



Perulangan 2

Pemrograman Dasar (Minggu ke-11)

Tujuan

Di akhir pertemuan, mahasiswa diharapkan mampu :

- Memahami konsep perulangan bersarang pada pemrograman
- Mengenal sintaks perulangan bersarang
- Mampu menerapkan perulangan bersarang dalam persoalan

Preface

- Dalam konsep dasar perulangan, logika perulangan digunakan untuk melakukan beberapa **proses/statement** program secara **berulang-ulang**, dengan suatu **pola tertentu**.
- **Proses/statement** akan terus dilakukan secara **berulang-ulang**, selama **kondisi perulangan** bernilai **benar/true**. Dan sebaliknya, perulangan akan **berhenti** dan proses/statement tidak akan dieksekusi lagi ketika **kondisi perulangan** bernilai **salah/false**.
- Suatu **kondisi perulangan** diperlukan untuk menentukan apakah suatu perulangan masih akan berlangsung lagi atau harus berhenti.

Definisi

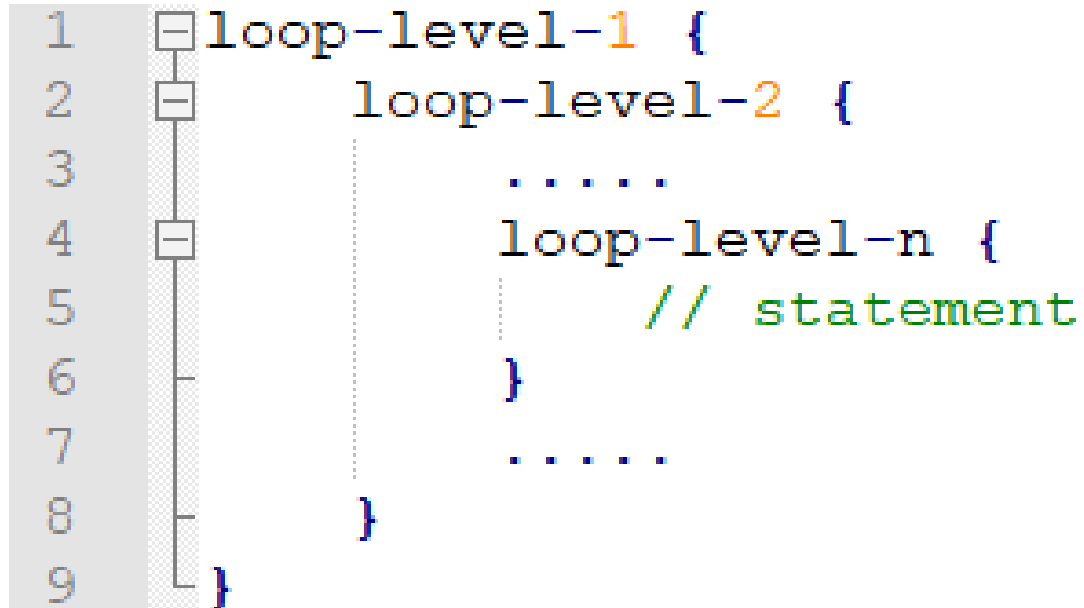
- Perulangan bersarang (*nested loop*) adalah struktur perulangan yang berada di dalam perulangan lainnya.
- *Loop* terluar dikenal dengan istilah ***outer loop***, sedangkan *loop* yang ada di dalamnya disebut ***inner loop***.
- *Nested loop* bisa lebih dari 2 tingkat/level.

Pseudocode Nested Loop

Nested loop bisa memiliki lebih dari 2 tingkat.

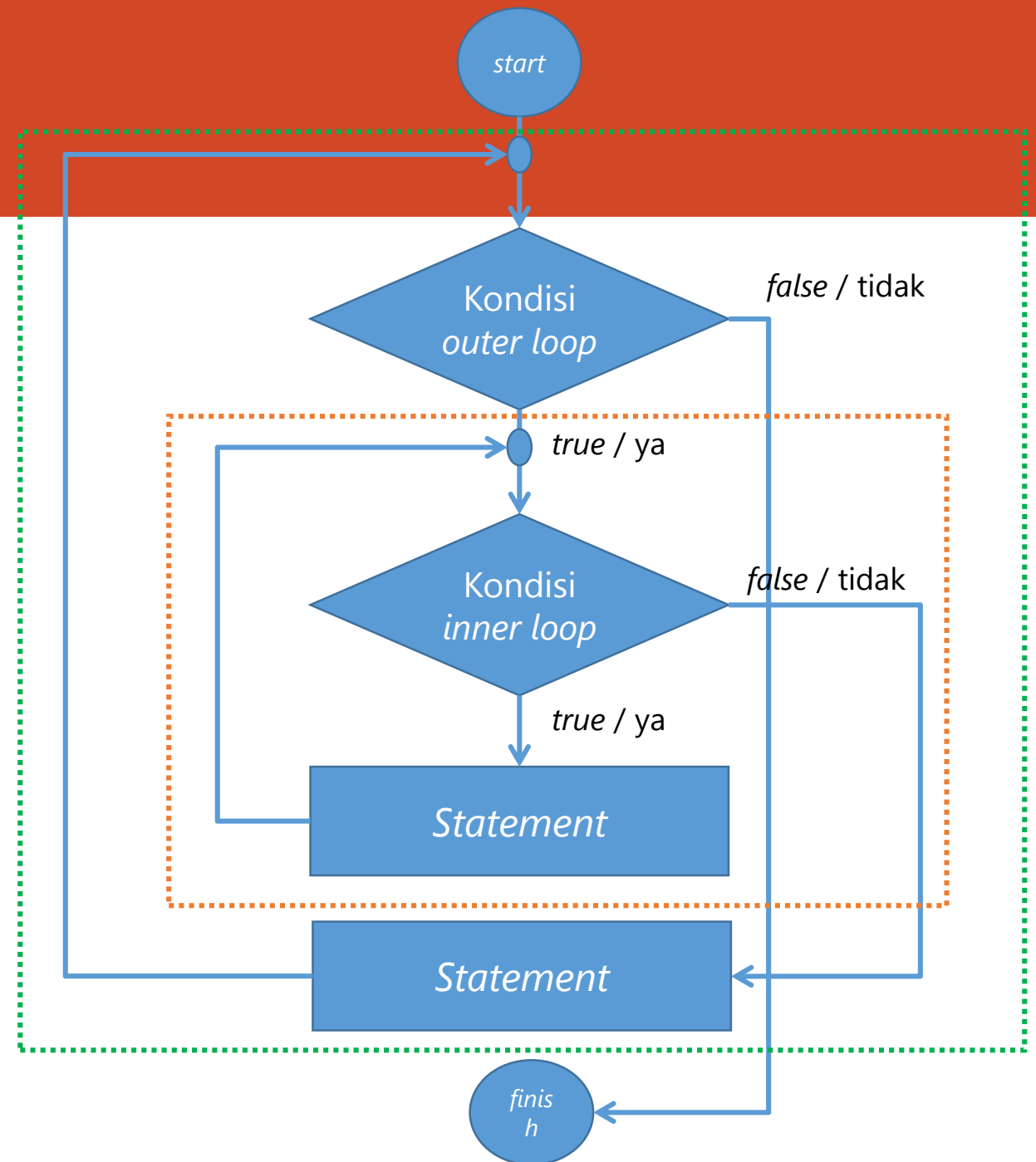
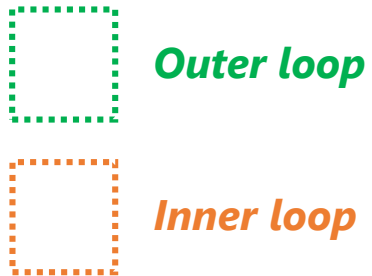
Secara umum gambaran *nested loop* seperti berikut:

```
1  loop-level-1 {  
2      loop-level-2 {  
3          .....  
4          loop-level-n {  
5              // statement  
6          }  
7          .....  
8      }  
9  }
```

The diagram illustrates the structure of a nested loop in pseudocode. It features a vertical list of line numbers from 1 to 9 on the left. To the right of these numbers, the code is formatted with indentation to show nesting levels. Line 1 starts the first loop 'loop-level-1 {'. Line 2 starts the second loop 'loop-level-2 {'. Line 3 shows '.....' representing intermediate code. Line 4 starts the third loop 'loop-level-n {'. Line 5 shows '// statement' as the body of the innermost loop. Line 6 closes the third loop with '}'. Line 7 shows '.....' for code between the second and first loops. Line 8 closes the second loop with '}'. Line 9 closes the first loop with '}'. Vertical lines and horizontal brackets connect the opening and closing braces of each loop level to their respective line numbers, visually demonstrating the nesting.

Flowchart Nested Loop

Secara umum *flowchart* untuk *nested loop* seperti pada gambar:



Sintaks *Nested Loop* : *FOR*

```
1  for (int i = 0; i < n; i++) {    // loop level 1
2
3      for (int j = 0; j < n; j++) {    // loop level 2
4          for (int k = 0; k < n; i++) {    // loop level 3
5              // statement
6          }
7      }
8      for (int l = 0; l < n; l++) {    // loop level 2
9          // statement
10     }
}
```

Outer loop

Inner loop

Sintaks *Nested Loop* : *WHILE*

Outer loop

```
1  int i = 0;
2  // pengecekan loop. Selama kondisi (i < n) bernilai true, loop terus berjalan
3  while (i < n) {      // loop level 1
4      int j = 0;
5
6      // pengecekan loop. Selama kondisi (j < n) bernilai true, loop terus berjalan
7      while (j < n) {    // loop level 2
8          // statement
9          j++;
10     }
11     i++;
12 }
```

Inner loop

Sintaks *Nested Loop* : DO-WHILE

```
1  int i = 0;
2  do {    // loop level 1
3      int j = 0;
4
5      do {    // loop level 2
6          // statement
7          j++;
8
9          // pengecekan loop. Selama kondisi (j < n) bernilai true, loop terus berjalan
10         } while (j < n);
11         i++;
12
13         // pengecekan loop. Selama kondisi (i < n) bernilai true, loop terus berjalan
14     } while (i < n);
```

Outer loop

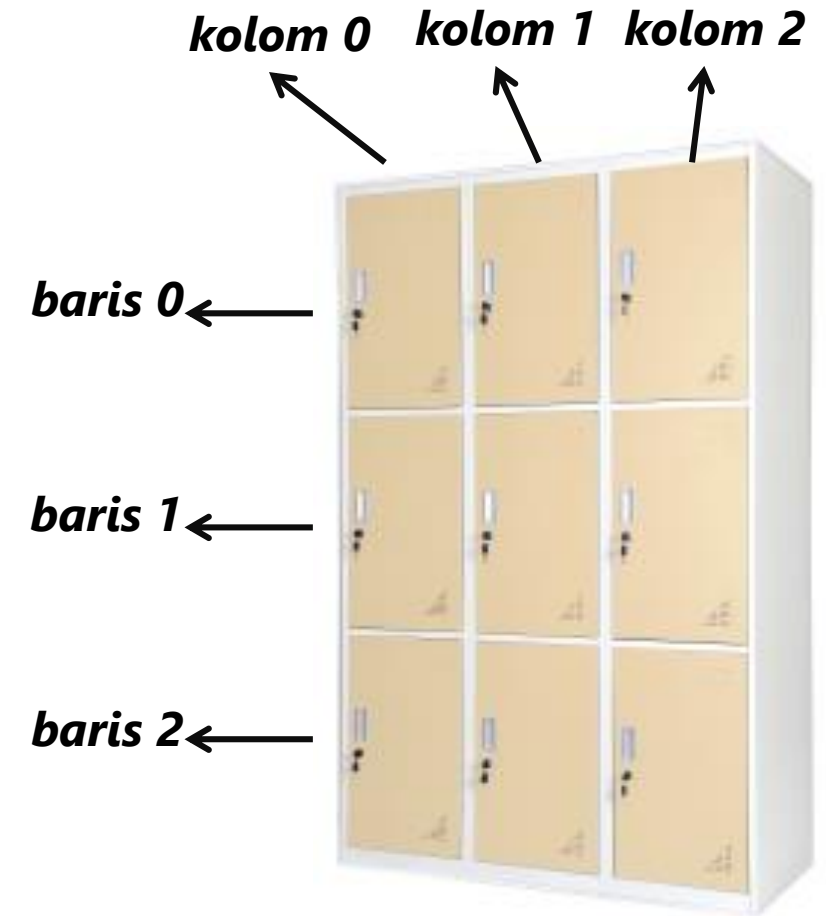
Inner loop

Logika Rak/Loker

Nested loop dengan 2 tingkat/level ibarat seperti loker.

Dimana ***outer loop*** kita identifikasi sebagai penunjuk **baris** dan ***inner loop*** kita identifikasi sebagai penunjuk **kolom**.

```
for(int baris = 0; baris < 3; baris++) {  
    for(int kolom = 0; kolom < 3; kolom++){  
        // statement  
    }  
}
```



Logika Rak/Loker (cont.)

```
for(int baris = 0; baris < 3; baris++) {  
    for(int kolom = 0; kolom < 3; kolom++){  
        System.out.print("Baris-"+baris+" & Kolom-"+kolom+" ");  
    }  
    System.out.println();  
}
```



Kombinasi *Nested Loop*

Nested loop tidak hanya berupa satu jenis *loop*/perulangan yang bertingkat, akan tetapi bisa kombinasi *loop* yang bertingkat.

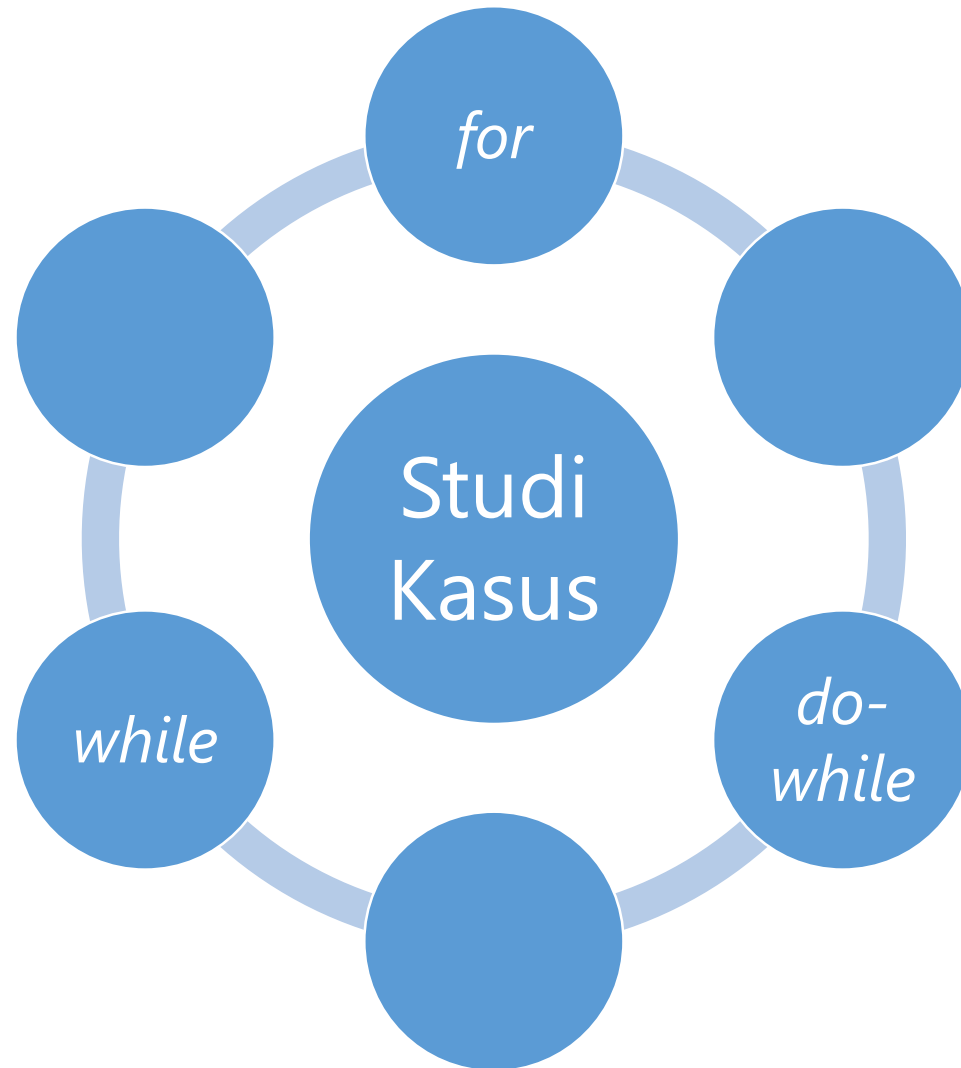
```
/* Kombinasi for dan do-while loop */
for(int i = 0; i < 10; i++){
    int j = 0;
    do {
        // statement
        j++;
    } while(j < 10);
}

/* Kombinasi while dan do-while loop */
int i = 0;
while(i < 10) {
    int j = 0;
    do {
        // statement
        j++;
    } while(j < 10);

    i++;
}
```

```
/* Kombinasi while dan for loop */
int i = 0;
while(i < 10) {
    for(int j = 0; j < 10; j++) {
        // statement
    }
    i++;
}

/* Kombinasi do-while dan for loop */
int i = 0;
do {
    for(int j = 0; j < 10; j++) {
        // statement
    }
    i++;
} while(i < 10);
```



Studi Kasus 1 – *Bintang Persegi*



```
****  
****  
****  
****
```

Bagaimana caranya untuk menampilkan tanda * seperti gambar di atas dengan menggunakan nested loop?

Baik menggunakan *nested loop* **for**, **while**, **do-while**?

Studi Kasus 1 – Logika Jawaban

- Program yang akan dibuat harus ada *outer loop* dan *inner loop*.
- *Outer loop* digunakan untuk menghitung atau melakukan perulangan sebanyak jumlah baris yaitu 4 baris ($i = 0; i < 4; i++;$).
- Setiap *inner loop* selesai di eksekusi, maka akan dibuatkan baris baru.
- *Inner loop* digunakan untuk mencetak simbol * pada layar, jumlah simbol yang ditampilkan perbarisnya akan menyesuaikan dengan nilai pada variable *j*, yaitu 4 simbol *. ($j = 0; j < 4; j++;$).

Studi Kasus 1 - FOR

1
2
3
4
5
6
7
8
9
10

```
public class StudiKasus1 {  
    public static void main(String args[]) {  
        for (int i = 0; i < 4; i++) {  
            for (int j = 0; j < 4; j++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```

Inner loop menghasilkan satu baris bintang (dengan 4 bintang pada satu baris)

Outer loop mengulang output dari *inner loop*

Studi Kasus 1 - *While*

```
1 public class StudiKasus1 {  
2     public static void main(String args[]) {  
3         int i = 0;  
4         while (i < 4) {  
5             int j = 0;  
6             while (j < 4) {  
7                 System.out.print("*");  
8                 j++;  
9             }  
10            System.out.println();  
11            i++;  
12        }  
13    }  
14 }  
15 }
```

Inner loop menghasilkan satu baris bintang
(dengan 4 bintang pada satu baris)

Outer loop mengulang *output* dari *inner loop*

Studi Kasus 1 - *do-while*

```
1 public class StudiKasus1 {  
2     public static void main(String args[]) {  
3         int i = 0;  
4         do {  
5             int j = 0;  
6             do {  
7                 System.out.print("*");  
8                 j++;  
9             }while (j < 4);  
10            System.out.println();  
11            i++;  
12        } while (i < 4);  
13    }  
14 }  
15  
16
```

Inner loop menghasilkan satu baris bintang
(dengan 4 bintang pada satu baris)

Outer loop mengulang output dari inner loop

Studi Kasus 2 – *Bintang Segitiga*

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```



Bagaimana caranya untuk menampilkan tanda * seperti gambar di atas dengan menggunakan nested loop?

Dimana jumlah bintang dalam satu baris, tergantung nomor baris tersebut (total 10 baris).

Studi Kasus 2 – Logika Jawaban

- Program yang akan dibuat harus ada ***outer loop*** dan ***inner loop***.
- ***Outer loop*** digunakan untuk menghitung atau melakukan perulangan sebanyak jumlah baris yaitu 10 baris (`i = 0; i < 10; i++;`). Setiap *inner loop* selesai di eksekusi, maka akan dibuatkan baris baru.
- ***Inner loop*** digunakan untuk mencetak simbol * pada layar, jumlah simbol yang ditampilkan perbarisnya akan **menyesuaikan (sama dengan)** dengan nilai/nomor baris pada **baris i**. (`j = 0; j <= i; j++;`).

Studi Kasus 2 - FOR


```
1 public class StudiKasus2 {  
2     public static void main(String args[]) {  
3         for(int i = 0; i < 10; i++) {  
4             for(int j = 0; j <= i; j++) {  
5                 System.out.print("*");  
6             }  
7             System.out.println();  
8         }  
9     }  
10 }
```

Inner loop menghasilkan satu baris bintang yang jumlahnya **menyesuaikan** dengan nilai/nomor baris pada **baris i**

Outer loop mengulang *output* dari *inner loop*


Studi Kasus 2 - While

```
1 public class StudiKasus2 {  
2     public static void main(String args[]) {  
3         int i = 0;  
4         while(i < 10) {  
5             int j = 0;  
6             while(j <= i){  
7                 System.out.print("*");  
8                 j++;  
9             }  
10            System.out.println();  
11            i++;  
12        }  
13    }  
14 }  
15 }
```



Outer loop mengulang output dari inner loop

*Inner loop menghasilkan satu baris bintang yang jumlahnya **menyesuaikan** dengan nilai/nomor baris pada **baris i***



Studi Kasus 2 – *do-while*

```
1 public class StudiKasus2 {  
2     public static void main(String args[]) {  
3         int i = 0;  
4         do {  
5             int j = 0;  
6             do {  
7                 System.out.print("*");  
8                 j++;  
9             } while(j <= i);  
10  
11             System.out.println();  
12             i++;  
13         } while(i < 10);  
14     }  
15 }
```

Inner loop menghasilkan satu baris bintang yang jumlahnya **menyesuaikan** dengan nilai/nomor baris pada **baris i**

Outer loop mengulang *output* dari *inner loop*