

# Perulangan 2

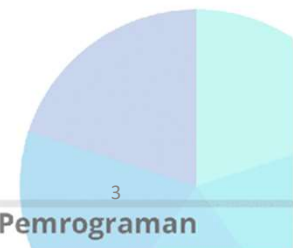
# Tujuan

Di akhir pertemuan, mahasiswa diharapkan mampu :

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

## ***Outline***

- Perulangan Bersarang
- Studi kasus



## ***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**.
- **Kondisi perulangan** (*syarat 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, ***atau***
  - ✓ suatu perulangan yang memiliki perulangan lagi di dalamnya.
- *Loop* terluar dikenal dengan istilah ***outer loop***, sedangkan *loop* yang ada di dalamnya disebut ***inner loop***.
- *Nested loop* bisa lebih dari 2 tingkat/level (*minimal 2 tingkat/level*)



## *Pseudocode Nested Loop*

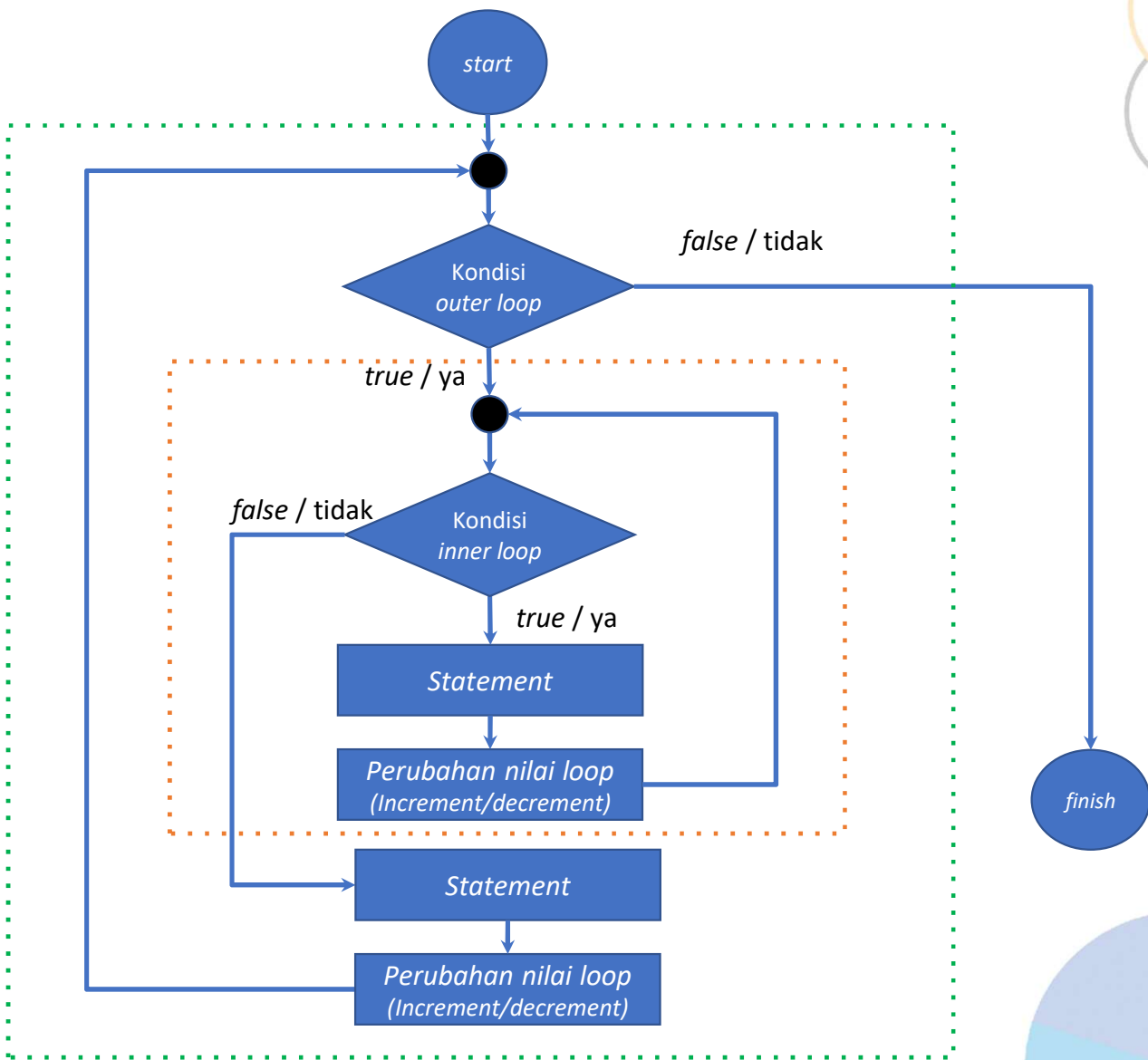
- *Nested loop* bisa memiliki lebih dari 2 tingkat.
- Secara umum gambaran *nested loop* seperti berikut:

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

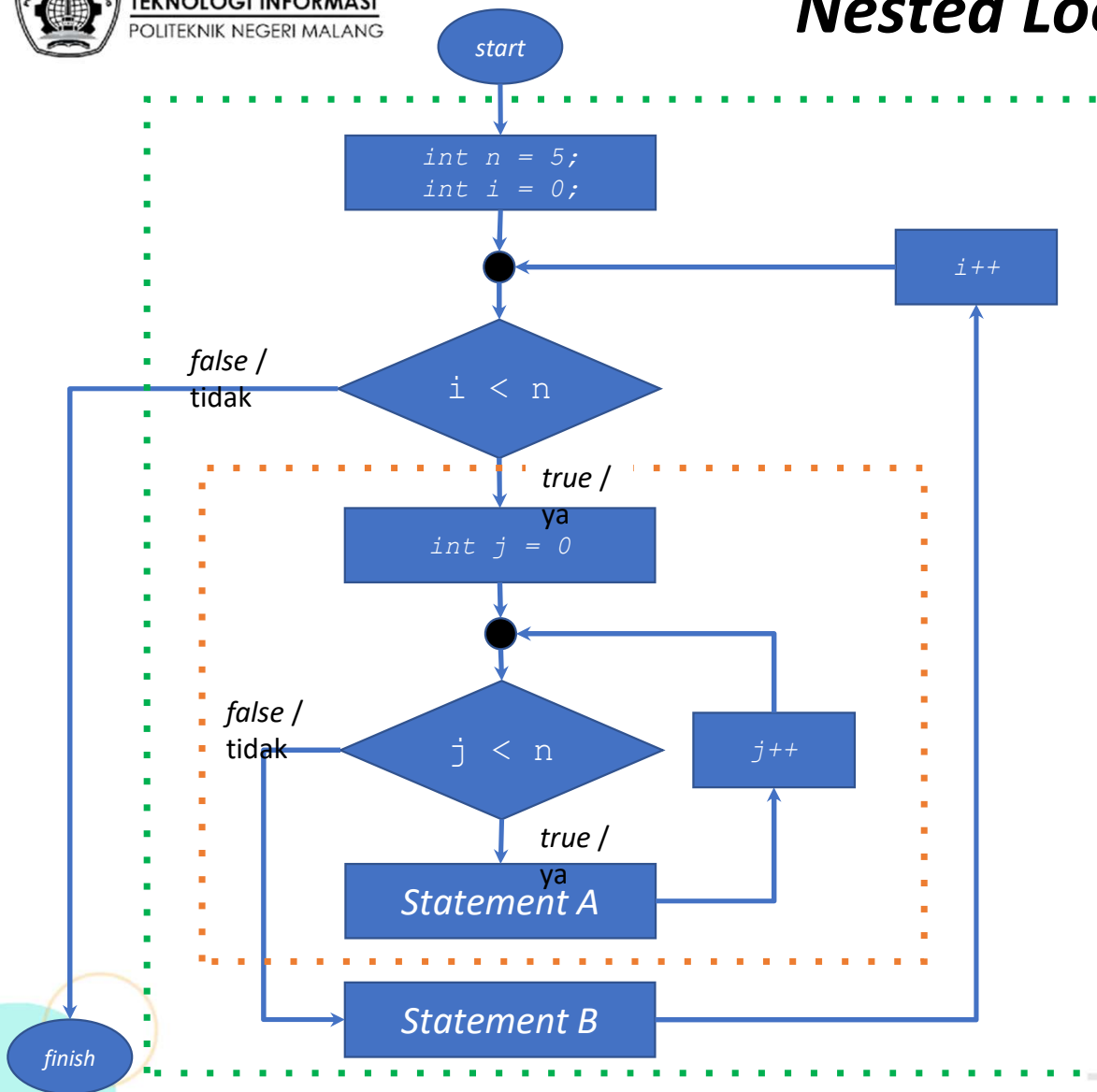


## Flowchart Nested Loop

secara umum *flowchart* untuk *nested loop* seperti pada gambar di samping



# Nested Loop : FOR



```
int n = 5;  
for(int i = 0; i < n; i++) {  
    for(int j = 0; j < n; j++) {  
        // statement A  
    }  
    // statement B  
}
```

Outer loop

Inner loop







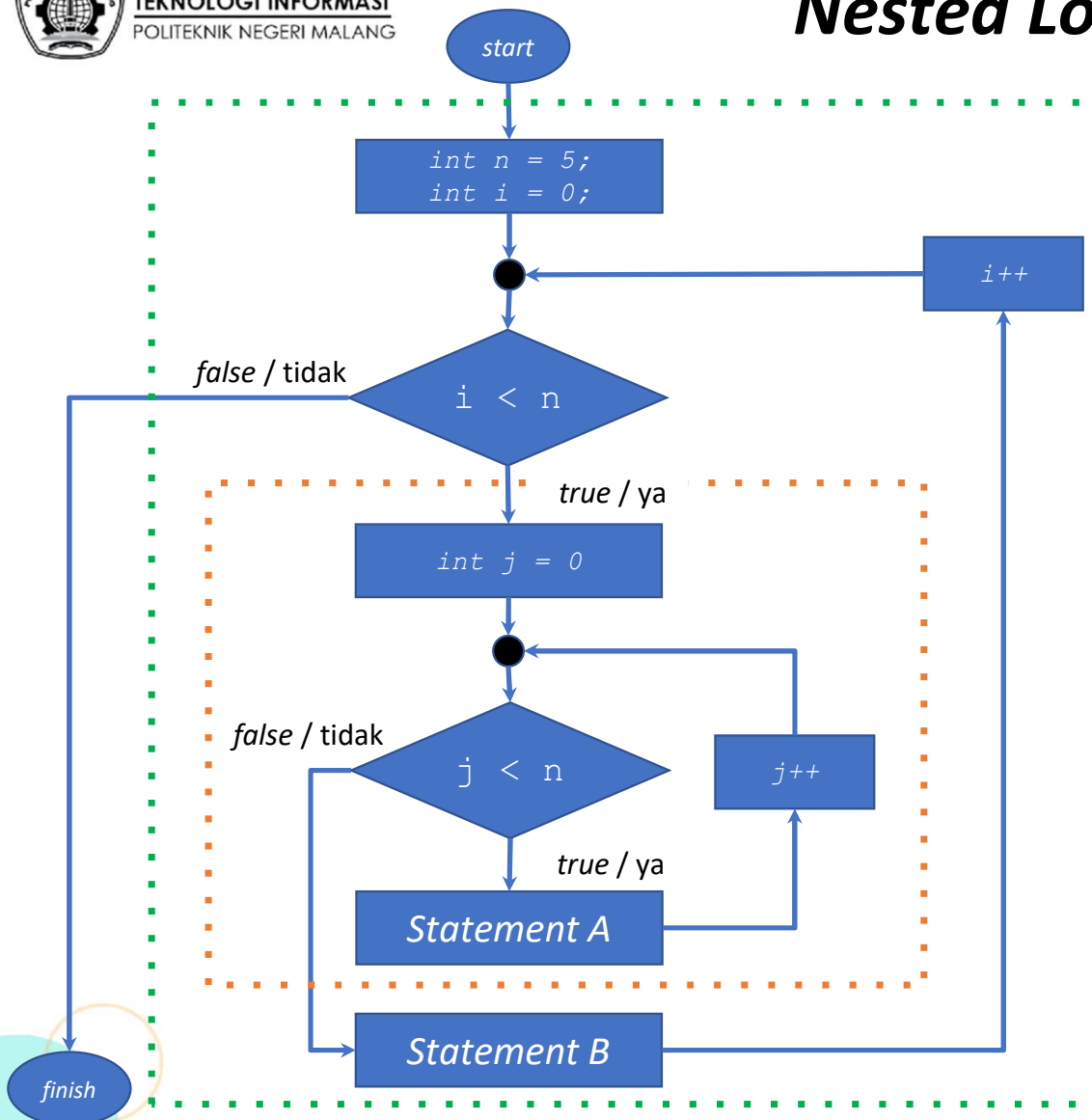
## ***Nested Loop : FOR (lebih dari 2 level)***

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

*Outer loop*

*Inner loop*

# Nested Loop : While



```
int n = 5;
int i = 0;

// pengecekan loop. Loop terus berjalan selama kondisi (i < n) bernilai true
while(i < n){ // loop level 1
    int j = 0;

    // pengecekan loop. Loop terus berjalan selama kondisi (j < n) bernilai true
    while(j < n){ // loop level 2
        // Statement A
        j++;
    }

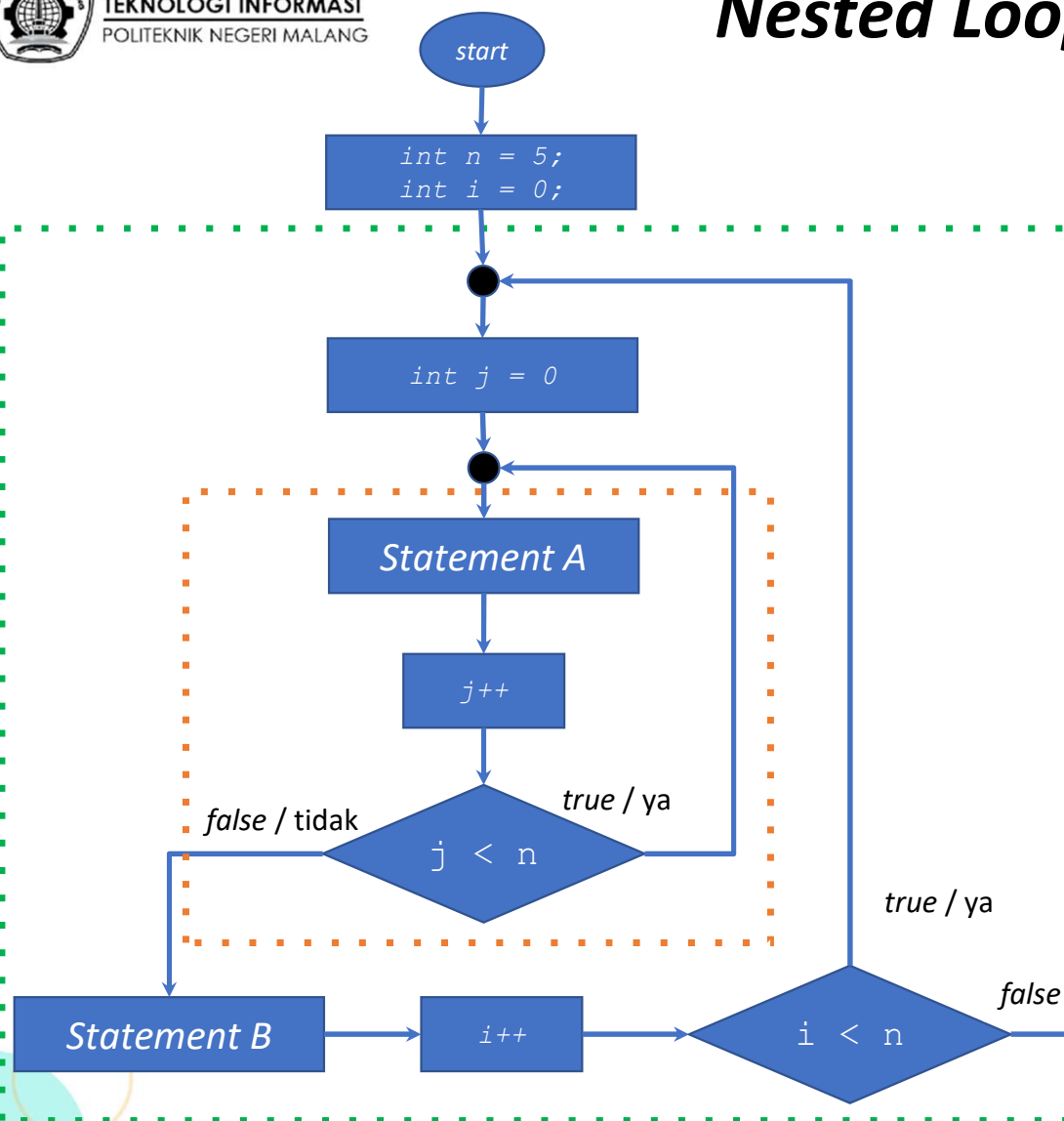
    // Statement B
    i++;
}
```

**Outer loop**

**Inner loop**



# Nested Loop : Do-While



```

int n = 5;
int i = 0;

do{           // loop level 1
    int j = 0;

    do{       // loop level 2
        // Statement A
        j++;

        // pengecekan loop. Loop terus berjalan selama kondisi (j < n) bernilai true
    } while(j < n);

    // Statement B
    i++;

    // pengecekan loop. Loop terus berjalan selama kondisi (i < n) bernilai true
} while(i < n);
    
```

Outer loop

Inner loop

Outer loop

Inner loop

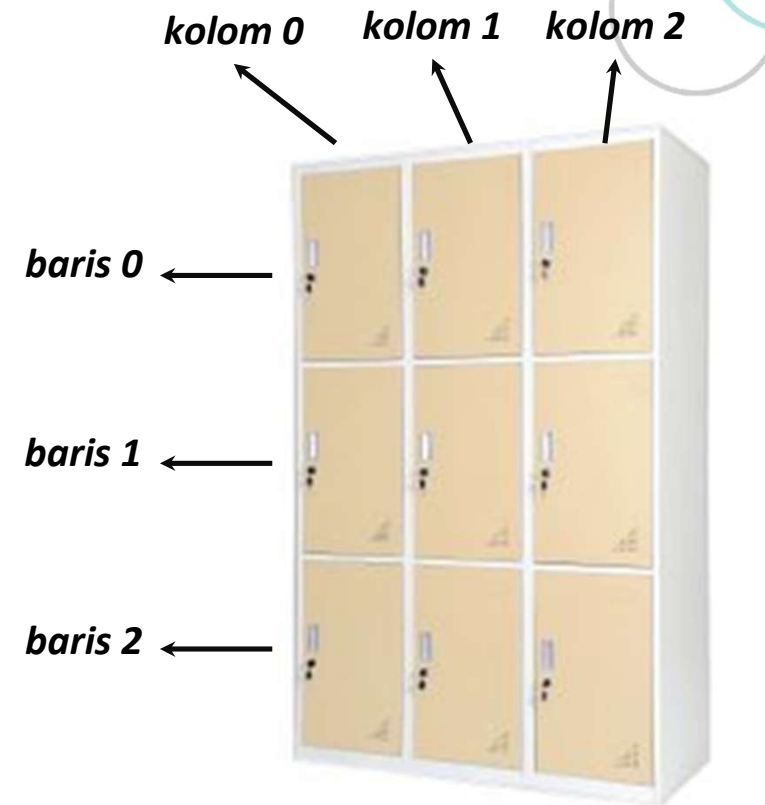


# Logika Rak/Loker

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

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

    i++;
}
```

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



# Nested loop pada Array

- *Nested loop* dalam mengisi nilai array

```
Scanner scanner = new Scanner(System.in);
int bookStore[][] = new int[3][3];
for (int i = 0; i < bookStore.length; i++) {
    System.out.println("Toko ke-" + i);
    for (int j = 0; j < bookStore[0].length; j++) {
        System.out.print("Data ke-" + (j + 1) + ": ");
        bookStore[i][j] = scanner.nextInt();
    }
    System.out.println();
}
```

```
Toko ke-0
Data ke-1: 3
Data ke-2: 4
Data ke-3: 5
```

```
Toko ke-1
Data ke-1: 2
Data ke-2: 5
Data ke-3: 6
```

```
Toko ke-2
Data ke-1: 7
Data ke-2: 8
Data ke-3: 9
```





## Nested loop pada Array(2)

- *Nested loop* dalam menampilkan nilai array

```
for (int i = 0; i < bookStore.length; i++) {  
    System.out.print("Toko ke-" + (i + 1) + ": ");  
    for (int j = 0; j < bookStore[i].length; j++) {  
        System.out.print(bookStore[i][j] + " ");  
    }  
    System.out.println();  
}
```

```
Toko ke-1: 3 4 5  
Toko ke-2: 2 5 6  
Toko ke-3: 7 8 9
```



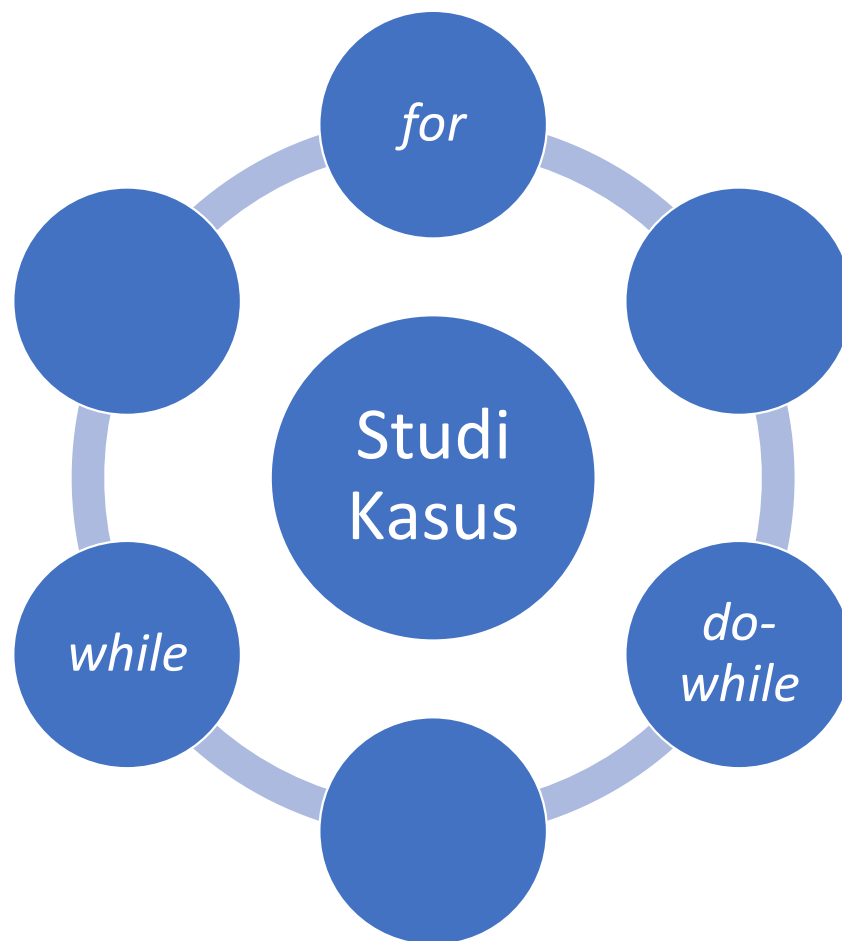


## Nested loop pada Array(3)

- *Nested loop foreach* dalam menampilkan nilai array

```
int i = 1;
for (int categories[] : bookStore) {
    System.out.print("Toko ke-" + i + ": ");
    for (int temp : categories) {
        System.out.print(temp + " ");
    }
    System.out.println();
    i++;
}
```

```
Toko ke-1: 3 4 5
Toko ke-2: 2 5 6
Toko ke-3: 7 8 9
```



## Studi Kasus 1 – *Bintang Persegi*

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



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

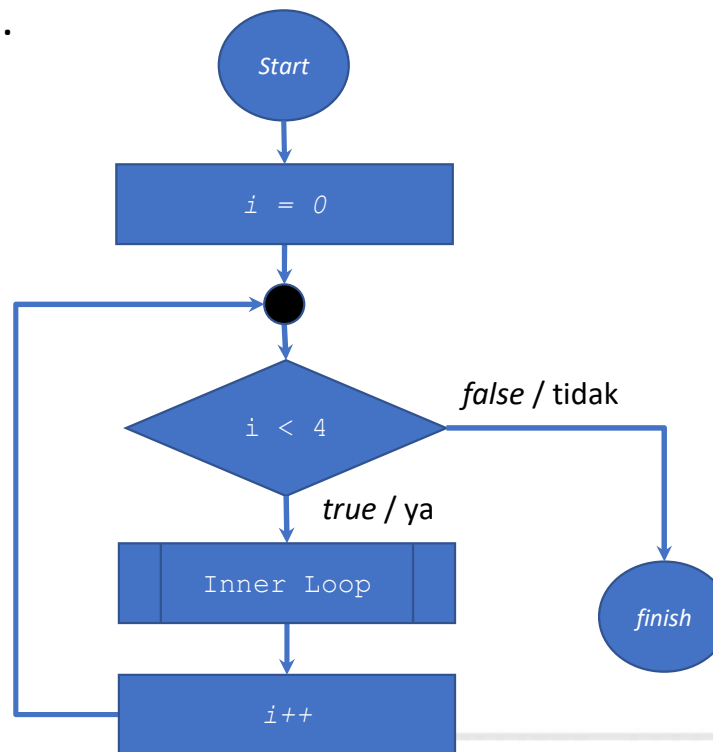
Baik menggunakan *nested loop* **for**, **while**, **do-while**?



## Studi Kasus 1 – Logika Jawaban

Program yang akan dibuat harus ada **outer loop** dan **inner loop**.

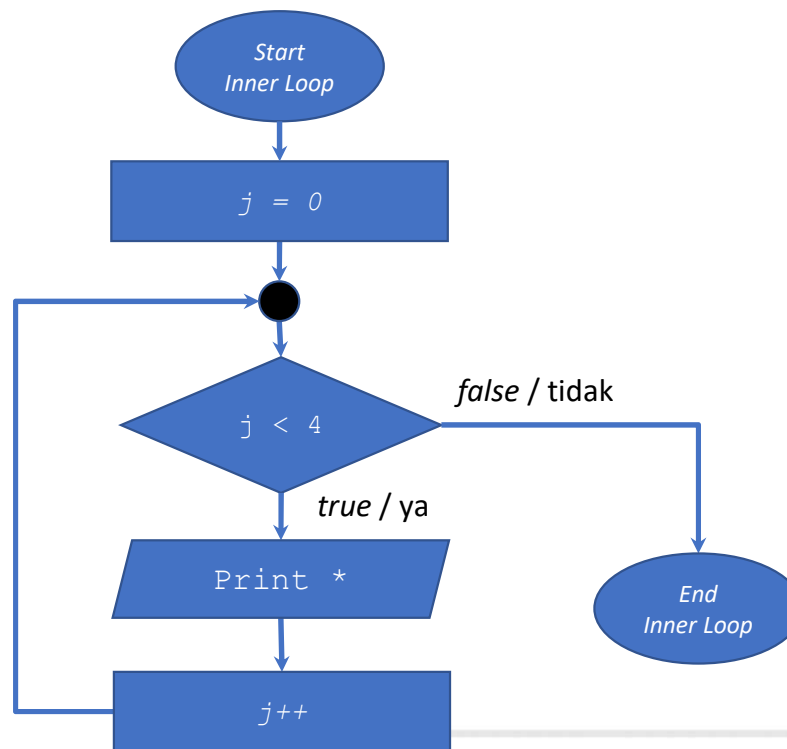
- **Outer loop** digunakan untuk menghitung atau melakukan perulangan sebanyak jumlah baris yaitu 4 baris (`i = 0; i < 4; i++;`). Setiap **inner loop** selesai di eksekusi, maka akan dibuatkan baris baru.





## Studi Kasus 1 – Logika Jawaban (cont.)

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

Outer loop mengulang *output* dari inner loop

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



## Studi Kasus 1 - *While*

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

Outer loop mengulang output dari inner loop

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



## Studi Kasus 1 - *do-while*

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

Outer loop mengulang output dari inner loop

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



## 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 }
```

*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 - 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 }
```

Outer loop mengulang output dari inner loop

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

# Tugas Kelompok

1. Silakan mengidentifikasi dalam bentuk flowchart sesuai dengan project masing-masing untuk fitur yang dikembangkan menggunakan nested loop?
2. Buatlah *flowchart* dari soal gambar di bawah ini  
(boleh pilih salah satu dengan menggunakan *nested loop for/while/do-while*)

a

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

b

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