

For Loop

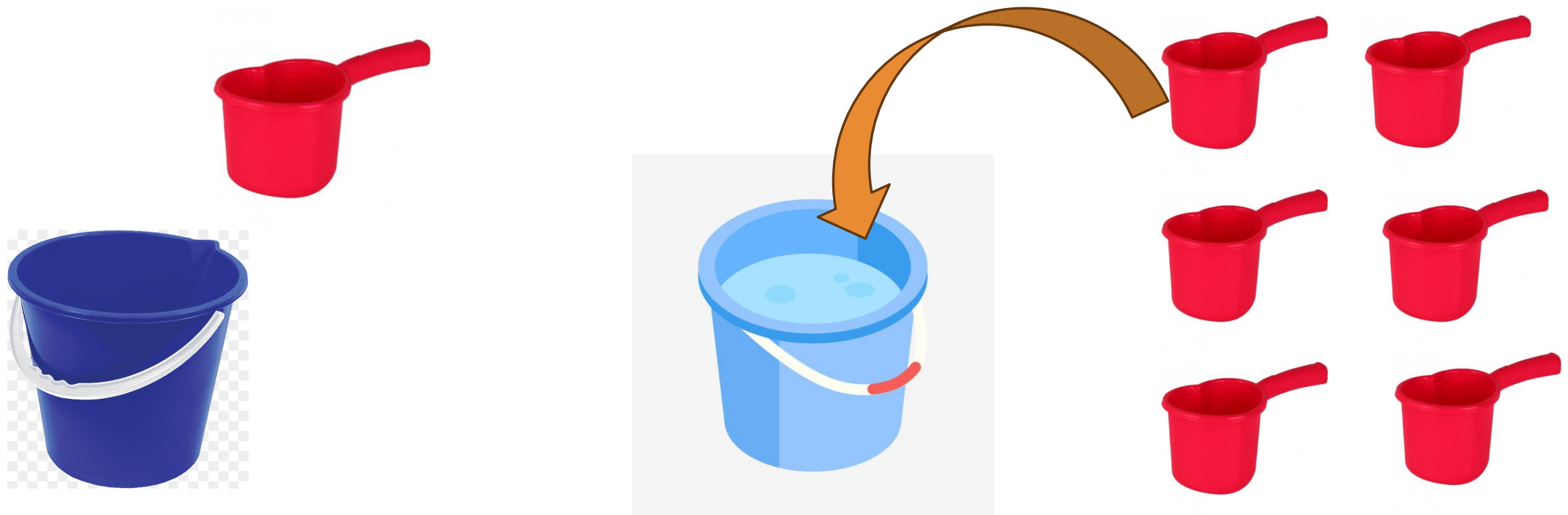


Pengertian instruksi perulangan

Instruksi pada komputer yang berasal dari algoritma adalah suatu perintah berupa pernyataan-pernyataan yang harus dikerjakan oleh komputer.

Salah satu bentuk instruksi utama pemrograman dari beberapa jenis instruksi yang ada adalah jenis instruksi perulangan.

Dalam dunia nyata pun kita tanpa menyadari telah berinteraksi dengan sejumlah situasi yang menggunakan teknik perulangan.





Instruksi perulangan adalah instruksi yang dapat mengulang sederetan instruksi-instruksi lainnya berulang sesuai persyaratan yang ditetapkan

Struktur perulangan pada dasarnya terdiri atas:

1. Kondisi perulangan
2. Badan (body) perulangan.
3. pencacah (counter) perulangan.



Perulangan dalam pemrograman dibutuhkan jika perlu melakukan eksekusi secara beberapa kali pada sekelompok pernyataan (statement).

Terdapat beberapa bentuk algoritma dasar perulangan, yaitu: for, while, do ... while, dan bentuk perulangan bersarang dan perulangan dengan seleksi.

Berikut ini bentuk umum algoritma perulangan *for* yang disajikan dalam bentuk pseudo code dan flowchart:

```
for (var=awal to akhir step n){  
    Statement(s)  
}
```



Penjelasan:

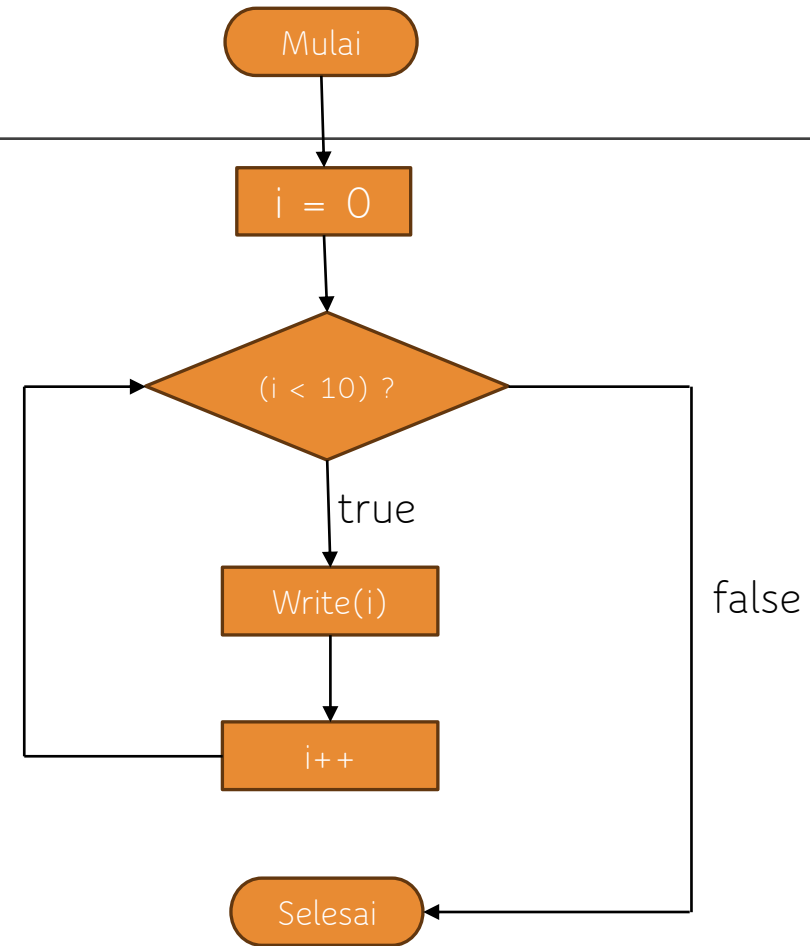
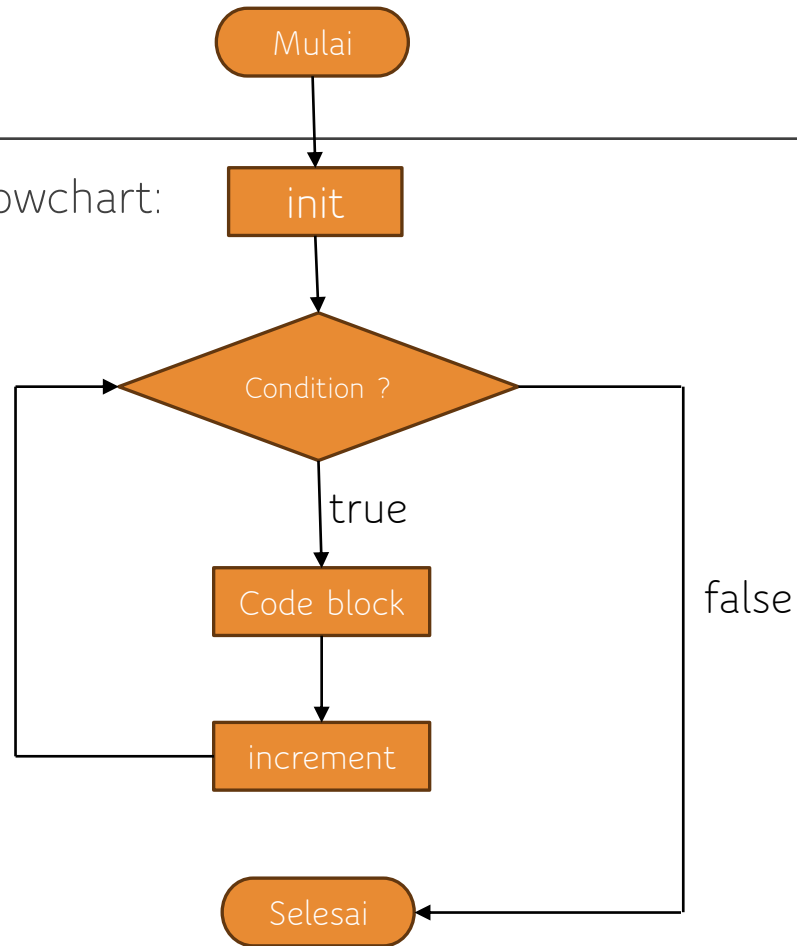
1. Variabel perulangan(var) bertipe dasar
2. Nilai awal harus lebih kecil dari akhir bila $n > 0$ (positif)
3. Nilai awal harus lebih besar dari akhir bila $n < 0$ (negatif)
4. Dimulai dengan var bernilai awal, kemudian setiap kali putaran maka nilai var akan bertambah sebesar n
5. Perulangan akan berhenti apabila nilai var sudah mencapai akhir



Dalam bentuk pseudo code lain:

```
for (init; condition; chng-condition){  
    Statement(s)  
}
```

Dalam bentuk flowchart:





Contoh 1. Program (C)

```
int main(){  
    for (int x = 1; x <10; x++) {  
        printf("%d ", x);  
    }  
}
```

Output :

```
1 2 3 4 5 6 7 8 9
```

x	x > 0	printf x	i++
1	True	1	2
2	True	2	3
3	True	3	4
4	True	4	5
5	True	5	6
6	True	6	7
7	True	7	8
8	True	8	9
9	True	9	10
10	False	Keluar dari loop	



Contoh 2. Program (Java)

```
public class Forloop{  
    Run | Debug  
    public static void main(String[] args) {  
        for (int x = 0; x < 10; x++) {  
            System.out.print(x+" |");  
        }  
    }  
}
```

Output :

```
0 1 2 3 4 5 6 7 8 9
```

Contoh 3. Program(C)

```
#include <stdio.h>

int main(){
    for (int x = 10; x > 0; x--) {
        printf("%d ", x);
    }
}
```

Output :

```
10 9 8 7 6 5 4 3 2 1
```

x	x > 0	printf x	i--
10	True	10	9
9	True	9	8
8	True	8	7
7	True	7	6
6	True	6	5
5	True	5	4
4	True	4	3
3	True	3	2
2	True	2	1
1	True	1	0
0	False	Keluar dari loop	



Contoh 4. Program (Java)

```
public class Forloop{  
    Run | Debug  
    public static void main(String[] args) {  
        for (int x = 10; x > 0; x--) {  
            System.out.print(x+" ");  
        }  
    }  
}
```

Output :

```
10 9 8 7 6 5 4 3 2 1
```



Latihan 1

Buat Program untuk melakukan push up sebanyak 10 kali, jika push up sudah mencapai 10 maka push up dihentikan / selesai:

Output program yang diinginkan hanya menampilkan "Persiapan", "mulai ", "push up 1", "push up 2" ..., "push up 10", "selesai"



Program C

```
#include <stdio.h>

int main(){
    printf("Persiapan ..... \n");
    printf("mulai \n");
    for (int i = 1; i <= 10; i++) {
        printf("Push up %d\n", i);
    }
    printf("Selesai");
}
```

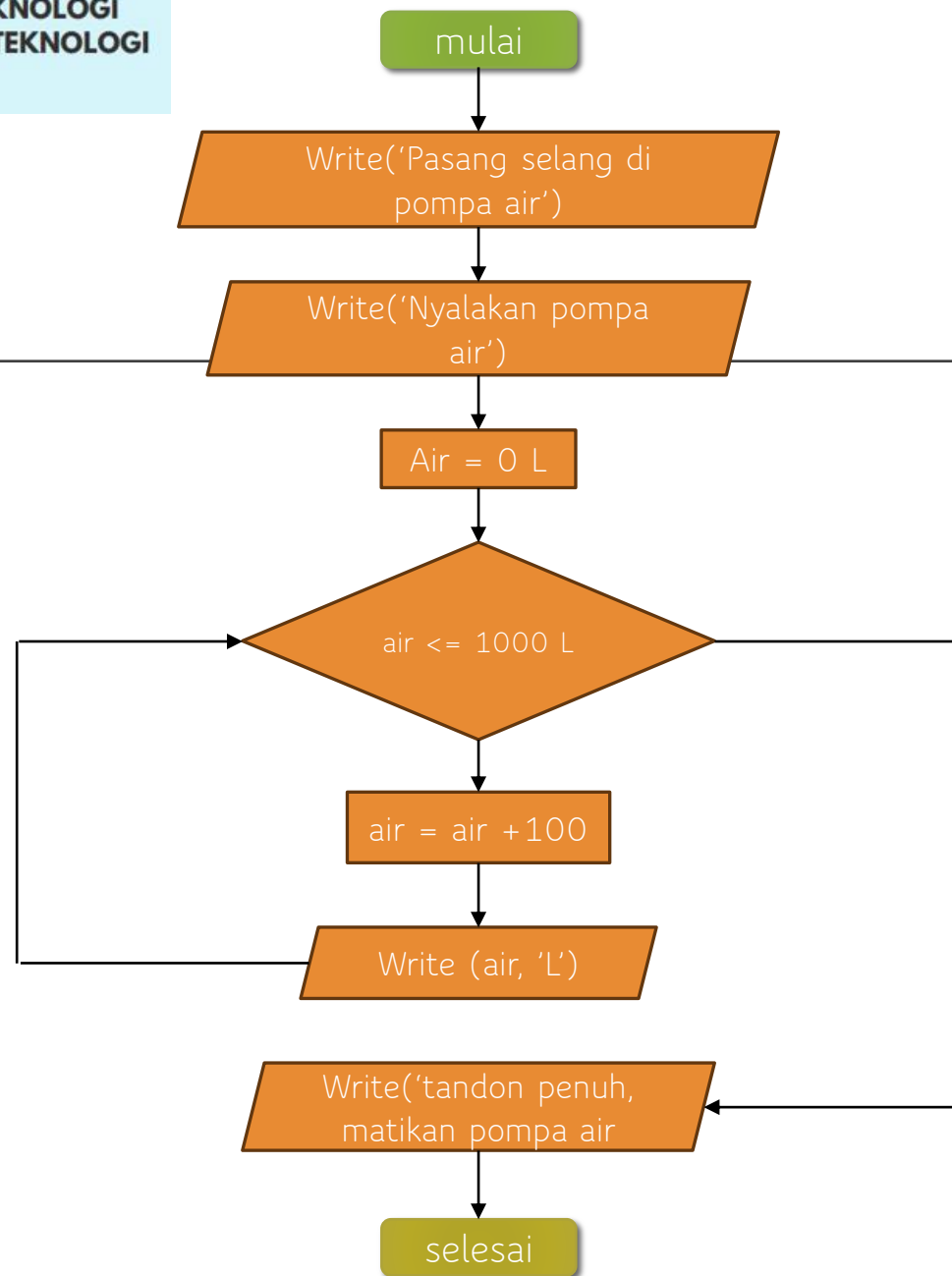


Program Java

```
public class PushUp {  
    Run | Debug  
    public static void main(String[] args) {  
        System.out.println(x:"Persiapan ...");  
        System.out.println(x:"mulai");  
        for (int i = 1; i <= 10; i++) {  
            System.out.println("Push Up " + i);  
        }  
        System.out.println(x:"Selesai ...");  
    }  
}
```




Latihan 2





GCD /FPB (Tugas)

Diberikan 2 buah bilangan bulat tak negatif m dan n ($m \geq n$)

Algoritma Euclidean

1. Jika $n = 0$ maka

m adalah PBB(m, n);

stop.

tetapi jika $n \neq 0$,

lanjutkan ke langkah 2.

2. Bagilah m dengan n dan misalkan r adalah sisanya.

3. Ganti nilai m dengan nilai n dan nilai n dengan nilai r , lalu ulang kembali ke langkah 1.

Buatkan program untuk menghitung GCD berdasarkan algoritma di atas menggunakan instruksi *for loop* !