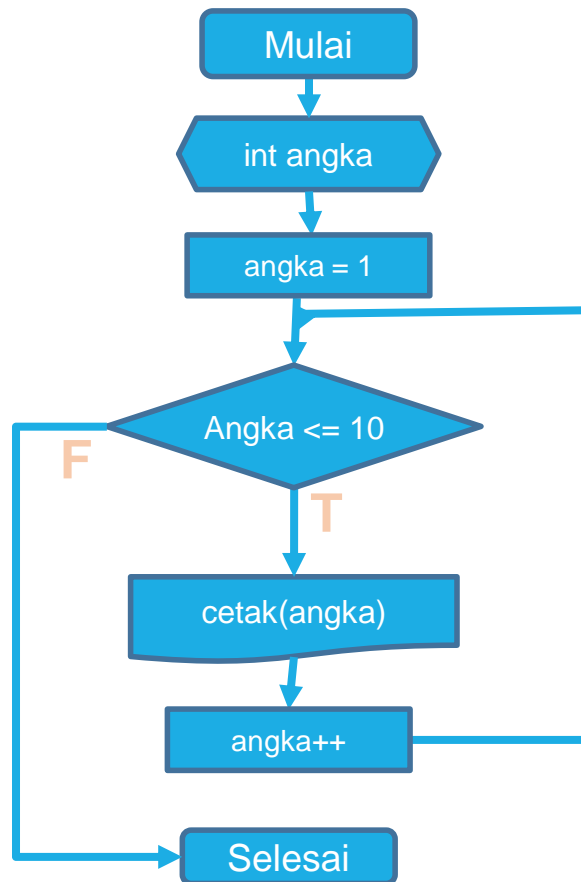
# FLASHBACK

- Masih ingatkah anda dengan program ini:
  - Buatlah program untuk mencetak ke layar angka dari 1 sampai 10



# CONTOH PERINTAH WHILE()

- Buatlah program untuk mencetak ke layar angka dari 1 sampai 10



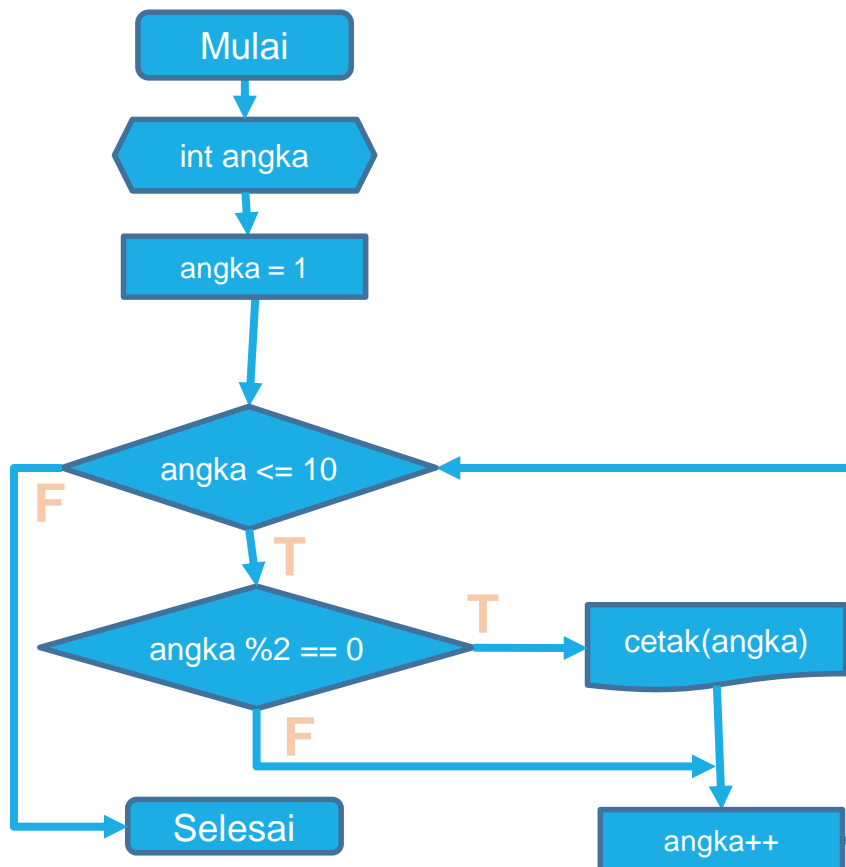
```
public static void main(String[] args)
{
    int angka;
    angka = 1;

    while (angka <= 10)
    {
        System.out.println (angka);
        angka ++;
    }
}
```

```
run:
1
2
3
4
5
6
7
8
9
10
BUILD SUCCESSFUL
```

# CONTOH PERINTAH WHILE()

- Buatlah program untuk mencetak bilangan genap dari 1 sampai 10



```
public static void main(String[] args) {
    int angka;
    angka = 1;

    while (angka <= 10)
    {
        if (angka%2==0)
        {
            System.out.println (angka);
        }
        angka ++;
    }
}
```

run:

2  
4  
6  
8  
10

BUILD SUCCESSFUL

## CONTOH PERINTAH WHILE()

- Buatlah program yang meminta input berupa bilangan negatif. Jika bilangan yang diinputkan positif, ulangi input.

```
int angka;  
System.out.println("Silahkan input bilangan positive ");  
Scanner input = new Scanner (System.in);  
angka = input.nextInt();
```

```
while (angka <0)  
{  
System.out.println("Maaf angka yang Anda masukkan adalah bilangan negatif ");  
System.out.println("Silahkan input bilangan lagi ");  
angka = input.nextInt();  
}  
System.out.println("Angka berhasil diinput");
```

Silahkan input bilangan positive

-2

Maaf angka yang Anda masukkan adalah bilangan negatif

Silahkan input bilangan lagi

0

Angka berhasil diinput

# CONTOH-CONTOH “WHILE”

```
import java.util.Scanner;

public class DemoWhile1 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int bil, jumlah=0;

        while (jumlah<=100){
            System.out.print("Masukkan angka : ");
            bil = input.nextInt();
            jumlah = jumlah+bil;
        }
        System.out.println("Selesai. Total masukan user = "+jumlah);
    }
}
```

Program yang akan menjumlahkan secara berulang integer masukan user dan berhenti ketika jumlahnya sudah >100



## Program miniATM

```
import java.util.Scanner;
public class DemoWhile2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String jawab="y";
        int pilih;
        long trf, saldo = 1500000;

        while (jawab.equalsIgnoreCase("y")){
            System.out.println("M E N U");
            System.out.println("1. Cek Saldo");
            System.out.println("2. Transfer");
            System.out.print("Masukkan pilihan : ");
            pilih = input.nextInt();
            if (pilih==1){
                System.out.println("Saldo rekening anda Rp "+saldo+",00");
            }else if (pilih==2) {
                System.out.println("Jumlah yang akan ditransfer : Rp ");
                trf = input.nextLong();
                saldo = saldo-trf;
                System.out.println("Transfer berhasil.");
            }else {
                System.out.println("Pilihan yang anda masukkan salah.");
            }
            System.out.print("Transaksi lagi (Y/T) ? ");
            jawab = input.next();
        }
        System.out.println("TERIMA KASIH.");
    }
}
```

# PERINTAH DO-WHILE()

- Syntax perintah do-while():

```
do
{
    ... // perintah-perintah yang akan diulang
}
while(syarat pengulangan);
```

- Pada prinsipnya, perintah do-while() sama dengan perintah while().
- Perintah do-while() akan mengulang statement miliknya selama syarat pengulangannya terpenuhi.
- Hanya saja, perintah do-while() menjalankan statementnya terlebih dahulu, setelah itu baru memeriksa syaratnya.
  - Sedangkan perintah while() memeriksa syarat terlebih dahulu.
- Oleh karena itu, perintah do-while() akan menjalankan statementnya sebanyak satu kali, meskipun syarat pengulangan tidak terpenuhi.

# PERULANGAN “DO-WHILE”

Formula

```
inisialisasi  
do {  
  //statement yang akan diulang  
  ...  
  iterasi  
} while (kondisi);
```

# CONTOH

```
class ContohDoWhile {  
    public static void main (String [] args) {  
        int i = 6;  
        do {  
            System.out.println("Java");  
            i++;  
        } while (i<5);  
    }  
}
```

# PERULANGAN BERSARANG (1)

“for bersarang”

```
public class NestedFor {  
    public static void main(String[] args) {  
        int i,j;  
  
        for (i=1;i<=5;i++){  
            for (j=1;j<=i;j++){  
                System.out.print(i+" ");  
            }  
            System.out.println();  
        }  
    }  
}
```

# PERULANGAN BERSARANG (2)

“while bersarang”

```
public class NestedWhile {  
    public static void main(String[] args) {  
        int i=1,j=1;  
  
        while (i<=5){  
            while (j<=i){  
                System.out.print(i+" ");  
                j++;  
            }  
            System.out.println();  
            i++;  
        }  
    }  
}
```

# PERULANGAN BERSARANG (3)

“do-while bersarang”

```
public class NestedDoWhile {  
    public static void main(String[] args) {  
        int i=1,j=1;  
  
        do {  
            do {  
                System.out.print(i+" ");  
                j++;  
            } while (j<=i);  
            System.out.println();  
            i++;  
        } while (i<=5);  
    }  
}
```

# PRETEST (10 MENIT)

1. Perbaiki kode berikut :

```
for ( i = 100, i >= 1, i++ )  
    System.out.println( i );
```

2. Kode berikut harusnya menampilkan integer ganjil dari 19 s/d 1 :

```
for ( i = 19; i >= 1; i += 2 )  
    System.out.println( i );
```



3. Kode berikut harusnya menampilkan integer genap dari 2 s/d 50 :

```
counter = 2;  
  
do {  
    System.out.println( counter );  
    counter += 2 ;  
} while ( counter < 50 );
```

d) Apakah output program berikut :

```
public class Printing {  
    public static void main(String args[]) {  
        for ( int i = 1; i <= 3; i++ ) {  
            for ( int j = 1; j <= 3; j++ )  
                System.out.print( '@' );  
            System.out.println();  
        } // end outer for  
    } // end main  
} // end class Printing
```

# LATIHAN

1. Buat program perulangan yang outputnya :
  - a. 1 3 5 7 9 ... 99
  - b. 100 99 98 97 ... 1
  - c. 5 10 15 20 ... 100
  - d. 2 4 8 16 32 ... 256
2. Modifikasilah contoh program di slide 12 sehingga banyaknya integer yang dimasukkan tergantung masukan user. Program akan terus menerus mengulang hingga user memasukkan huruf 't' pada pertanyaan "Ingin mengulang lagi (y/t) ?"