

# PRAKTIKUM 9

## PEMROGRAMAN

**LOOPING** (for, while, do-while, break & continue, Format Output Data)



Oleh :

NAMA : MUHAMMAD TOMY ISKANDAR  
NIM : 361655401091  
KELAS : 1D  
MATA KULIAH : KONSEP PEMROGRAMAN

POLIWANGI

TAHUN AJARAN 2016/2017

## Dasar teori

Laporan kontrol loop yang digunakan untuk mengeksekusi pernyataan atau sekelompok pernyataan beberapa kali. Misalkan Anda ingin mencetak angka 10 kali. Alih-alih menggunakan 10 `System.out.println ()` pernyataan, kita bisa hanya loop melalui `System.out.println ()` pernyataan tunggal 10 kali. Mari kita lihat konstruksi perulangan yang didukung oleh Java.

'Untuk' loop Pertimbangkan potongan kode berikut:

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

`i = 0;` is the initialization statement.

`i <= 10;` is the condition statement.

`i++;` is the increment statement.

Inisialisasi (`i = 0;`) pernyataan dieksekusi sekali. Kemudian, kondisi (`i < 10;`) pernyataan diperiksa. Jika mengevaluasi untuk Benar, tubuh loop dijalankan. Jika mengevaluasi ke Salah, lingkaran dihentikan dan pernyataan setelah badan loop dijalankan. Setelah semua pernyataan dalam tubuh loop dieksekusi, kontrol melompat ke pernyataan kenaikan dan pernyataan kondisi dievaluasi lagi. Setiap loop berikut sintaks ini:

```
for ( init; condition; increment / decrement ) {  
    statement 1;  
    statement 2;  
    .  
    .  
    statement n;  
}
```

untuk loop dikenal sebagai entri dikendalikan lingkaran sejak cek kondisi dilakukan pada awal loop. Program berikut mengilustrasikan penggunaan untuk loop:

'while' lingkaran Sintaks loop sementara adalah:

```
while ( condition ) {  
    statement 1;  
    statement 2;  
    .  
    .  
    statement n;  
}
```

Initialization statement is declared just before the while statement.

Increment / Decrement statement is present inside the body of while loop.

while loop is also known as entry controlled loop. It is possible that the body of the while loop is not executed even once. Following program illustrates the use of while loop :

'Do ... while loop Sintaks do ... sementara sementara loop adalah:

```
do {  
    statement 1;  
    statement 2;  
    .  
    .  
    statement n;  
} while ( condition );
```

'Do ... while loop dikenal sebagai exit dikendalikan lingkaran sejak cek kondisi dilakukan pada akhir loop. Program berikut mengilustrasikan penggunaan do ... while:

bersarang Loops Kita dapat menggunakan loop di dalam tubuh loop lain. Hal ini dikenal sebagai sarang loop. Program berikut mengilustrasikan penggunaan loop bersarang:

istirahat 'dan' lanjutkan 'pernyataan 'Istirahat' pernyataan digunakan untuk mengakhiri loop segera tanpa pergi untuk cek bersyarat lanjut. Dalam kasus loop bersarang, loop terdalam dihentikan dan eksekusi dimulai dari baris berikutnya dari program setelah loop batin. Kita telah melihat bahwa istirahat digunakan untuk keluar dari saklar kasus konstruksi. 'Lanjutkan' pernyataan tidak menghentikan loop tetapi pergi ke awal loop untuk iterasi berikutnya melewati sisa badan loop. Program berikut mengilustrasikan penggunaan pernyataan break:

Untuk loop seharusnya memiliki 10 iterasi mencetak 0 1 2 3 4 5 6 7 8 9 tetapi pernyataan istirahat menyebabkan penghentian loop setelah saya menjadi 7. Jadi output akan 0 1 2 3 4 5 6. Program berikut mengilustrasikan penggunaan terus pernyataan:

Membangun & Run Output dari program di atas adalah 1 3 5 7 9. Pernyataan cetak dilewati menggunakan pernyataan terus ketika kita melihat genap.

## ANALISA DAN PROJEK

### for Increment (percob1)

```
package LOOPING; // projek ini berada di packaage bernama LOOPING
```

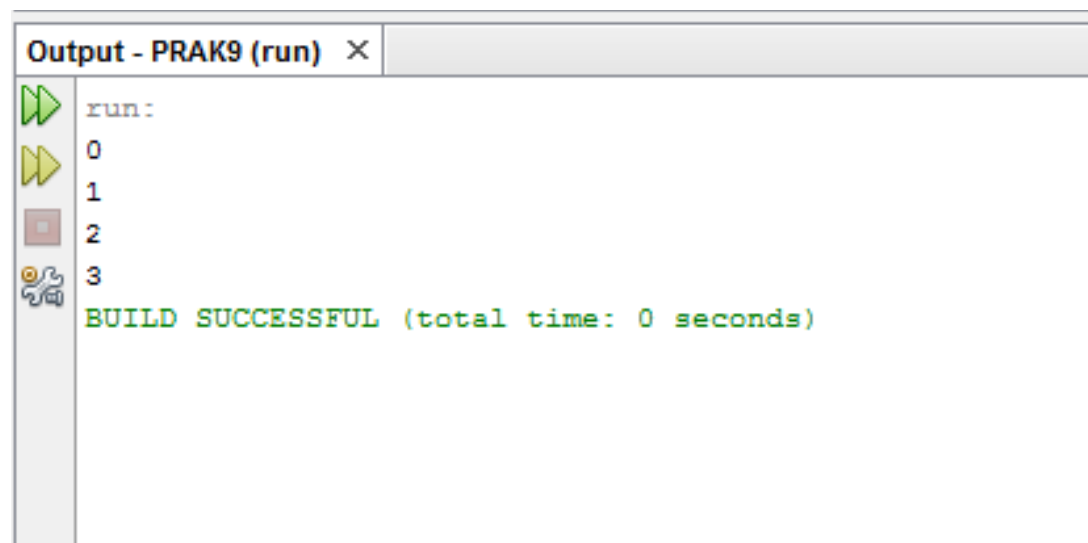
```
public class percob1 { // projek ini berada di class bernama percob1
```

```
    public static void main(String[] args) // fungsi main public pada bagian ini berarti
    bahwa metode ini dapat dipanggil diluar atau didalam kelas, static berarti metode tidak akan
    mengirim apapun setelah semua selesai. fungsi main berisi argumen yang diambil dari ruang
    eksekusi
```

```
    { for(int i=0;i<4;i++) // Keyword for digunakan untuk melakukan perulangan dengan
    nilai seleksi adalah sebuah angka numeric. Bisa berupa bilangan bulat maupun pecahan.
    Untuk nilai. Untuk i=0 mendefinisikan bahwa nilai awal yang nantinya akan diproses adalah
    angka 0. Dan variabel i<4 mendefinisikan bahwa nilai inilah yang menjadi batas dari proses
    yaitu angka kurang dari angka 4. Sedangkan variabel i++, mendefinisikan bahwa proses ini
    merupakan proses perubahan nilai agar nilai variabel mencapai batas. Sebelum mencapai
    batas maka program akan menjalankan statement yang ada di blok FOR terus menerus. Dan
    fungsi dari ++ adalah prosesnya dalam menjalankan program dengan cara perubahan lebih
    dari nilai awal dengan tampilan dari atas kebawah
```

```
        {
            System.out.println(i); //menampilkan hasil output
        }
    }
}
```

### Output percobaan 1



```
Output - PRAK9 (run) X
run:
0
1
2
3
BUILD SUCCESSFUL (total time: 0 seconds)
```

### for Decrement (percob2)

```
package LOOPING; // berada di package LOOPING
```

```
public class percob2 { //berada di class percob2
```

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

```
    { for(int i=4;i>0;i--) // mendefinisikan bahwa nilai awalnya adalah angka 4 dan batas dari  
    nilai angkanya adalah lebih dari angka 0 dan untuk proses perubahannya atau proses  
    menjalankan program adalah dengan cara kurang dari nilai awal. Dengan tampilan dari bawah  
    ke atas ( - )
```

```
    {
```

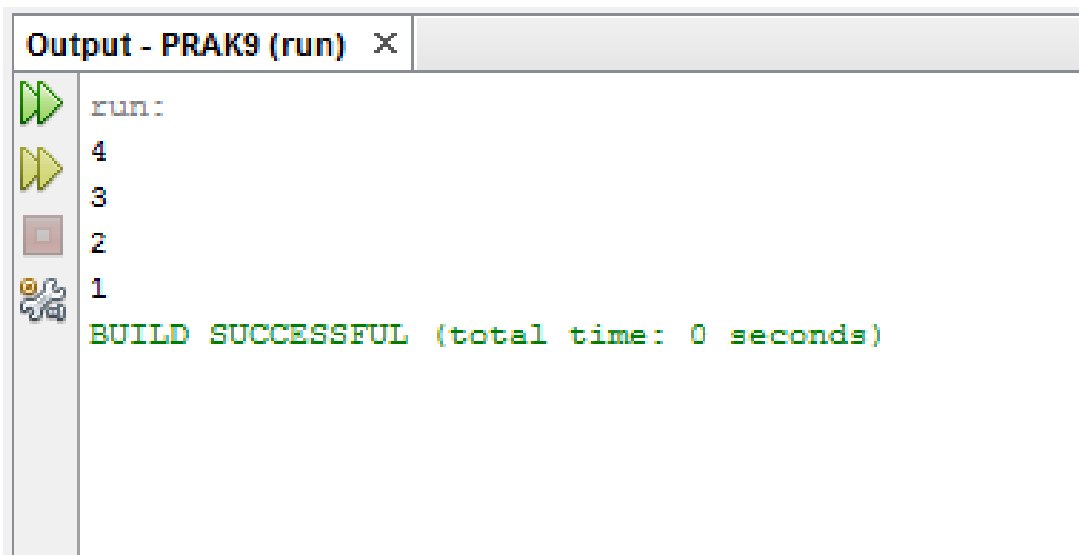
```
        System.out.println(i); //menampilkan output
```

```
    }
```

```
    }
```

```
}
```

### Output percobaan 2



### Pencetak bilangan genap (even)

```
package LOOPING; //berada di package LOOPING

public class percob3 { //berada di class percob3

    public static void main(String[] args) {

        for(int i=0; i <= 10 ; i += 2){ //mendeklarasikan bahwa nilai awalnya adalah angka 0
            dan batas nilai prosesny adalah kurang dari samadengan 10 dan fungsi dari ( i+=2 ) adalah
            sebagai nilai perubahan prosesnya yaitu ditambah angka nilai dua ( 2 ) per proses sehingga
            outputnya nanti urut dari 0 hingga 10 namun ditambah dua per proses urut dari atas kebawah
            karena menggunakan attribut +

                System.out.print(i + " "); // menampilkan output pertama satu persatu diproses dengan
                cara ditambah 2 per proses dengan tampilan horizontal bukan vertikal

        }

        System.out.println(); // menampilkan outputlagi

        for(int i=10; i >= 0 ; i -= 2){ //mendeklarasikan bahwa nilai awalnya adalah angka 10
            dan batas nilai prosesny adalah kurang dari sama dengan 0 dan fungsi dari ( i-=2 ) adalah
            sebagai nilai perubahan prosesnya yaitu dikurang angka nilai dua ( 2 ) per proses sehingga
            outputnya nanti urut dari 10 hingga 0 namun dikurang dua per proses urut dari ke atas karena
            menggunakan attribut -

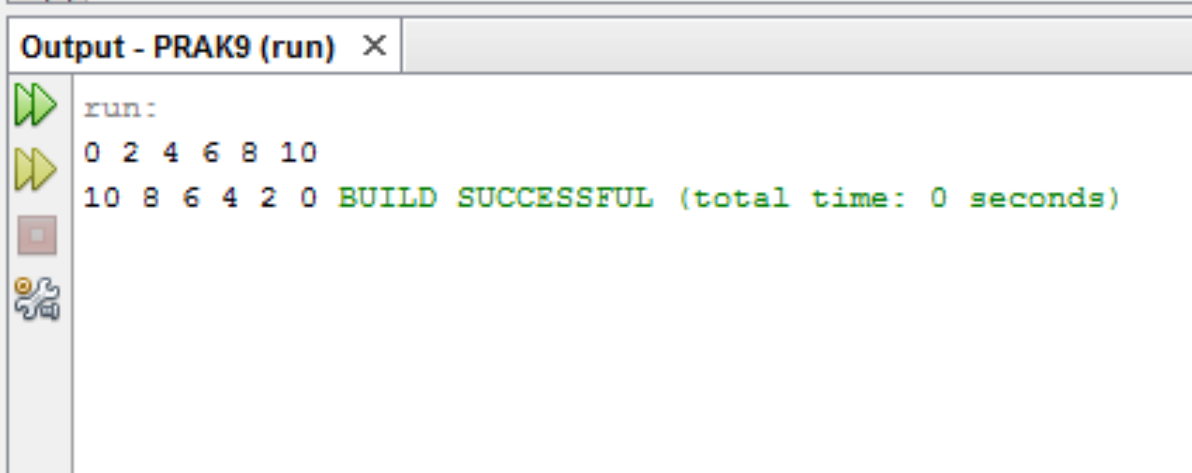
                System.out.print(i + " "); // menampilkan output kedua satu persatu diproses dengan
                cara dikurang 2 per proses dengan tampilan horizontal bukan vertikal

        }

    }

}
```

### Output dari percobaan 2



```
Output - PRAK9 (run) X
run:
0 2 4 6 8 10
10 8 6 4 2 0 BUILD SUCCESSFUL (total time: 0 seconds)
```

### Pencetak bilangan ganjil (odd)

```
package LOOPING; //berada di package LOOPING

public class percob4 { //berada di class percob4

    public static void main(String[] args) {

        for(int i=1; i <= 10 ; i += 2){ // mendefinisakan bahwa nilai awalnya adalah angka 1 dan
            batas angka prosesnya adalah angka kurang dari 10 dan perubahan dari proses per prosesnya
            adalah dengan ditambah dua per proses dan menampilkan dari atas kebawah ( + )

            System.out.print(i + " "); // menampilkan output pertama satu persatu diproses dengan
            cara ditambah 2 per proses dengan tampilan horizontal bukan vertikal

        }

        System.out.println(); //menampilkan output lagi

        for(int i=9; i > 0 ; i -= 2){ // mendefinisakan bahwa nilai awalnya adalah angka 9 dan
            batas angka prosesnya adalah angka lebih dari 10 dan perubahan dari proses per prosesnya
            adalah dengan ditambah dua per proses dan menampilkan dari bawah ke atas kareana
            menggunakan ( - )

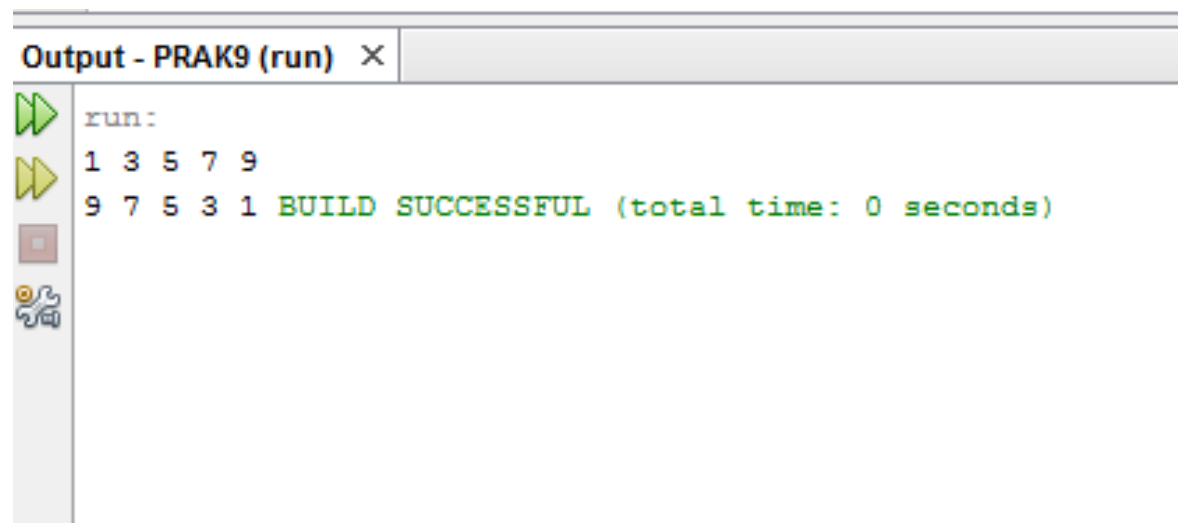
            System.out.print(i + " "); // menampilkan output pertama satu persatu diproses dengan
            cara ditambah 2 per proses dengan tampilan horizontal bukan vertikal

        }

    }

}
```

### Outputan dari percob4 bilangan ganjil

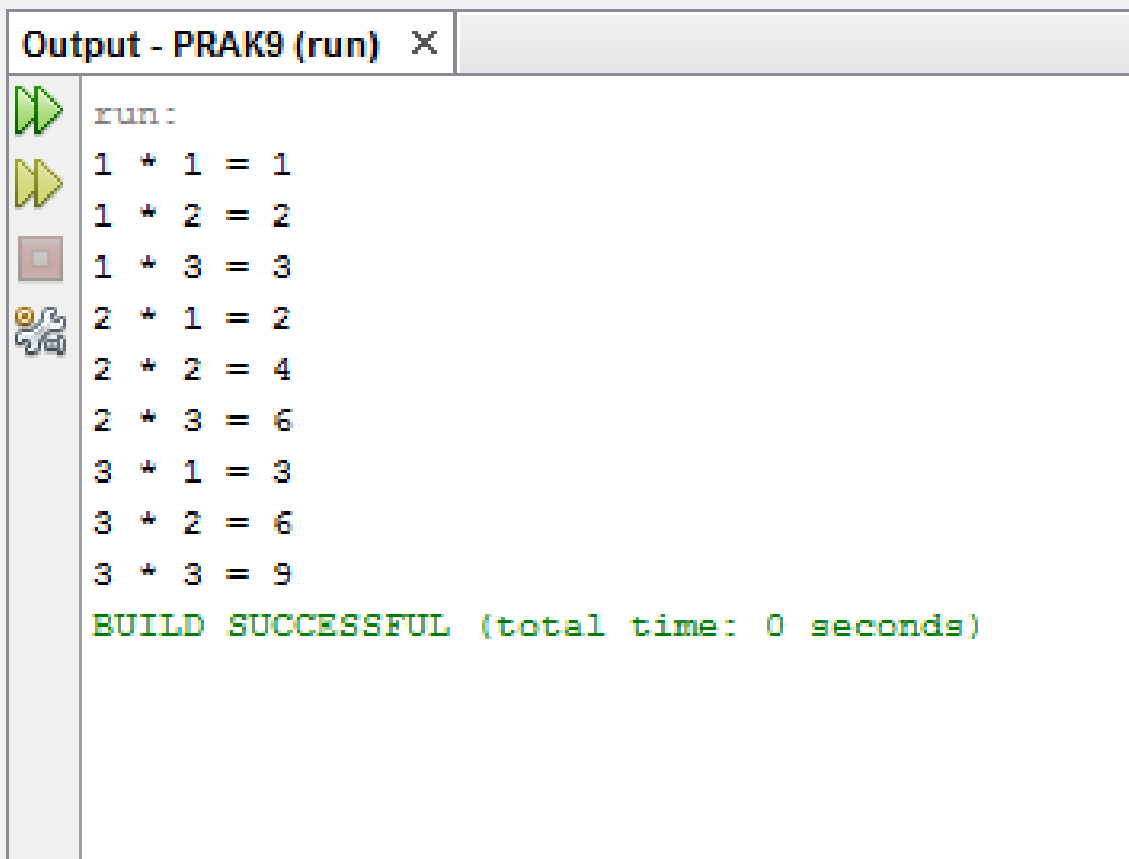


```
Output - PRAK9 (run) ×
run:
1 3 5 7 9
9 7 5 3 1 BUILD SUCCESSFUL (total time: 0 seconds)
```

### Percob5 = NESTED FOR

```
package LOOPING; //berada di package LOOPING
public class percob5 {    //berada di class percob5
    public static void main (String[] args) {
        int i, j, product; //mendeklerasikan tipe data yang digunakan, yaitu int
        for ( i =1; i <= 3; i++) {    // mendefinisakan bahwa nilai dari variabel i awalnya
            adalah angka 1 dan batas angka prosesnya adalah angka kurang dari samadengan 3 dan
            perubahan dari proses per prosesnya adalah menampilkan dari atas kebawah ( + )
                for (j=1; j <= 3; j++) {    // mendefinisakan bahwa nilai dari variabel j
                    awalnya adalah angka 1 dan batas angka prosesnya adalah angka kurang dari samadengan 3
                    dan perubahan dari proses per prosesnya adalah menampilkan dari atas kebawah ( + )
                        product = i * j; // mendeklerasikan rumus
                        System.out.println(i + " * " + j + " = " +product); //menampilkan
output
                    }
                }
            }
        }
    }
```

### Output dari percob5



```
run:
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
BUILD SUCCESSFUL (total time: 0 seconds)
```



### Percob6 = whileDo

```
package LOOPING; // berada di package LOOPING

public class percob6 { //berada di class percob6

    public static void main(String[] args) {

        int i = 0; // mendeklarasikan variabel yaitu int

        while(i < 6){ //mendeklarasikan bahwa projek ini menggunakan while yaitu
//merupakan proses looping dengan seleksi tertentu. Selama seleksi masih
//mempunyai nilai true maka loop akan terus berjalan. Maka dari itu perlu
//ditetapkan variabelnya. Yaitu kurang dari angka 6 variable akan terus diseleksi ketika
//loop berjalan. Selama var_selection menghasilkan nilai true maka statement yang ada di
//dalam block while akan dieksekusi.

            System.out.println(i); // menampilkan output

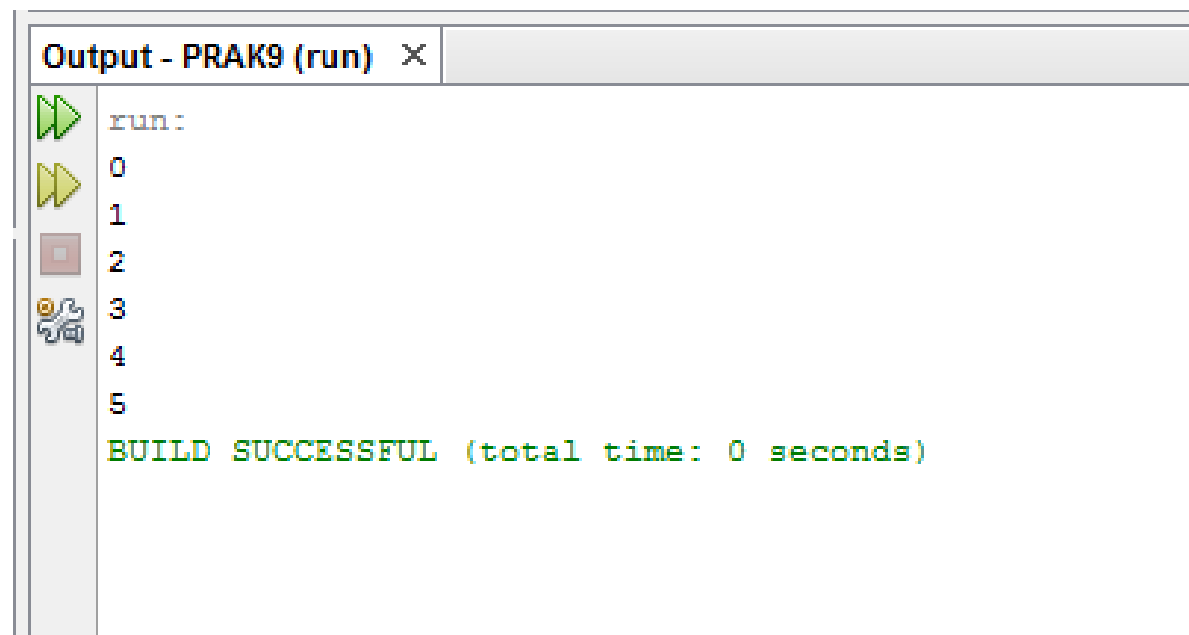
            i++; // menampilkan dengan cara urut dari atas kebawah mulai dari nol bukan 6

        }

    }

}
```

### Hasil run dari percob6



```
Output - PRAK9 (run) X
run:
0
1
2
3
4
5
BUILD SUCCESSFUL (total time: 0 seconds)
```

### Percob7 = doWhile

```
package LOOPING; //berada di package LOOPING

public class percob7 { //berada di class percob7

    public static void main(String[] args) {

        int i = 0; //mendeklarasikan tipe data yaitu int

        do{          //digunakan untuk melakukan looping minimal adalah satu kali
//pengerjaan.

            System.out.println(i); //menampilkan output

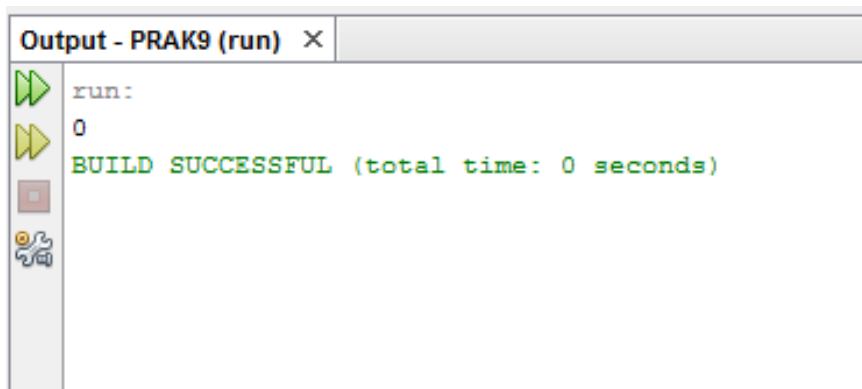
            i++;        //menampilkan hasil lagi dengan cara dari atas kebawah

        }while(i < 0);    // melakukan seleksi Jika menghasilkan true maka statement akan
//dikerjakan ulang, sebaliknya akan keluar dari loop. Karena i < 0 maka yang keluar adalah 0

    }

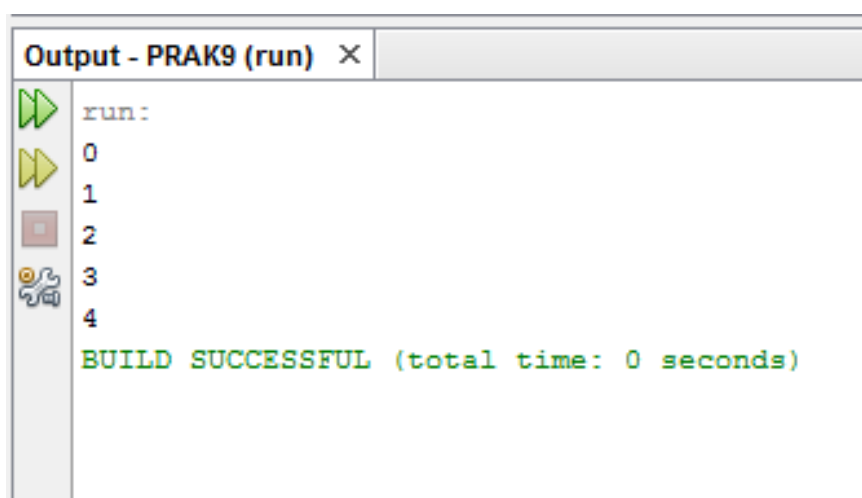
}
```

### Hasil run percob7



```
Output - PRAK9 (run) ×
run:
0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Jika while (i < 5) maka yang tampil adalah



```
Output - PRAK9 (run) ×
run:
0
1
2
3
4
BUILD SUCCESSFUL (total time: 0 seconds)
```

dan lain – lain.

### Percob8 = break nestedfor

```

package LOOPING;

public class percob8 {

    public static void main (String[] args) {

        int i, j, product;

        for ( i =1; i <= 3; i++) { // mendefinisakan bahwa nilai awalnya dari variabel i
                                // adalah 1 dan batasnya adalah kurang dari 3 menggunakan perubahan urut dari atas kebawah

            for (j=1; j <= 3; j++) { // mendefinisakan bahwa nilai awalnya dari variabel j
                                // adalah 1 dan batasnya adalah kurang dari 3 menggunakan perubahan urut dari atas kebawah

                product = i * j; //mendeklerasikan rumus

                if ( j == 2 ) break; // mendeklerasikan bahwa proses ini akan
                                // berakhir jika j==2 dan akan keluar. Jika tidak diberi break, maka proses akan berjalan terus
                                // hingga batas angka terakhir

                System.out.println(i+ " * " + j + " = " +product); //output

            }

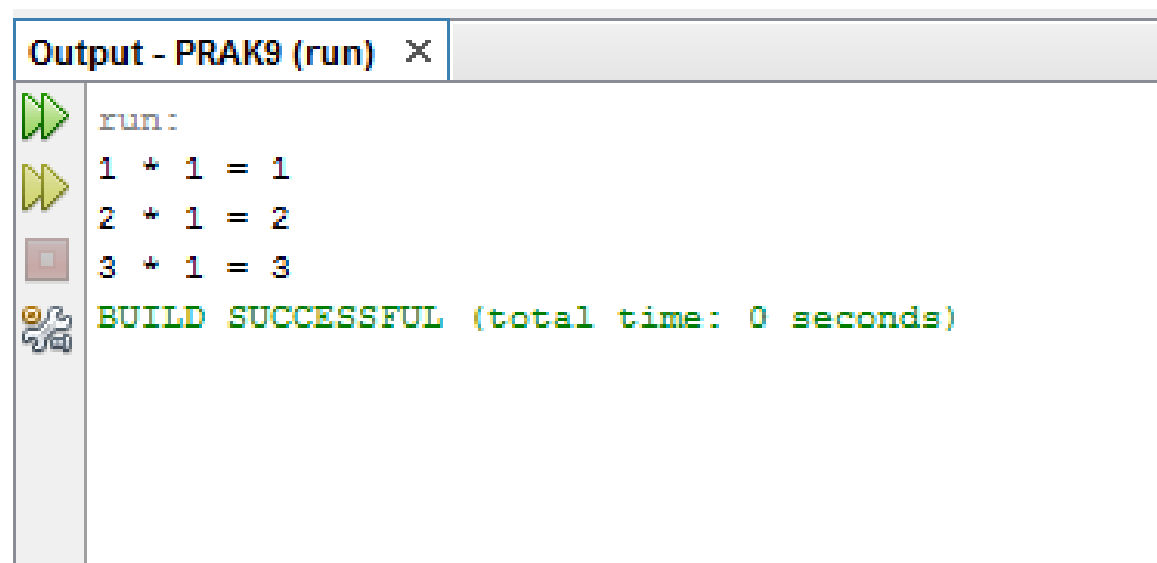
        }

    }

}

```

Hasil run percob8



```

Output - PRAK9 (run) ×
run:
1 * 1 = 1
2 * 1 = 2
3 * 1 = 3
BUILD SUCCESSFUL (total time: 0 seconds)

```

Percob9 = continue nestedfor

```

package LOOPING;

public class percob9 {

    public static void main (String[] args) {

        int i, j, product;

        outer: for ( i=1; i <= 3; i++) {    //dengan menambahkan label pada awal for
        terluar dapat menjadikan perintah break label menghentikan semua pengulangan. Maksudnya
        outer yang atas berfungsi sebagai awal dari proses.

            for (j=1; j <= 3; j++) {

                product = i * j;

                if ( j == 2 ) break outer;    //mendeklarasikan akhiran dari proses jika tidak
                diberi outer dibawah maka akan sama saja, yaitu proses tidak akan berhenti sampai batas
                akhir.

                System.out.println(i+ " * " + j + " = " +product);

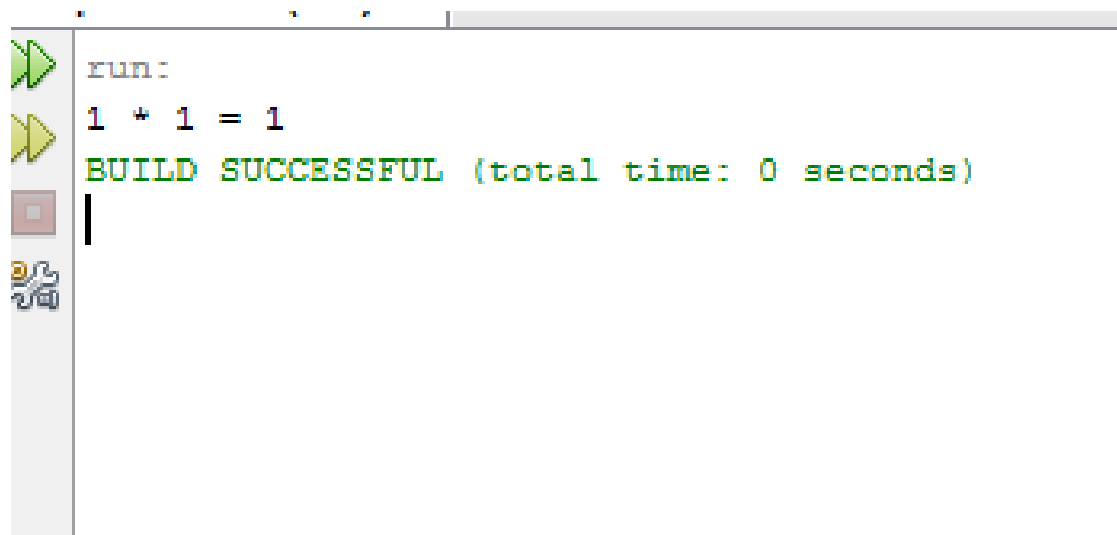
            }

        }

    }
}

```

Hasil run



```

run:
1 * 1 = 1
BUILD SUCCESSFUL (total time: 0 seconds)

```

Percob10 = continue

```

package LOOPING;

public class percob10 {

    public static void main (String[] args) {

        int i, j, product;

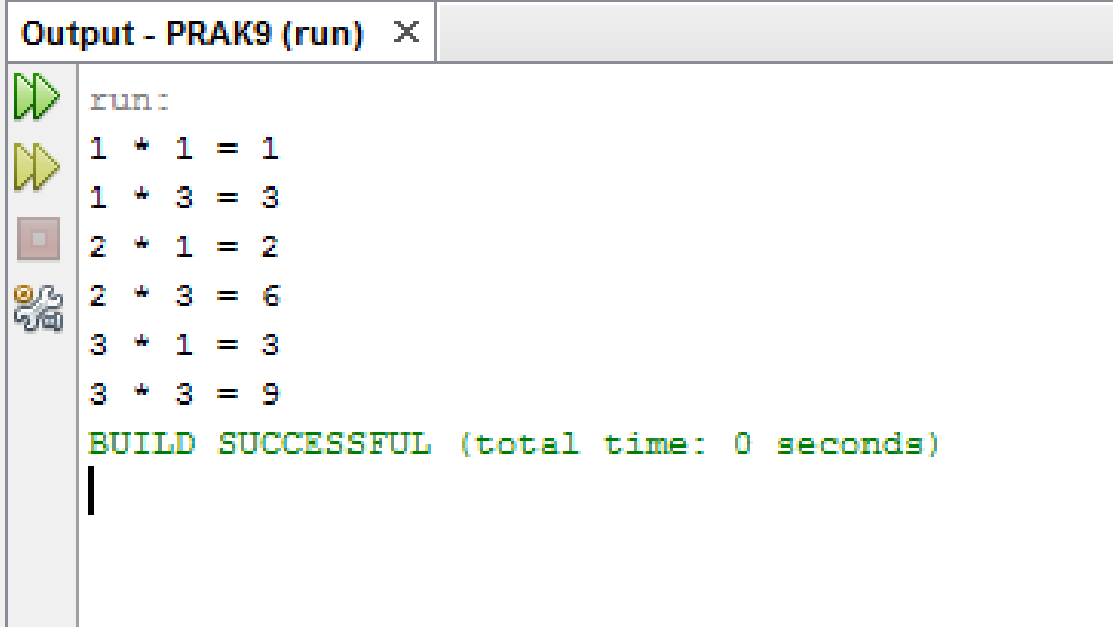
        for ( i =1; i <= 3; i++) {
            for (j=1; j <= 3; j++) {
                product = i * j;

                if ( j == 2 ) continue;    //adalah perintah yang digunakan untuk
                melewati kegiatan yang ada dibawah perintah tersebut dan melanjutkan
                pengulangan. Maksudnya proses tidak akan berhenti jika statemennya adalah
                benar.

                System.out.println(i+ " * " + j + " = " +product);
            }
        }
    }
}

```

Hasil run



```

Output - PRAK9 (run) ×
run:
1 * 1 = 1
1 * 3 = 3
2 * 1 = 2
2 * 3 = 6
3 * 1 = 3
3 * 3 = 9
BUILD SUCCESSFUL (total time: 0 seconds)

```

Percob11 = continue outer

```

package LOOPING;

public class percob12 {

    public static void main (String[] args) {

        int i, j, product;

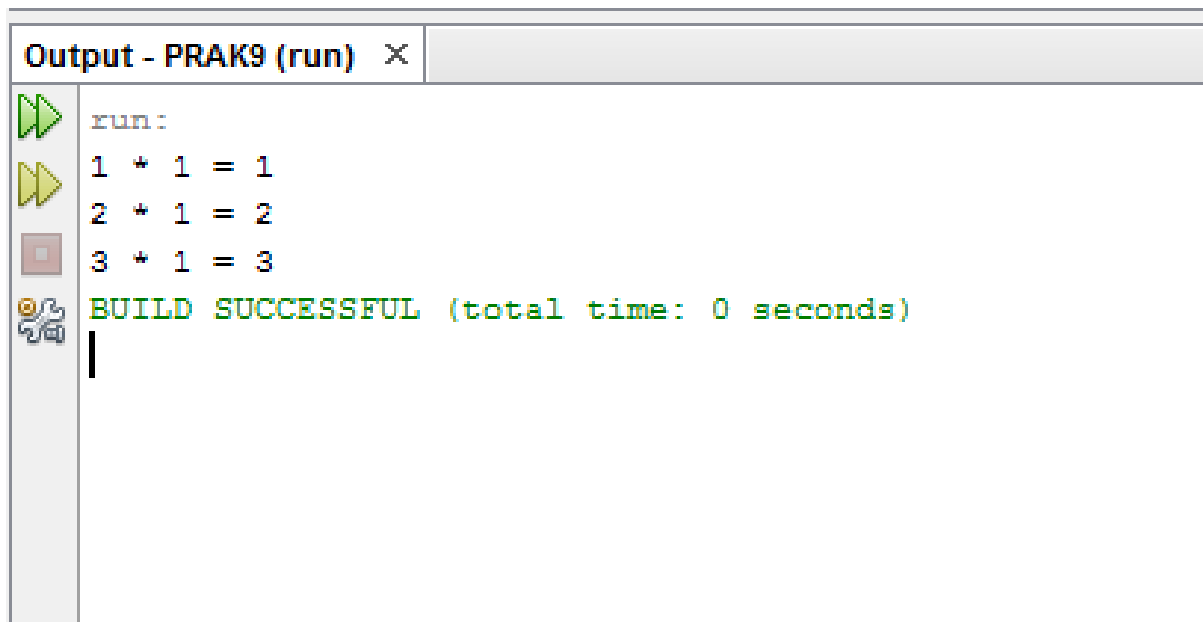
        outer: for ( i =1; i <= 3; i++) {    //awal dari proses outer
            for (j=1; j <= 3; j++) {
                product = i * j;

                if ( j == 2 ) continue outer;    //dengan menambahkan label pada
                //awal for terluar dapat menjadikan perintah continue label melanjutkan
                //pengulangan ke for terluar. Jika outer dipadukan dengan continue maka proses
                //akan berakhir jika batas angkanya sudah terpenuhi

                System.out.println(i+ " * " + j + " = " +product);
            }
        }
    }
}

```

Hasil run



```

Output - PRAK9 (run) ×
run:
1 * 1 = 1
2 * 1 = 2
3 * 1 = 3
BUILD SUCCESSFUL (total time: 0 seconds)

```

## Kesimpulan :

Dari praktikum yang sudah saya lakukan saya dapat menganalisa bahwa kita bisa menggunakan looping atau pengembalian data. For digunakan untuk digunakan untuk melakukan perulangan dengan nilai seleksi adalah sebuah angka numeric. Bisa berupa bilangan bulat maupun pecahan. While digunakan untuk penyeleksian dari data data. while digunakan untuk proses looping dengan seleksi tertentu. Selama seleksi masih mempunyai nilai true maka loop akan terus berjalan. Break digunakan sebagai proses untuk melakukan pemberhentian dan continue adalah perintah yang digunakan untuk melewati kegiatan yang ada dibawah perintah tersebut dan melanjutkan pengulangan.