**MATERI LOOPING PADA JAVA**

Perulangan dalam pemprograman dibagi menjadi dua jenis yaitu:

* **Counted loop**: perulangan yang jumlah pengulangannya terhitung atau tentu
* **Uncounted loop**: perulangan yang jumlah pengulangannya tidak terhitung atau tidak tentu

1. COUNTED LOOP

Seperti yang sudah dijelaskan perulangan ini memiliki jumlah pengulangan yang tentu dan terhitung. Perulangan ini terdiri dari perulangan for dan for each.

1. **Perulangan For**

Format penulisan perulangan For di java adalah sebagai berikut:

For( int hitungan = 0; hitungan <= 10; hitungan++ ){

// blok kode yang akan diulang

}

Penjelasan:

1. Fariabel hitungan tugasnya untuk menyimpan hitungan pengulangan.
2. Hitungan <= 10 artinya selama nilai hitungannya lebih kecil atau sama dengan 10, maka pengulangan akan terus dilkakukan. Dengan kata lain, perulangan ini akan mengulang sebnyak 10 kali.
3. Hitungan++ fungsinya untuk menambah satu (+1) nilai hitungan pada setiap pengulangan.
4. Blok kode For dimulai dengan tanda ‘{‘ dan akhir dengan ‘}’.

**Contoh program perulangan For**

Kita buat class baru bernama Bintang kemudian ikuti kode berikut:

Class Bintang{

Public static void main(String[ ] args){

For(int i=0; I <= 5; i++){

System .out.printin(“\*\*\*\*\*”);

}

}

Hasil output:

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

Sekarang kita membuat program untuk menampilkan nilai dari 1 sampai 10 dengan menggunakan perulangan.

Class CetakAngka{

Public static void main ( String [ ] argument ){

For (int i=0; I <= 10; i++){

System .out.printin( I + “ “);

}

}

Hasil output:

0 1 2 3 4 5 6 7 8 9 10

Selanjutnya kita coba lakukan sedikit eksperimen.

Buat sebuah program yang menampilkan bilangan ganjil saja.

Class CetakBilanganGanjil{

Public static void main(String[ ] argument){

For (int I =1; I <= 20; i += 2)

System .out.printin( I + “ “);

}

}

Hasil output:

1 3 5 7 9 11 13 15 17 19

Perhatikan: disana kita menggunakan i+= 2, bukan i++.

Maksudnya , nilai I akan ditambah dua (+2) di setiap pengulangan.

1. **Perulangan For Each**

Perulangan ini sebenarnya digunakan untuk menampilkan isi dari array.

Apa itu array?

Singkatnya, array itu variable yang menyimpan lebih dari satu nilai dan memiliki indeks.

Perulangan For Each pada java, dilakukan juga dengan kata kunci For.

Contohnya seperti ini:

For ( int item : dataArray ) {

// blok kode yang diulang

}

Penjelasan :

1. Variable item akan menyimpan nilai dari array
2. Kita bisa baca seperti ini: “untuk setiap item dalam dataArray, maka lakukan perulangan.

**Contoh program For Each**

Buat sebuah kelas baru bernama perulanganForeach, kemudian ikuti kode berikut.

Public class perulanganForeach {

Public static void main( String [ ] args) {

// membuat array

Int angka [ ] = {3, 1, 42, 24, 12};

// menggunakan perulangan for each untuk menampilkan angka

For ( int x : angka ){

System .out.print (x + “ “);

}

}

}

Hasil outputnya:

3 1 42 24 12

1. **UNCOUNTED LOOP**

Seperti yang dijelaskan dari awal tadi, perulangan ini tidak jelas jumlah pengulangannya.

Tapi tidak menutup kemungkinan juga, jumlah pengulangannya dapat ditentukan. Perulngan uncounted loop terdiri dari perulangan while dan Do while.

1. **Perulangan while**

While bisa kita artikan selama. Cara kerja perulangan Ini seperti percabangan, ia akan melakukan perulangan selama kondisinya bernilai true.

Struktur penulisan perulangan while:

While ( kondisi) {

// blok kode yang akan diulang

}

Penjelasan:

1. Kondisi bisa kita isi dengan perbandingan maupun variable Boolean. Kondisi ini hanya memiliki nilai true dan flase.
2. Perulangan while akan berhenti sampai kondisi bernilai false.

Contoh program dengan perulangan while

Program ini akan melakukan perulangan selama jawabannya tidak

Import java.util.Scanner;

Public class perulanganwhile {

Public static void main( String [ ] args ) {

// membuat variable dan Scanner

Boolean running = true;

Int counter = 0;

String jawab;

Scanner scan = new scanner(system.in);

While ( running ) {

System.out.printing(“apakah anda ingin keluar?”);

System.out.printin(“ jawab [ ya / tidak] > “);

Jawab = scan.nextline( );

// cek jawabannya, kalau ya maka berhenti mengulan

If( jawab.equalsignoreCase(“ya”) ){

Running = false;

}

Counter++;

}

System,out.printin( “anda sudah melakukan perlangan sebnyak” + counter + “ kali “);

}

}

Hasil outputnya:

Run:

Apakah anda ingin keluar?

Jawab [ ya/tidak ]> tidak

Apakah anda ingin keluar?

Jawab [ ya/tidak ]> tidak

Apakah anda ingin keluar?

Jawab [ ya/tidak ]> tidak

Apakah anda ingi keluar?

Jawab [ ya/tidak ]> ya

Anda sudah melakukan perulangan sebanyak 4 kali

Build successful ( total time: 9 second)

Disana telah dilakukan perulangan sebanyak 4 kali. Bisa saja terjadi 10 kali itu tergantung dari kondisinya. Kalau nilai variable running bernilai false maka perulangan berhenti.

Contoh kode while di atas dapat kita baca seperti ini: “lakukan perulangan selama nilai running bernilai true.” Tidak menutup kemungkinan juga, perulangan ini dapat melakukan counted loop.

Contohnya seperti ini:

Int I = 0;

While ( I <= 10 ){

// blok kode yang akan diulang

System.out.printin( ‘perulangan ke-‘ + i);

// increment nilai i

I++;

}

Hasil outputnya:

Perulangan ke-0

Perulangan ke-1

Perulangan ke-2

Perulangan ke-3

Perulangan ke-4

Perulangan ke-5

Perulangan ke-6

Perulangan ke-7

Perulangan ke-8

Perulangan ke-9

Perulangan ke-10

Pastikan melakukan increment ( i++) terhadap variable counter. Karena kalau tidak, perulngannya akan terus-menurus dilakukan sampai komputernya hang.

1. **Perulangan Do/While**

Cara kerja perulangan Do/While sebenarnya sama seperti perulangan dulu. Kemudian mengecek kondisinya.

Struktur penulisannya seperti ini:

Do {

// blok kode yang akan diulang

} while ( kondisi)

Jadi kerjakan dulu ( Do), baru di cek kondisnya while ( kondisi ). Kalau kondisi bernilai ture, maka lanjutkan perulangan.

Contoh program dengan perulangan Do/While

Public class perulnganDoWhile {

Public static void main ( string [ ] args ) {

// membuat variable

Int I = 0;

Do {

System.out.printin( “ perulangan ke-“ + i);

I++;

} while ( I <= 10);

}

}

Hasil outputnya:

Perulangan ke-0

Perulangan ke-1

Perulangan ke-2

Perulangan ke-3

Perulangan ke-4

Perulangan ke-5

Perulangan ke-6

Perulangan ke-7

Perulangan ke-8

Perulangan ke-9

Perulangan ke-10