Pemrograman Berbasis Obyek

Flow Control dan Looping

Oleh Politeknik Elektronika Negeri Surabaya

2017

Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Konten

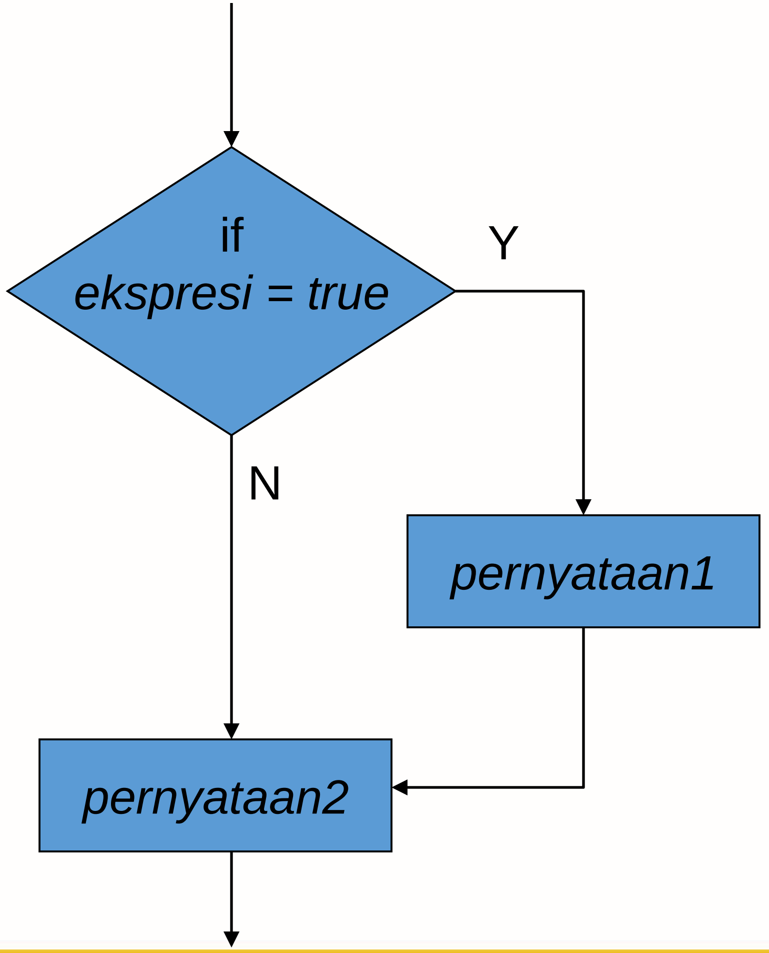
- Selection statements
- Looping statements
- Special loops

The Selection Statements

- *`if`*
- *`if-else`*
- *`else-if`*
- *`switch`*

if

```
if (ekspresi boolean) {
        pernyataan1;
}
pernyataan2;
```



```
public class If {
    public static void main(String args[]) {
        int bilangan = -5;
        if (bilangan<0)
            System.out.println(“Bilangan adalah negatif”);
    }
}
```

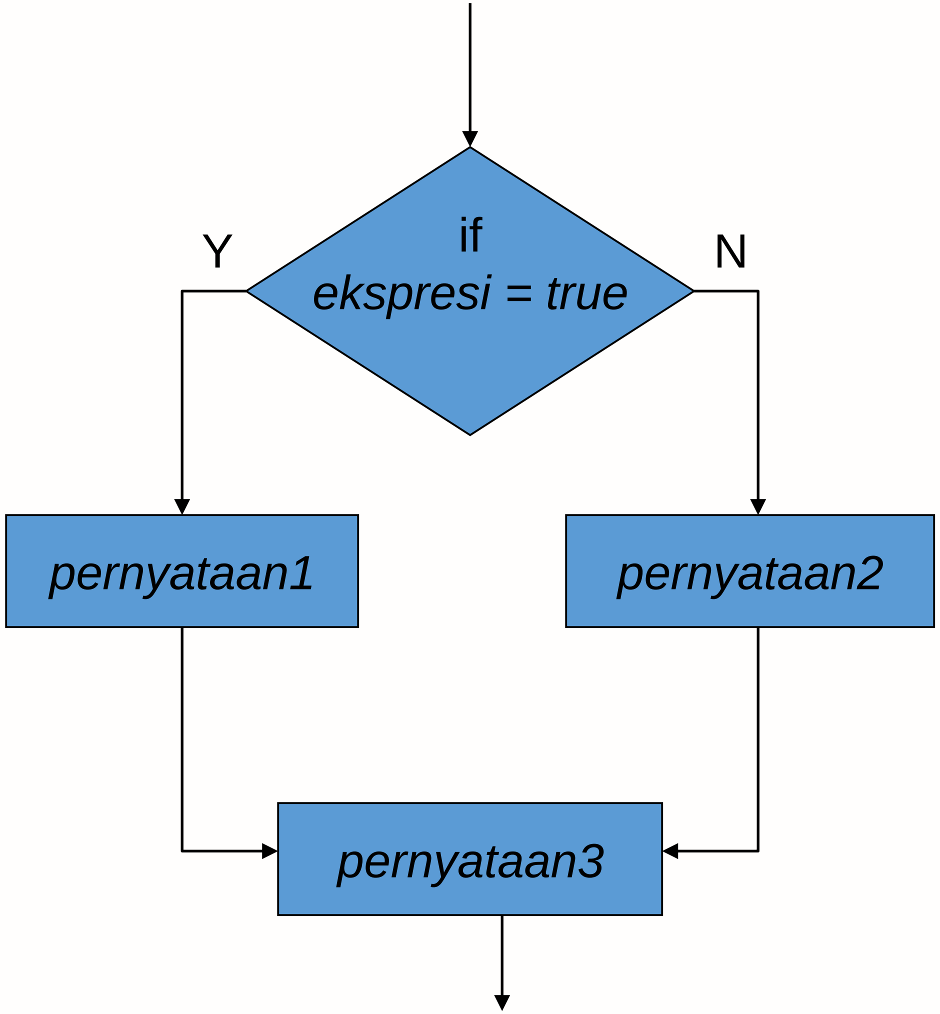
For example, the Bicycle class could allow the brakes to decrease the bicycle's speed *only if* the bicycle is already in motion. One possible implementation of the applyBrakes method could be as follows:

```
  void applyBrakes() {
          if (isMoving){
                    currentSpeed--;
          }
  }
```

```
void applyBrakes() {
          if (isMoving)
                    currentSpeed--;
}
```

if-else

```
if (ekspresi boolean) {
        pernyataan1;
} else {
        pernyataan2;
}
pernyataan3;
```



If - else

- `If()` statement takes a boolean expression, **not a numeric value.**
- You cannot convert or cast boolean types and numeric types.
- If you have:

```
if (x) // x is int
```

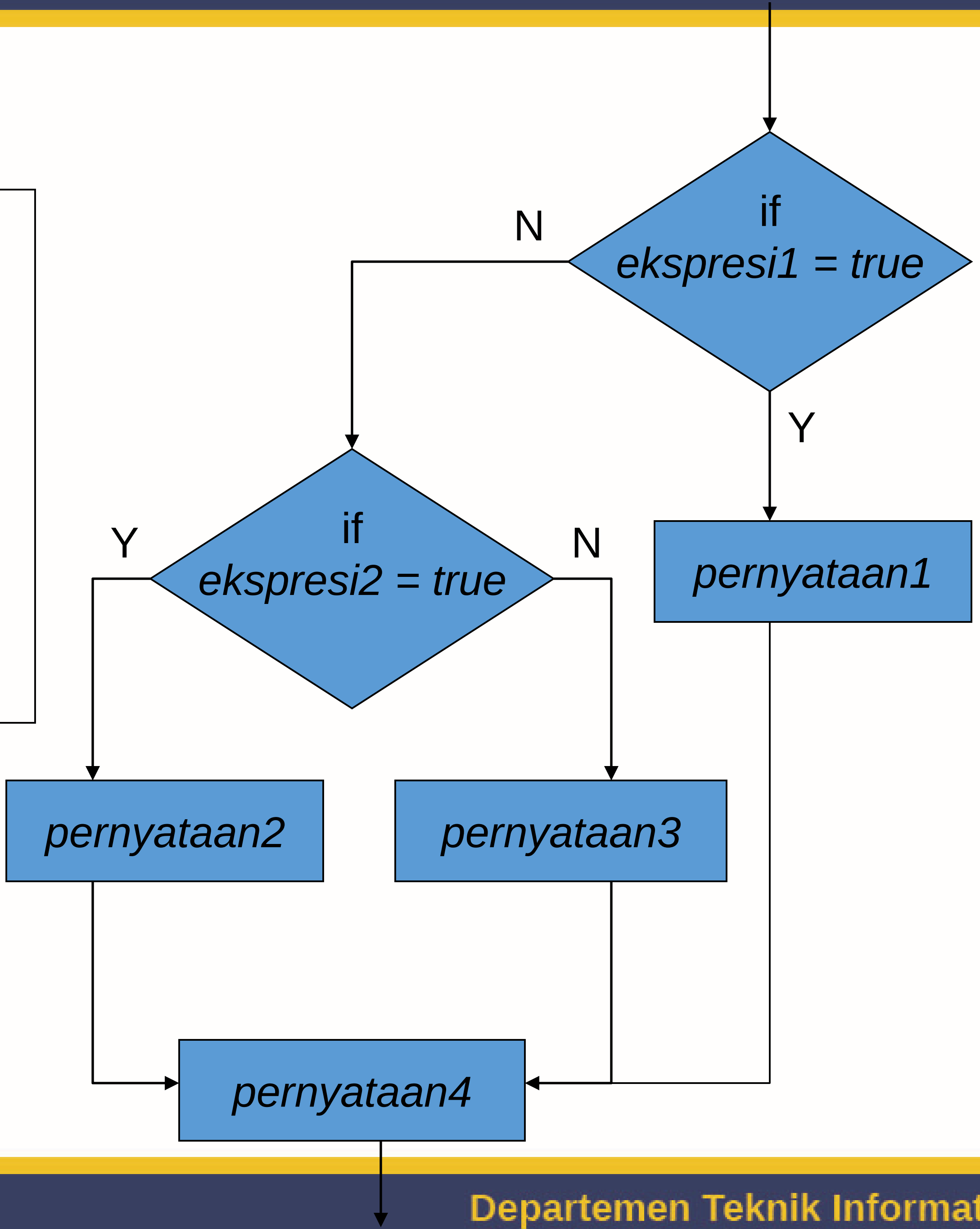
use

```
if (x!=0)
```

```
public class IfElse {
    public static void main(String args[]) {
        int bilangan = -5;
        if (bilangan<0)
             System.out.println(“Bilangan adalah negatif”);
        else
             System.out.println(“Bilangan adalah positif”);
    }
}
```

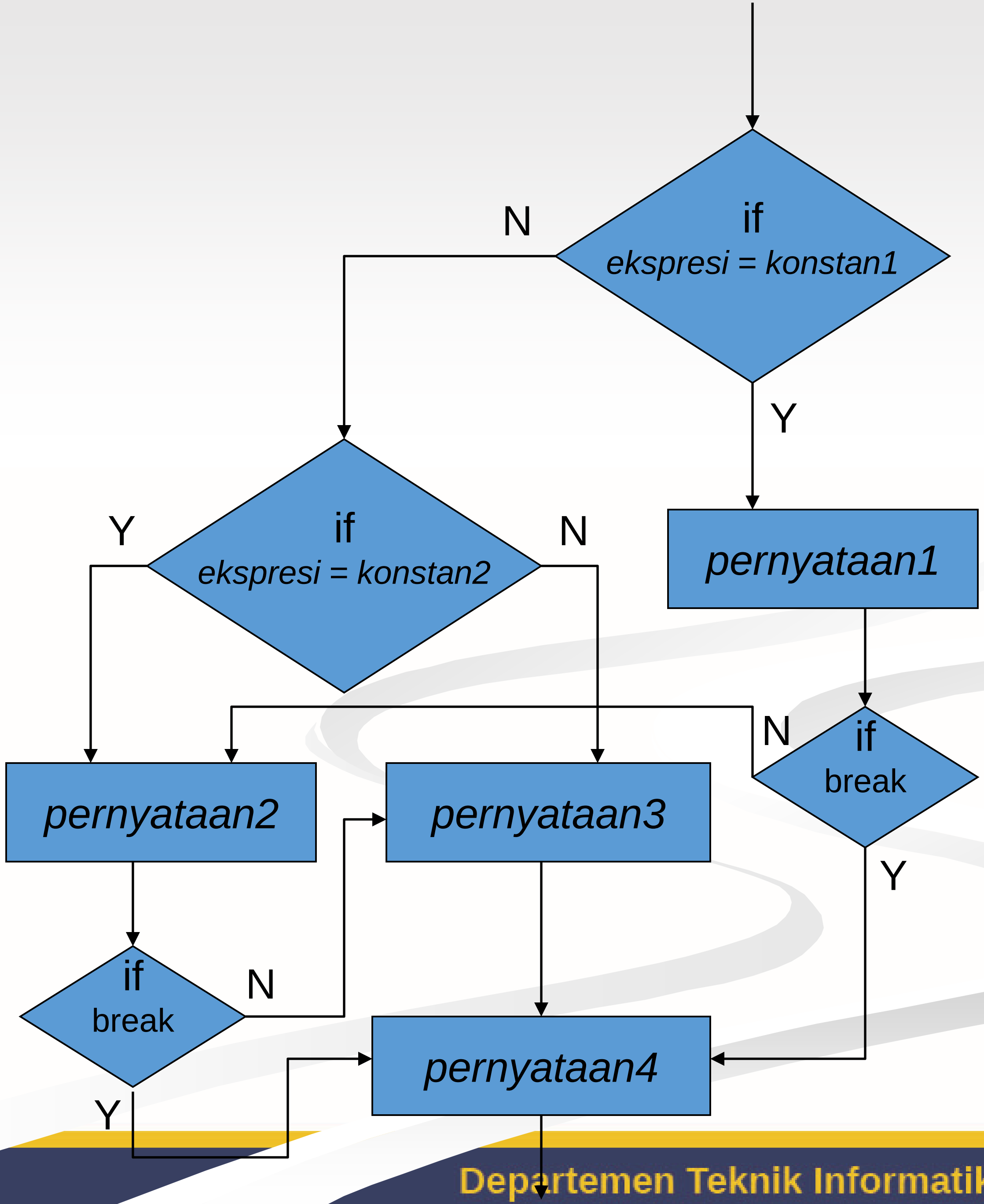
else-if

```
if (ekspresi boolean1) {
    pernyataan1;
} else if (ekspresi boolean2) {
    pernyataan2;
} else {
    pernyataan3;
}
pernyataan4;
```



switch

```
switch (ekspresi) {
     case konstanta1 :
          pernyataan1;
          break;
     case konstanta1:
          pernyataan2;
          break;
     default :
          pernyataan3;
}
pernyataan4;
```



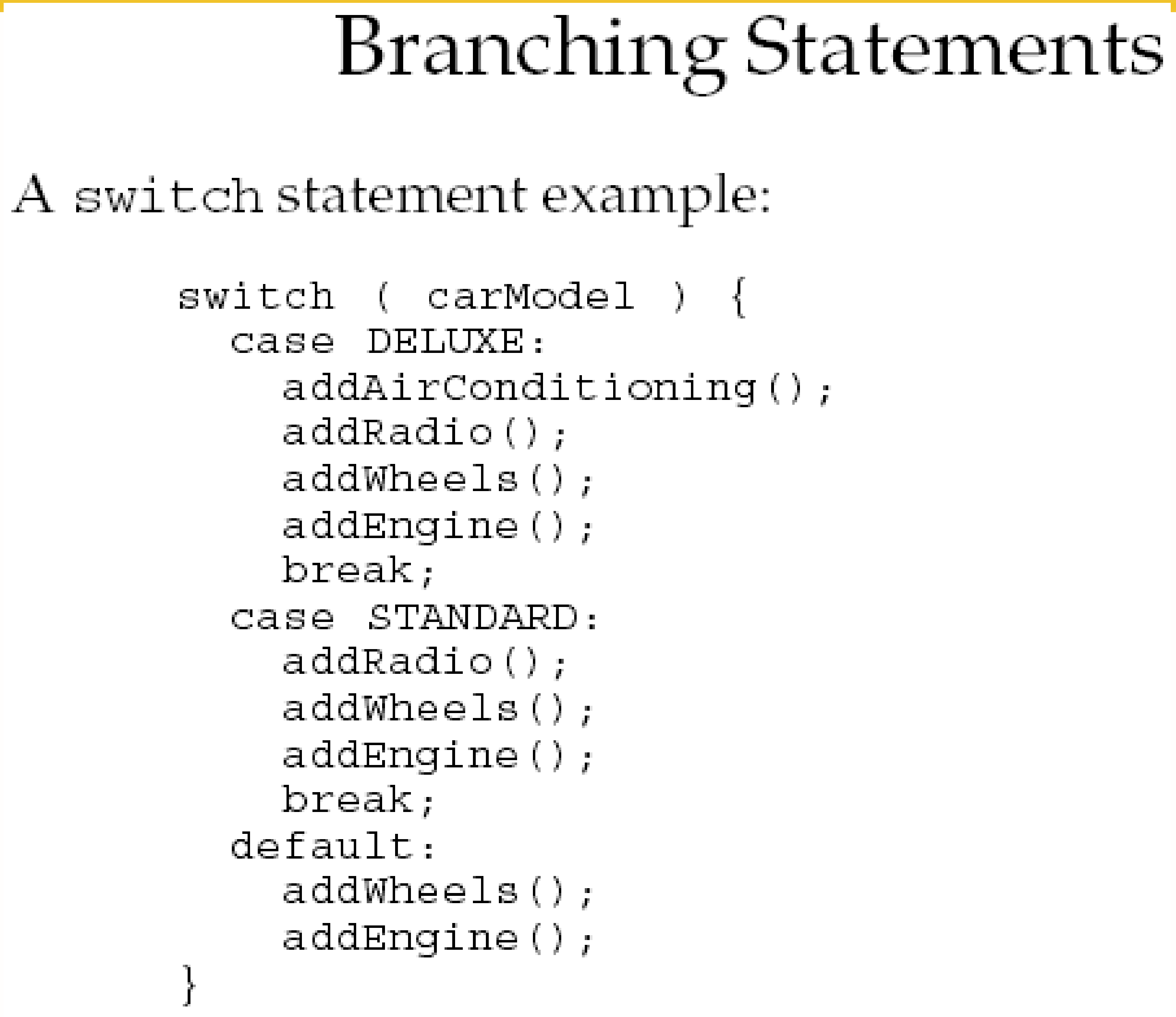
switch(x)

- Variabel x harus bertipe byte, short, char, atau int.
- Floating point, long, atau class references (termasuk String) tidak diperbolehkan.
- In Java SE 7 and later, you can use a String object in the switch statement's expression.
- Kedudukan statement pada default sama dengan kedudukan else pada if-else.

The following code example, StringSwitchDemo, displays the number of the month based on the value of the String named month:

```
public class StringSwitchDemo {

  public static int
getMonthNumber(String month) {

    int monthNumber = 0;


    if (month == null) {

      return monthNumber;

    }


    switch (month.toLowerCase()) {

      case "january":

        monthNumber = 1;

        break;
case "february":

        monthNumber = 2;

        break;

      case "march":

        monthNumber = 3;

        break;
case "april":

        monthNumber = 4;

        break;

      case "may":

        monthNumber = 5;

        break;
case "june":

        monthNumber = 6;

        break;

      case "july":

        monthNumber = 7;

        break;

      case "august":

        monthNumber = 8;

        break;
case "september":

        monthNumber = 9;

        break;
case "october":

        monthNumber = 10;

        break;

      case "november":

        monthNumber = 11;

        break;

      case "december":

        monthNumber = 12;

        break;

      default:

        monthNumber = 0;

        break;

    }

    return monthNumber;

  }
public static void main(String[] args) {

    String month = "August";

    int returnedMonthNumber =
      StringSwitchDemo.getMonthNumber(month);

    if (returnedMonthNumber == 0) {
      System.out.println("Invalid month");
    } else {
      System.out.println(returnedMonthNumber);
    }
  }
}
```

The output from this code is 8.



Branching Statements

A switch statement example:

```
switch ( carModel ) {
  case DELUXE:
    addAirConditioning();
    addRadio();
    addWheels();
    addEngine();
    break;
  case STANDARD:
    addRadio();
    addWheels();
    addEngine();
    break;
  default:
    addWheels();
    addEngine();
}
```

Branching Statements

A switch statement example:

```
switch ( carModel ) {
  case THE_WORKS:
    addGoldPackage();
    add7WayAdjustableSeats();
  case DELUXE:
    addFloorMats();
    addAirConditioning();
  case STANDARD:
    addRadio();
    addDefroster();
  default:
    addWheels();
    addEngine();
}
```

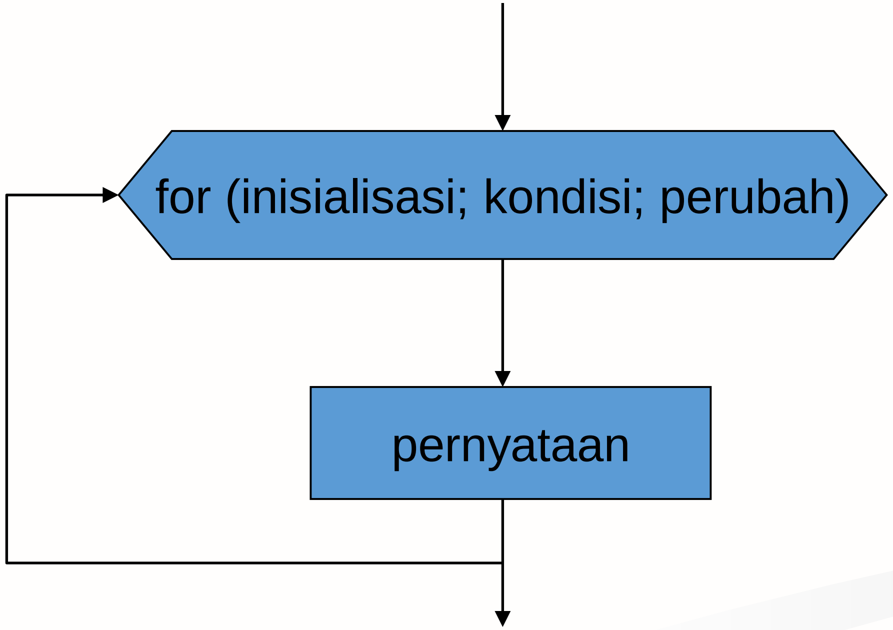
```
public class Switch {
    public static void main(String args[]) {
        int i=2;
        switch (i) {
            case 1 : i+=3;
                     break;
            case 2 : i+=5;
                     break;
            default: i+=10;
        }
        System.out.println(i);
    }
}
```

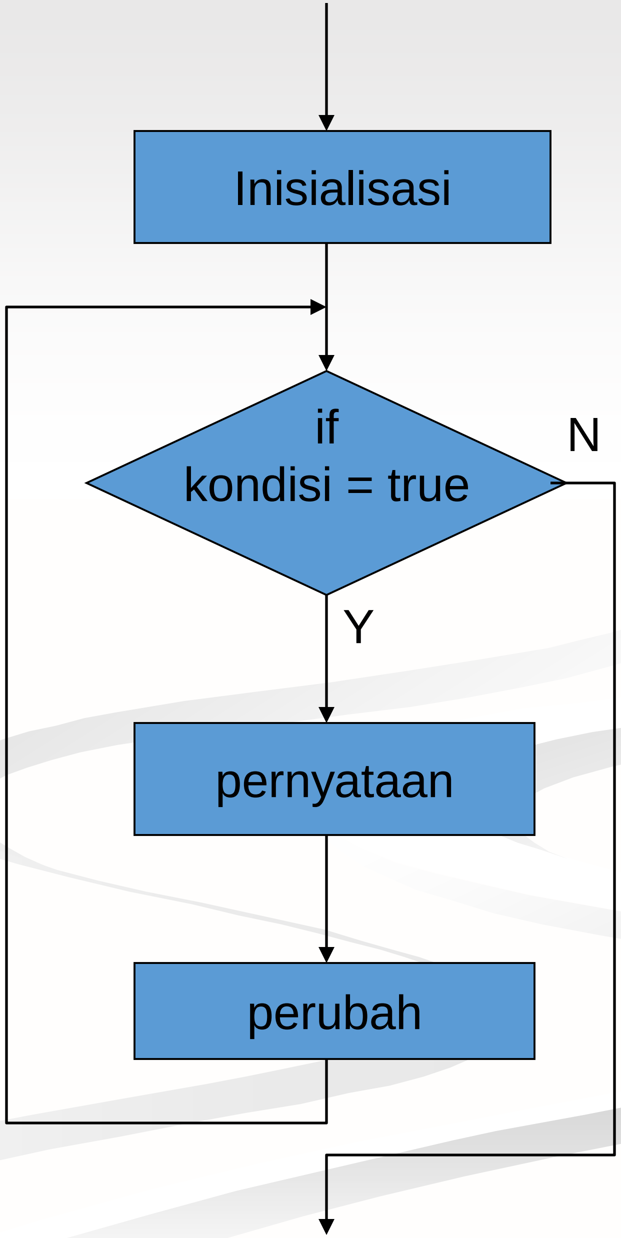
The Loop Statements

- The `for()` Loop
- The `while()` Loop
- The `do – while()` Loop

for

```
for (inisialisasi; ekspresi boolean;
perubah) {
            pernyataan;
}
```





for

- Java programming language allows the comma separator in a for() loop structure.
- Example:

```
for (i=0, j = 0; j<10; i++, j++){}
for (int i=0; i<10; i++) {
        System.out.println(“Hore !!”);
}
```

```
for (int i = 0; i < 10; i++) {
    System.out.println("Are you finished yet?");
}
System.out.println("Finally!");
```

Mengakses elemen array dengan for

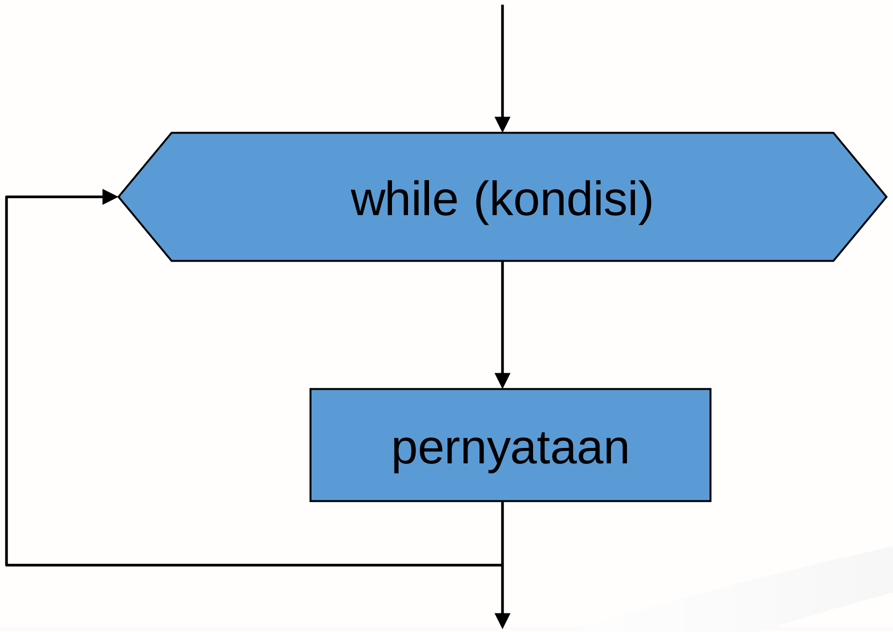
```
class EnhancedForDemo {
    public static void main(String[] args){
            int[] numbers = {1,2,3,4,5,6,7,8,9,10};
            for (int item : numbers) {
            System.out.println("Count is: " + item);                }
    }
}
```

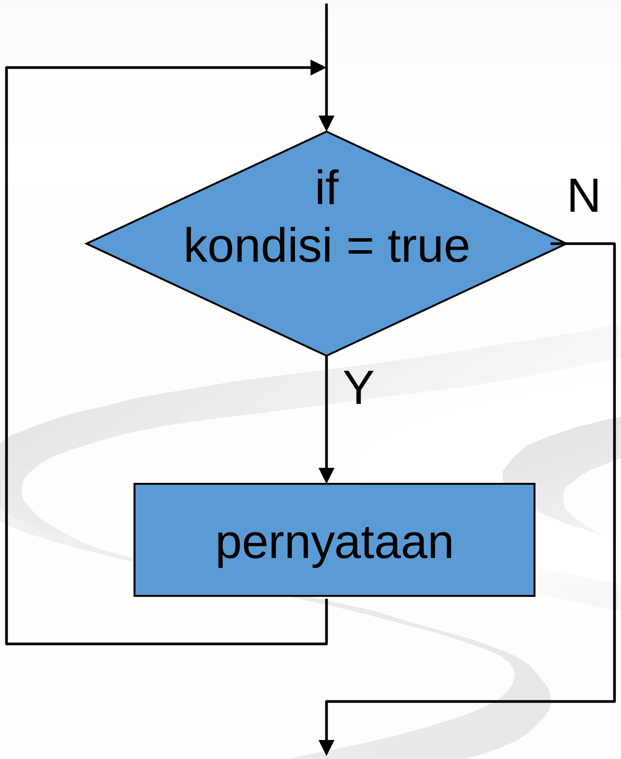
- The output of this program is:

```
Count is: 1
Count is: 2
Count is: 3
Count is: 4
Count is: 5
Count is: 6
Count is: 7
Count is: 8
Count is: 9
Count is: 10
```

while

```
while (ekspresi boolean) {
        pernyataan;
}
```



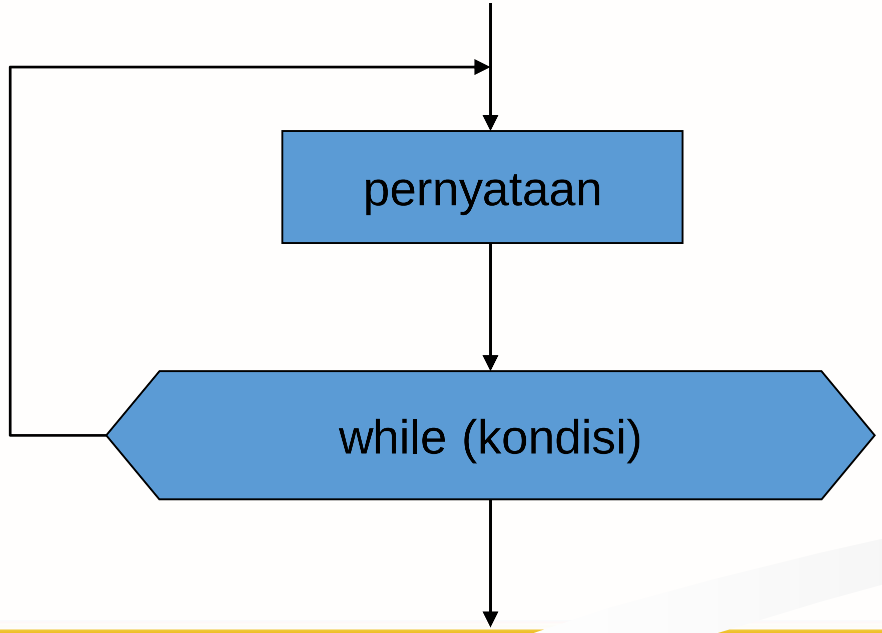


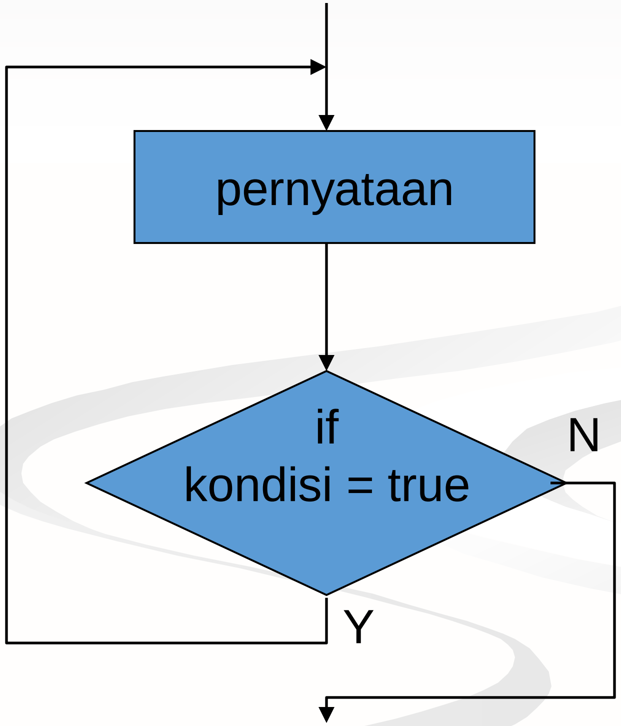
```
int i = 0;
while (i<10) {
        System.out.println(“Hore !!”);
        i++;
}
```

```
int i = 0;
while (i < 10) {
    System.out.println("Are you finished yet?");
    i++;
}
System.out.println("Done");
```

do-while

```
do {
        pernyataan;
} while (ekspresi boolean);
```





```
int i = 0;
do {
        System.out.println(“Hore !!”);
        i++;
} while (i<10);
```

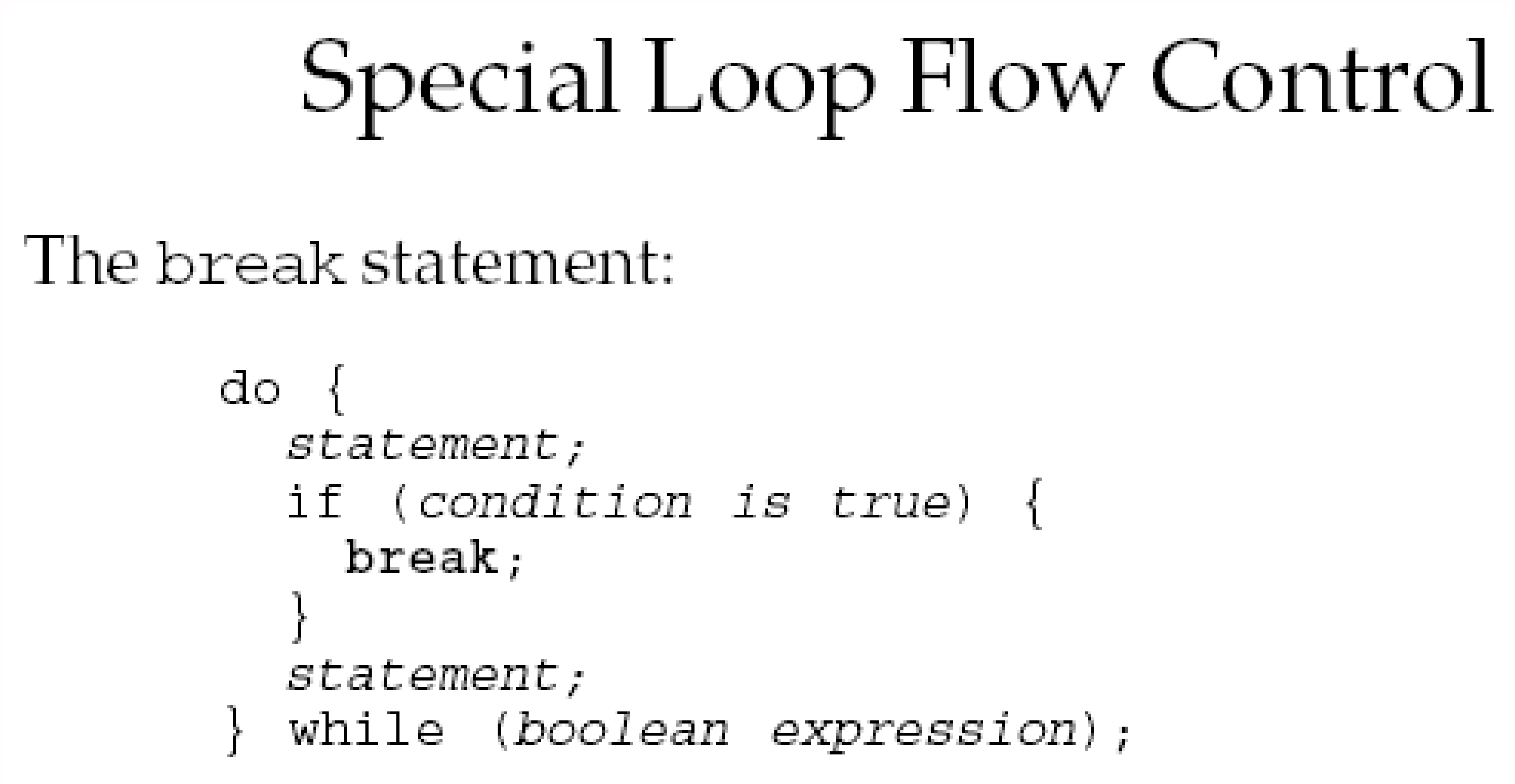
```
int i = 0;
do {
    System.out.println("Are you finished yet?");
    i++;
} while (i < 10);
System.out.println("Done");
```

Special Loop Control

- break [label];
- continue [label];
- label : statement; (statement ini berupa loop)

Special Loop Control

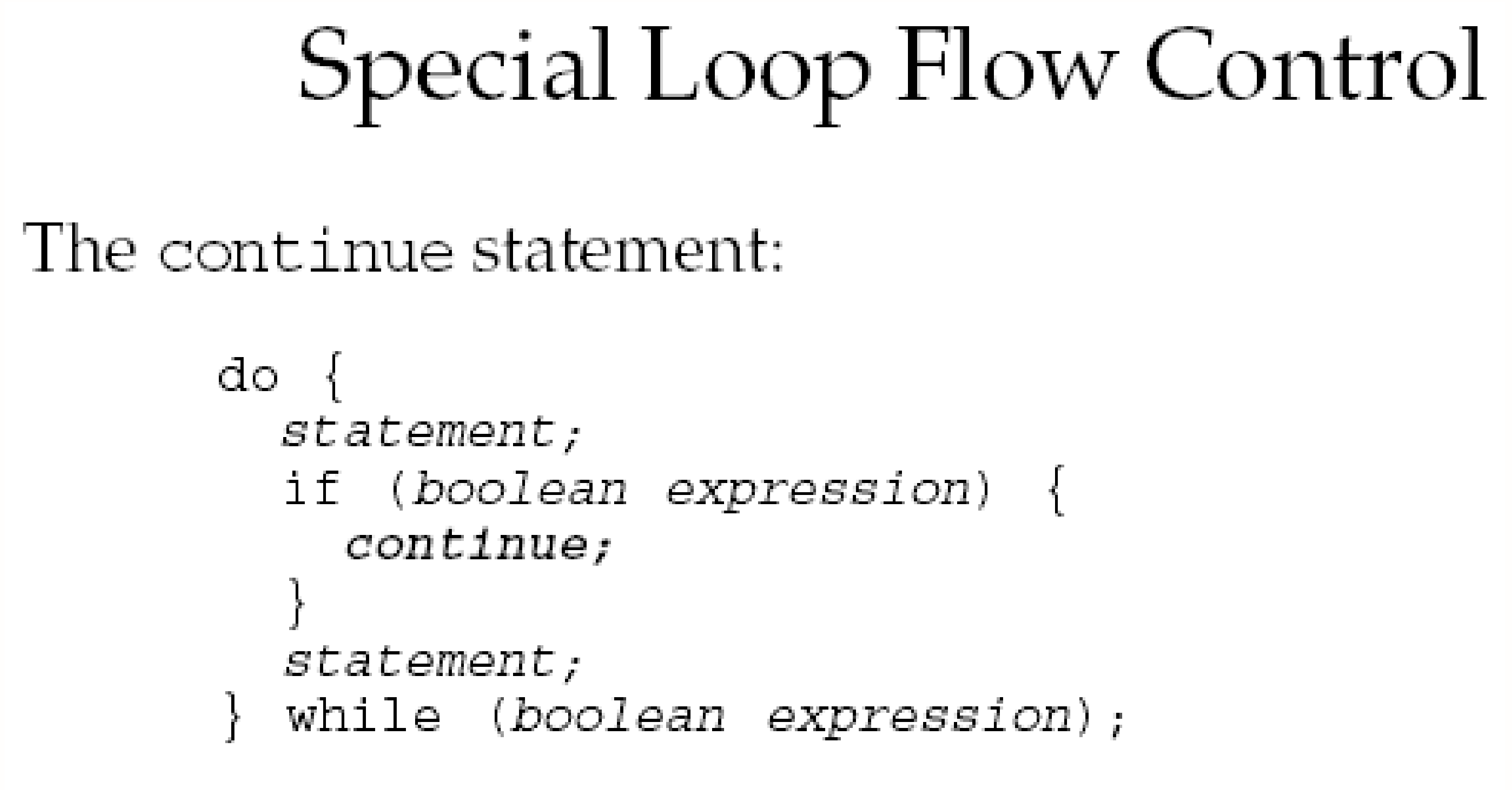
- **`break`** digunakan untuk keluar ("prematurely exit") dari switch statements, loop statements, dan labeled blocks.
- **`continue`** digunakan untuk meneruskan ("skip over and jump) ke akhir dari loop body, dan kembali ke loop control statement.
- **`label`** digunakan untuk mengidentifikasi statement lain dimana statement lain ini meminta supaya block statement pada label ini dikerjakan.



Special Loop Flow Control

The break statement:

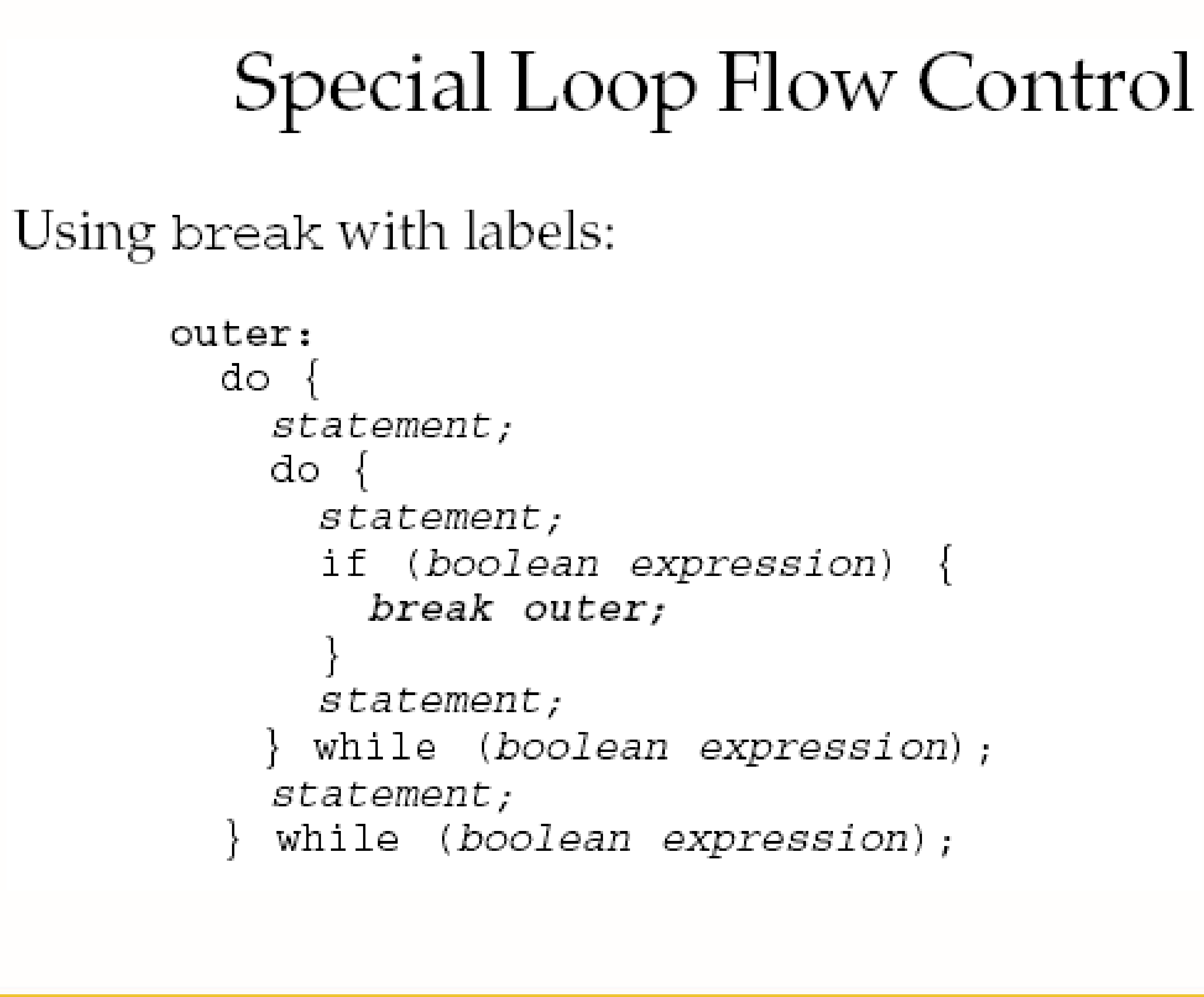
```
do {
  statement;
  if (condition is true) {
    break;
  }
  statement;
} while (boolean expression);
```



Special Loop Flow Control

The continue statement:

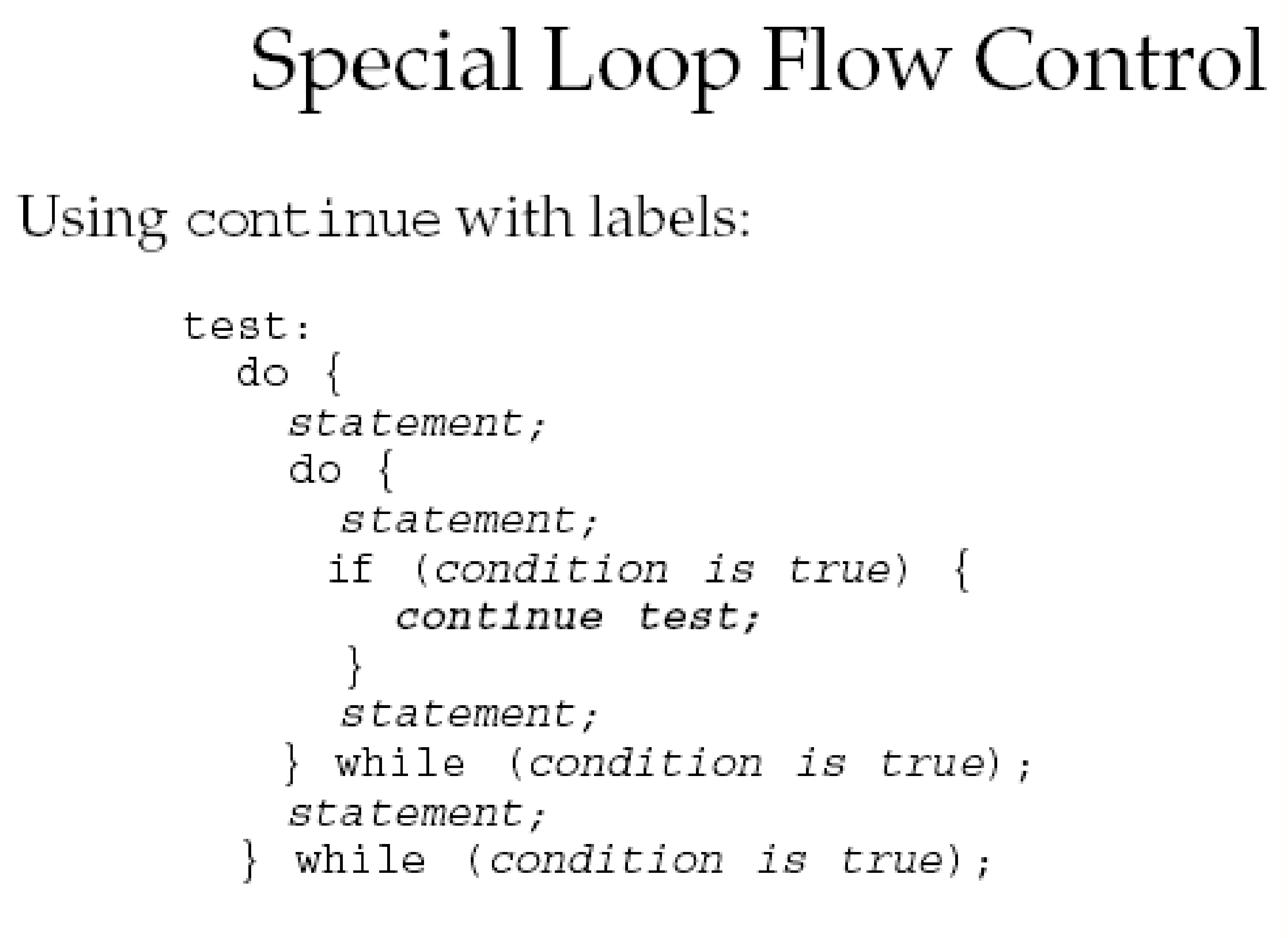
```
do {
  statement;
  if (boolean expression) {
    continue;
  }
  statement;
} while (boolean expression);
```



Special Loop Flow Control

Using break with labels:

```
outer:
  do {
    statement;
    do {
      statement;
      if (boolean expression) {
        break outer;
      }
      statement;
    } while (boolean expression);
    statement;
  } while (boolean expression);
```



Special Loop Flow Control

Using continue with labels:

```
test:
  do {
    statement;
    do {
      statement;
      if (condition is true) {
        continue test;
      }
      statement;
    } while (condition is true);
    statement;
  } while (condition is true);
```

Tugas

1. Sebutkan dan jelaskan berbagai macam sintaks percabangan yang digunakan di Java!
2. Sebutkan dan jelaskan berbagai macam sintaks perulangan yang digunakan di Java!
3. Jelaskan perbedaan if-else dan switch-case!
4. Jelaskan perbedaan perulangan for dan while!
5. Jelaskan perbedaan perulangan while dan do-while!

1. Oracle Java Documentation, The Java™ Tutorials, https://docs.oracle.com/javase/tutorial/, Copyright © 1995, Oracle 2015.
2. Tita Karlita, Yuliana Setrowati, Rizky Yuniar Hakkun, Pemrograman Berorientasi Obyek, PENS-2012
3. Sun Java Programming, Sun Educational Services, Student Guide, Sun Microsystems, 2001.
4. John R. Hubbard, Programming With Java, McGraw-Hill, ISBN: 0-07-142040-1, 2004.
5. Patrick Niemeyer, Jonathan Knudsen, Learning Java, O’reilly, CA, ISBN: 1565927184, 2000.
6. Philip Heller, Simon Roberts, Complete Java 2 Certification Study Guide, Third Edition, Sybex, San Francisco, London, ISBN: 0-7821-4419-5, 2002.
7. Herbert Schildt, The Complete Reference, Java™ Seventh Edition, Mc Graw Hill, Osborne, ISBN: 978-0-07-163177-8, 2007