

Modul 5

Perulangan / Looping

Pengulangan adalah berupa pernyataan dari Java yang mengijinkan kita untuk mengeksekusi blok code berulang-ulang sesuai dengan jumlah tertentu yang diinginkan. Ada tiga macam jenis daristruktur pengulangan yaitu while, do-while, dan for-loops.

While Loop

Pernyataan while loop adalah pernyataan atau blok pernyataan yang diulang-ulang sampai mencapai kondisi yang cocok. Bentuk pernyataan while,

```
while(boolean_expression)
{
    statement1;
    statement2;
}
```

Pernyataan didalam while loop akan dieksekusi berulang-ulang selama kondisi boolean_expression bernilai benar (true). Contoh, pada kode dibawah ini,

```
int i = 4;
while ( i > 0 )
{
    System.out.print(i);
    i--;
}
```

Contoh diatas akan mencetak angka 4321 pada layar.

Do-While Loop

Do-while loop mirip dengan while-loop. Pernyataan di dalam do-while loop akan dieksekusi beberapa kali selama kondisi bernilai benar(true). Perbedaan antara while

dan do-while loop adalah dimana pernyataan di dalam do-while loop akan dieksekusi sedikitnya satu kali. Perbedaan antara while dan do-while loop adalah dimana pernyataan di dalam do-while loop akan dieksekusi sedikitnya satu kali. Bentuk pernyataan do-while,

```
do
{
    statement1;
    statement2;
}
while( boolean_expression );
```

Pernyataan di dalam do-while loop akan dieksekusi pertama kali, dan akan dievaluasi kondisi dari boolean_expression. Jika nilai pada boolean_expression tersebut bernilai true, pernyataan di dalam do while loop akan dieksekusi lagi. Berikut ini beberapa contoh do-while loop:

```
int x = 0;
do
{
    System.out.println(x);
    x++;
}while (x < 10);
```

Contoh ini akan memberikan output 0123456789 pada layar.

For Loop

Pernyataan for loop memiliki kondisi hampir mirip seperti struktur pengulangan sebelumnya yaitu melakukan pengulangan untuk mengeksekusi kode yang sama sebanyak jumlah yang telah ditentukan. Bentuk dari for loop,

```
for (InitializationExpression; LoopCondition; StepExpression){
    statement1;
    statement2;
}
```

- **Initialization Expression** – inialisasi dari variabel loop.
- **Loop Condition** - membandingkan variabel loop pada nilai batas tertentu
- **Step Expression** - melakukan update pada variabel loop.

Berikut ini adalah contoh dari for loop,

```
for( int i = 0; i < 10; i++ )
{
    System.out.print(i);
}
```

Pada contoh ini pernyataan `i=0` merupakan inialisasi dari variabel. Selanjutnya, kondisi `i<10` diperiksa. Jika kondisi bernilai true, pernyataan di dalam for loop dieksekusi. Kemudian, ekspresi `i++` dieksekusi, lalu akan kembali pada bagian pemeriksaan terhadap kondisi `i<10` lagi. Kondisi ini akan dilakukan berulang-ulang sampai mencapai nilai yang salah (false).

Percabangan dalam Perulangan

Pernyataan percabangan memungkinkan kita untuk mengatur aliran eksekusi program. Java memberikan tiga bentuk pernyataan percabangan: `break`, `continue` dan `return`.

- **Pernyataan break**

Pernyataan break tidak berlabel

Pernyataan `break` tidak berlabel (unlabeled) digunakan untuk menghentikan jalannya pernyataan `switch`. Selain itu pernyataan `break` unlabeled juga bias digunakan untuk menghentikan pernyataan-pernyataan `for`, `while` atau `do-while` loop.

```
for( int i=0; i < 5; i++ )
{
    if( i==3)
    {
        System.out.println(i);
        break;
    }
}
```

```
}
```

Dari pernyataan diatas didapatkan jika i berposisi pada nilai maka perulangan akan dihentikan. Dan akan melakukan pencetakan angka i, yaitu angka 3.

Pernyataan break berlabel

Bentuk label dari pernyataan break akan menghentikan pernyataan di luarnya, dimana sebelumnya harus diberikan label yang sudah di spesifikasikan pada program pada pernyataan break. Ketika sebuah nilai ditemukan, break akan menghentikan pernyataan yang diberi label searchLabel yang terletak di luar pernyataan for loop.

```
searchLabel:

for( int i=0; i<6; i++ )
{
    if(i==5 )
    {
        System.out.println(i);
        break searchLabel;
    }
}
```

Pernyataan break menghentikan pernyataan yang diberi label; dan tidak menjalankan aliran kontrol apapun pada label. Aliran control pada label akan diberikan secara otomatis pada pernyataan yang terletak dibawah label. Jadi jika pernyataan yang tidak dibawahnya maka tidak akan mengenali label yang telah dituliskan.

- **Pernyataan Continue**

Pernyataan continue memiliki dua bentuk: berlabel dan tidakberlabel. Andadapat menggunakan pernyataan continue untuk melanjutkan pengulangan yang sedang dijalankan oleh pernyataan for,while, atau do- while loop.

Pernyataan continue tidak berlabel

Bentuk pernyataan continue tidak berlabel (unlabeled) akan melewati bagian pernyataan setelah pernyataan ini dituliskan dan memeriksa ekspresi logika(boolean) yang mengontrol pengulangan. Jika ekspresi logika (boolean) masih bernilai true, maka pengulangan tetap dilanjutkan. Pada dasarnya pernyataan ini akan melanjutkan bagian pengulangan pada pernyataan loop.

```
for( int i=0; i<6; i++ )
{
    if(i==3 )
    {
        continue;
    }
    System.out.println(i);
}
```

Pernyataan continue berlabel

Bentuk pernyataan continue berlabel (labeled) akan melanjutkan pengulangan yang sedang terjadinya dilanjutkan ke pengulangan berikutnya dari pernyataan pengulangan yang diberi label (tanda).

```
search:
for( int i=0; i<6; i++ )
{
    if(i==3 )
    {
        continue search;
    }
    System.out.println(i);
}
```

- **Pernyataan Return**

Pernyataan return digunakan untuk keluar dari sebuah method. Pernyataan return memiliki dua bentuk: memberikan sebuah nilai, dan tidak memberikan nilai. Untuk memberikan sebuah nilai, cukup berikan nilai(atau ekspresi yang menghasilkan sebuah nilai) sesudah kata return. Contohnya, `return ++count;` atau `return "Hello";` Tipe data dari nilai yang diberikan harus sama dengan tipe dari method yang dibuat. Ketika sebuah method void dideklarkasikan, gunakan bentuk return yang tidak memberikan nilai. Contohnya, `return;`

Kita akan membahas lebih lanjut tentang pernyataan return anda akan mempelajari lebih lanjut ketika mempelajari pemrograman berorientasi objek.

Percobaan Dilakukan

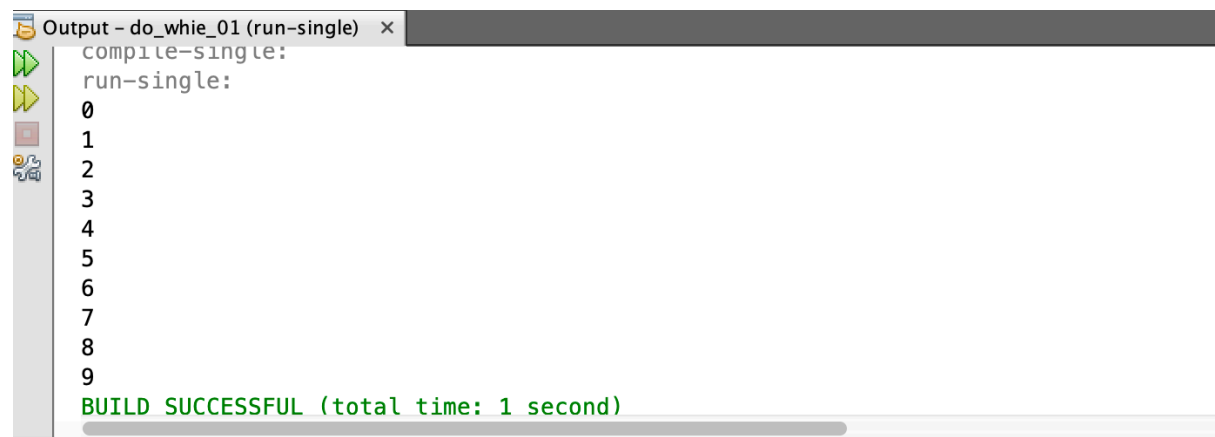
Program Do-While

Buatlah program berikut ini :

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  package do_while_01;
8
9  /**
10   *
11   * @author dani
12   */
13  public class Do_while_01 {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          int x = 0;
21          do
22          {
23              System.out.println(x);
24              x++;
25          }
26          while (x < 10);
27      }
28  }
```

Praktek 1 : Screen capture progam yang telah dibuat seperti tampilan diatas pada e-learning!

Jalankan program yang telah dibuat sebelumnya !

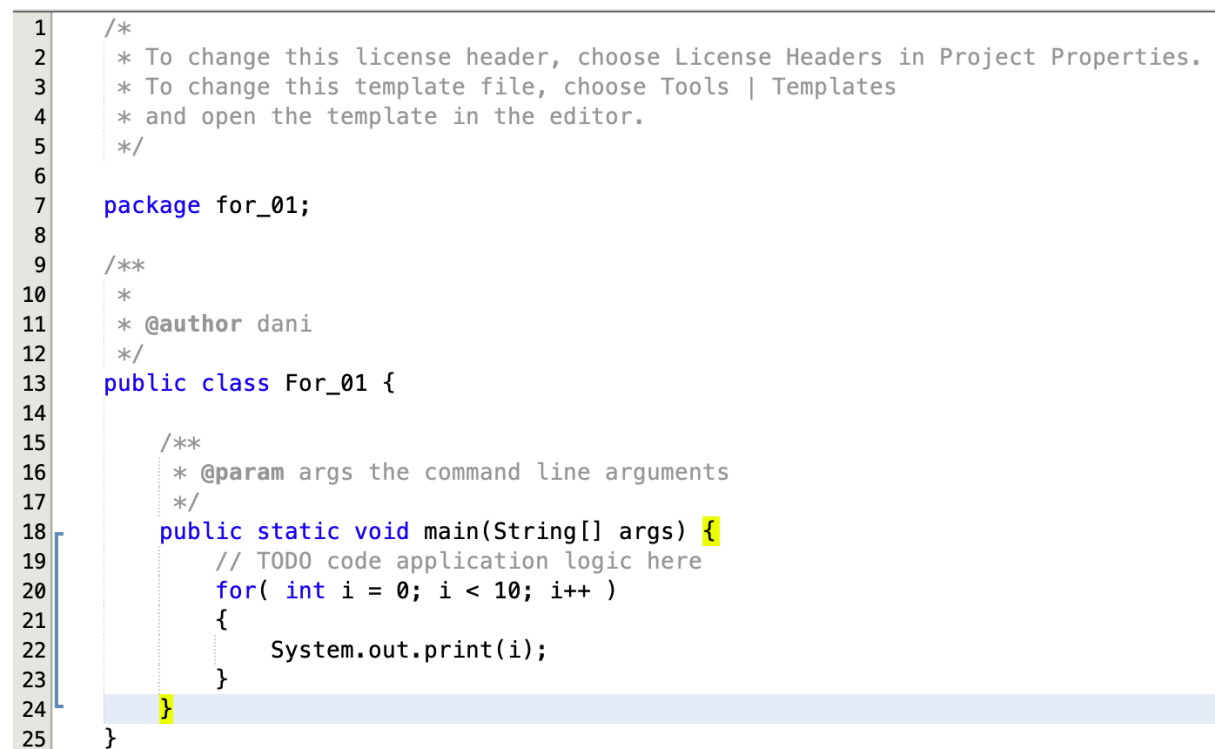


```
Output - do_whie_01 (run-single) x
compile-single:
run-single:
0
1
2
3
4
5
6
7
8
9
BUILD SUCCESSFUL (total time: 1 second)
```

Praktek 2 : Screen capture tampilan output program dibuat seperti tampilan diatas pada e-learning!

For Looping

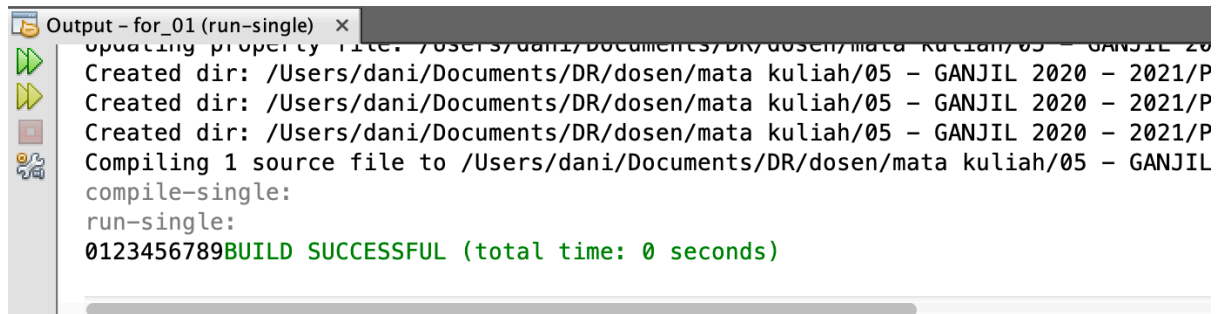
Buatlah program berikut ini :



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7   package for_01;
8
9   /**
10    *
11    * @author dani
12    */
13   public class For_01 {
14
15       /**
16        * @param args the command line arguments
17        */
18       public static void main(String[] args) {
19           // TODO code application logic here
20           for( int i = 0; i < 10; i++ )
21           {
22               System.out.print(i);
23           }
24       }
25   }
```

Praktek 3 : Screen capture program yang telah dibuat seperti tampilan diatas pada e-learning!

Jalankan program yang telah dibuat sebelumnya !

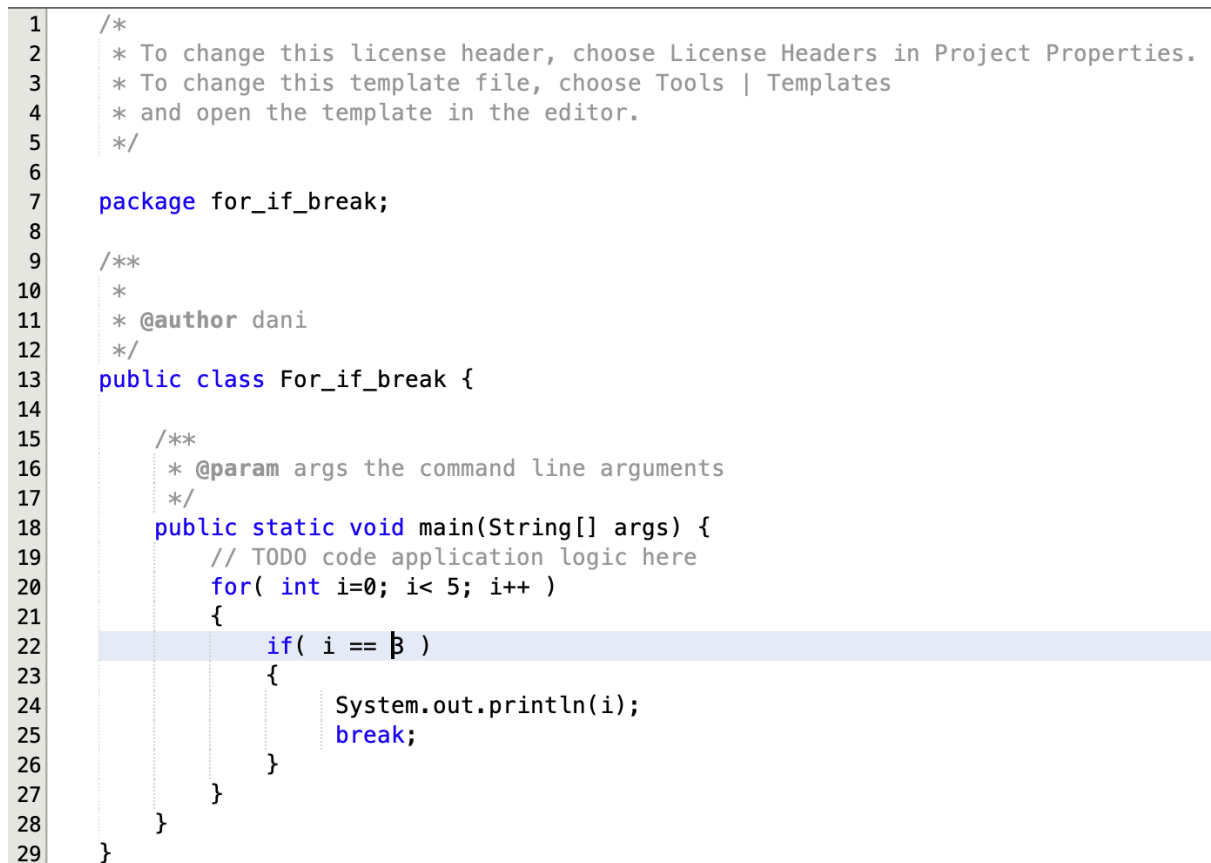


```
Output - for_01 (run-single) x
Updating property file: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 20
Created dir: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 2020 - 2021/P
Created dir: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 2020 - 2021/P
Compiling 1 source file to /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL
compile-single:
run-single:
0123456789BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktek 4 : Screen capture tampilan output program dibuat seperti tampilan diatas pada e-learning!

Percabangan dalam Looping dengan Break

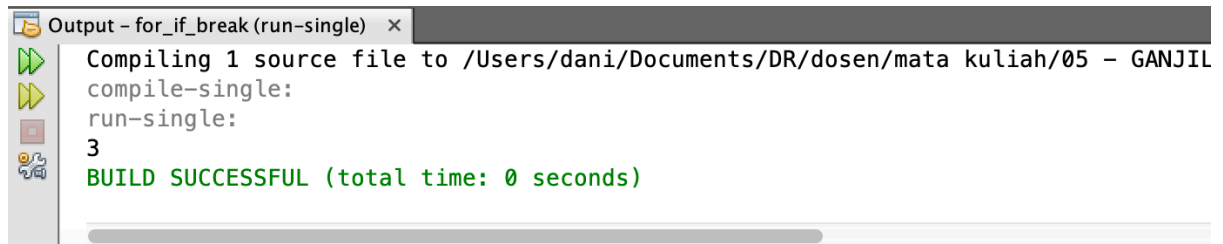
Buatlah program berikut ini :



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  package for_if_break;
8
9  /**
10   *
11   * @author dani
12   */
13  public class For_if_break {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          for( int i=0; i< 5; i++ )
21          {
22              if( i == 3 )
23              {
24                  System.out.println(i);
25                  break;
26              }
27          }
28      }
29  }
```

Praktek 5 : Screen capture program yang telah dibuat seperti tampilan diatas pada e-learning!

Jalankan program yang telah dibuat sebelumnya !

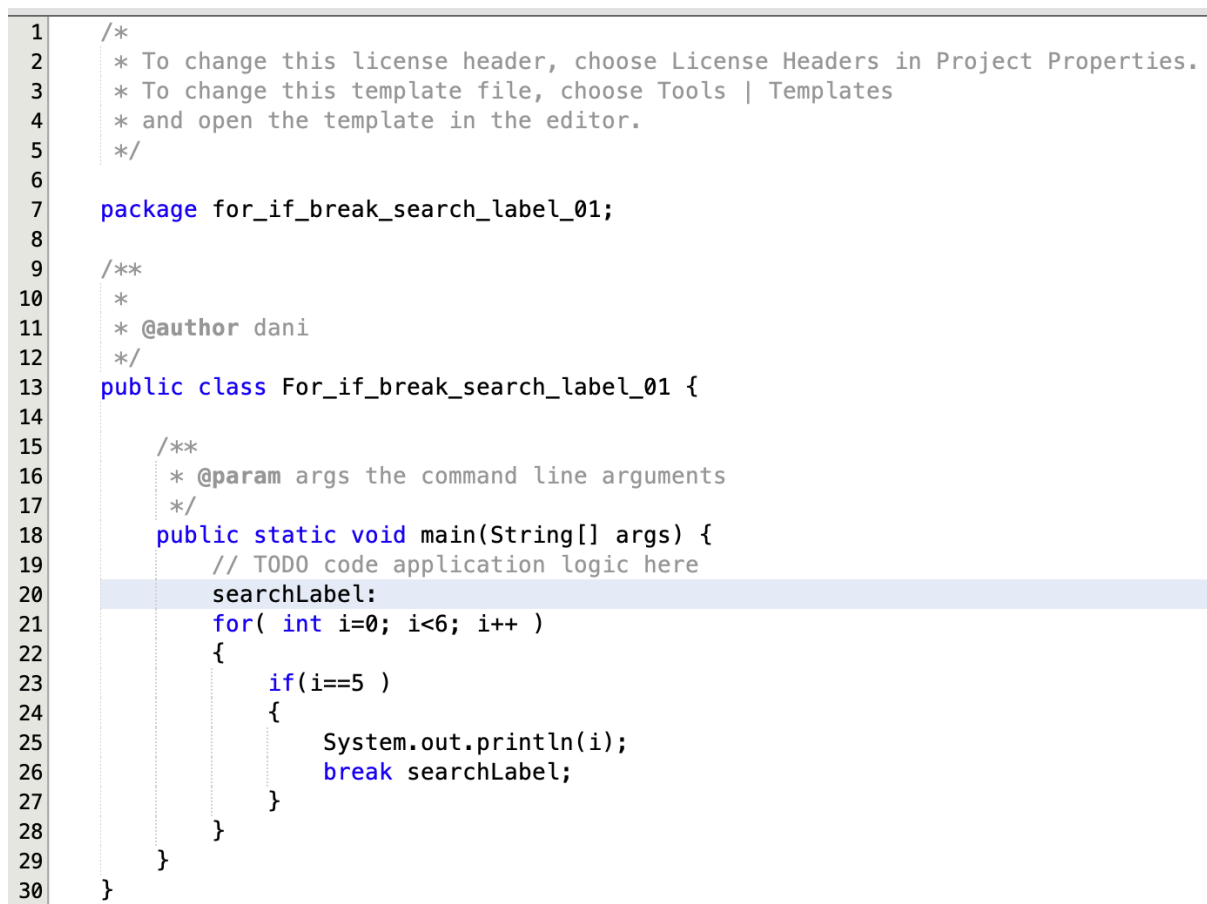


```
Output - for_if_break (run-single) x
Compiling 1 source file to /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL
compile-single:
run-single:
3
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktek 6 : Screen capture tampilan output program dibuat seperti tampilan diatas pada e-learning!

Percabangan dalam Looping dengan Break Label

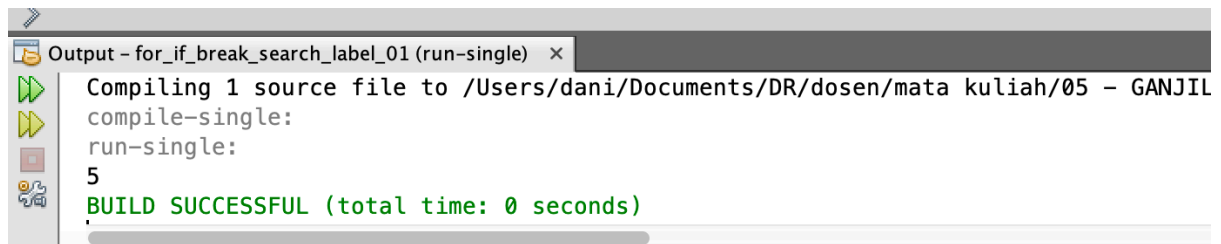
Buatlah program berikut ini :



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  package for_if_break_search_label_01;
8
9  /**
10   *
11   * @author dani
12   */
13  public class For_if_break_search_label_01 {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          searchLabel:
21          for( int i=0; i<6; i++ )
22          {
23              if(i==5 )
24              {
25                  System.out.println(i);
26                  break searchLabel;
27              }
28          }
29      }
30  }
```

Praktek 7 : Screen capture program yang telah dibuat seperti tampilan diatas pada e-learning!

Jalankan program yang telah dibuat sebelumnya !

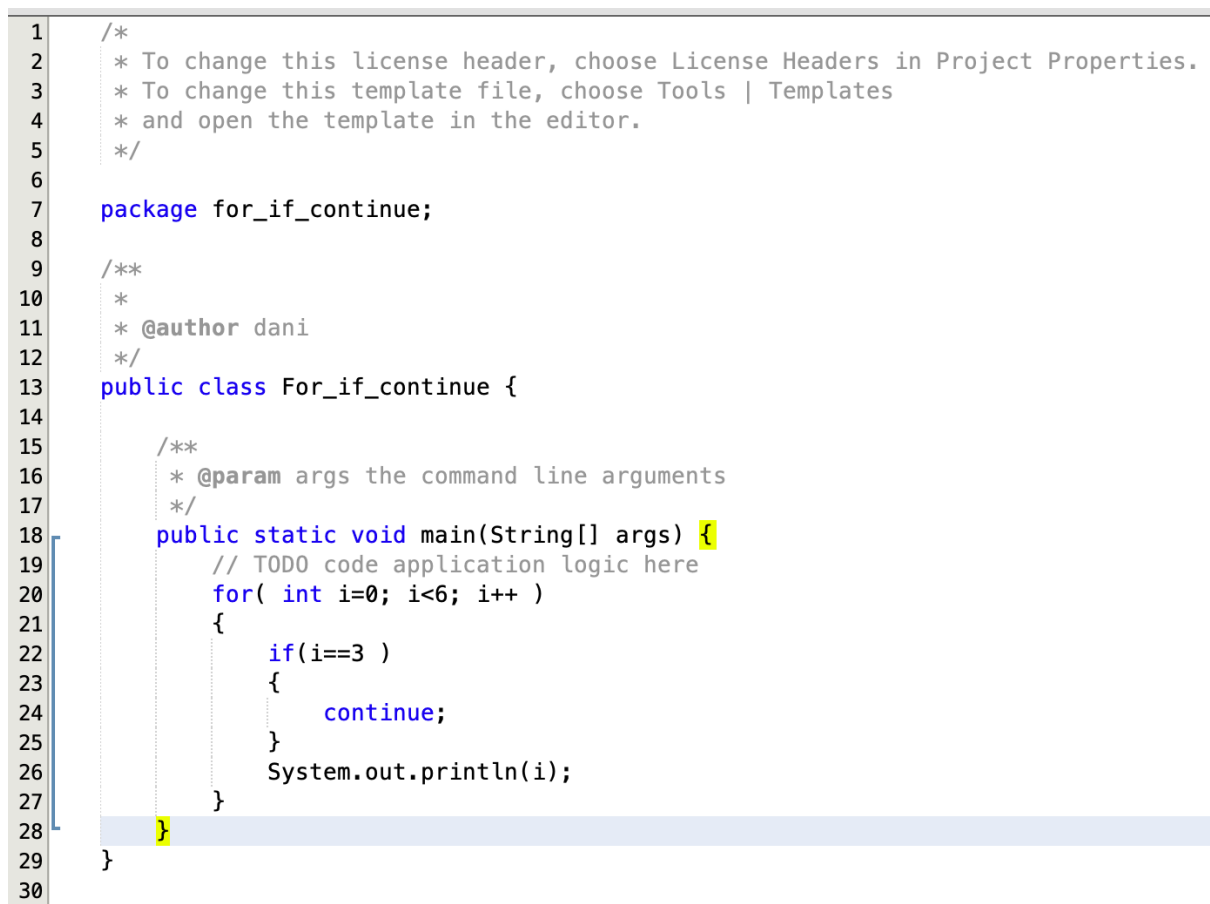
A screenshot of an IDE's output window. The title bar reads "Output - for_if_break_search_label_01 (run-single) x". The output text shows the compilation process: "Compiling 1 source file to /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL", "compile-single:", "run-single:", "5", and finally "BUILD SUCCESSFUL (total time: 0 seconds)".

```
Output - for_if_break_search_label_01 (run-single) x
Compiling 1 source file to /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL
compile-single:
run-single:
5
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktek 8 : Screen capture tampilan output program dibuat seperti tampilan diatas pada e-learning!

Percabangan dalam Looping dengan Continue

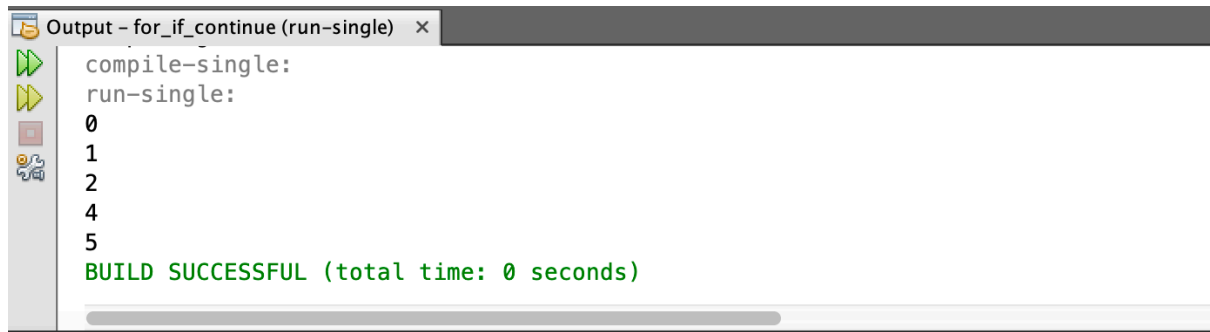
Buatlah program berikut ini :

A screenshot of a code editor showing a Java program. The code is for a package named 'for_if_continue' and a class 'For_if_continue'. The 'main' method contains a 'for' loop from 0 to 5. Inside the loop, there is an 'if' statement that checks if 'i' is equal to 3. If true, it executes 'continue', which skips the rest of the loop body and jumps to the next iteration. Otherwise, it prints the value of 'i' using 'System.out.println(i)'. The code is numbered from 1 to 30 on the left margin.

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  package for_if_continue;
8
9  /**
10   *
11   * @author dani
12   */
13  public class For_if_continue {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          for( int i=0; i<6; i++ )
21          {
22              if(i==3 )
23              {
24                  continue;
25              }
26              System.out.println(i);
27          }
28      }
29  }
30
```

Praktek 9 : Screen capture program yang telah dibuat seperti tampilan diatas pada e-learning!

Jalankan program yang telah dibuat sebelumnya !

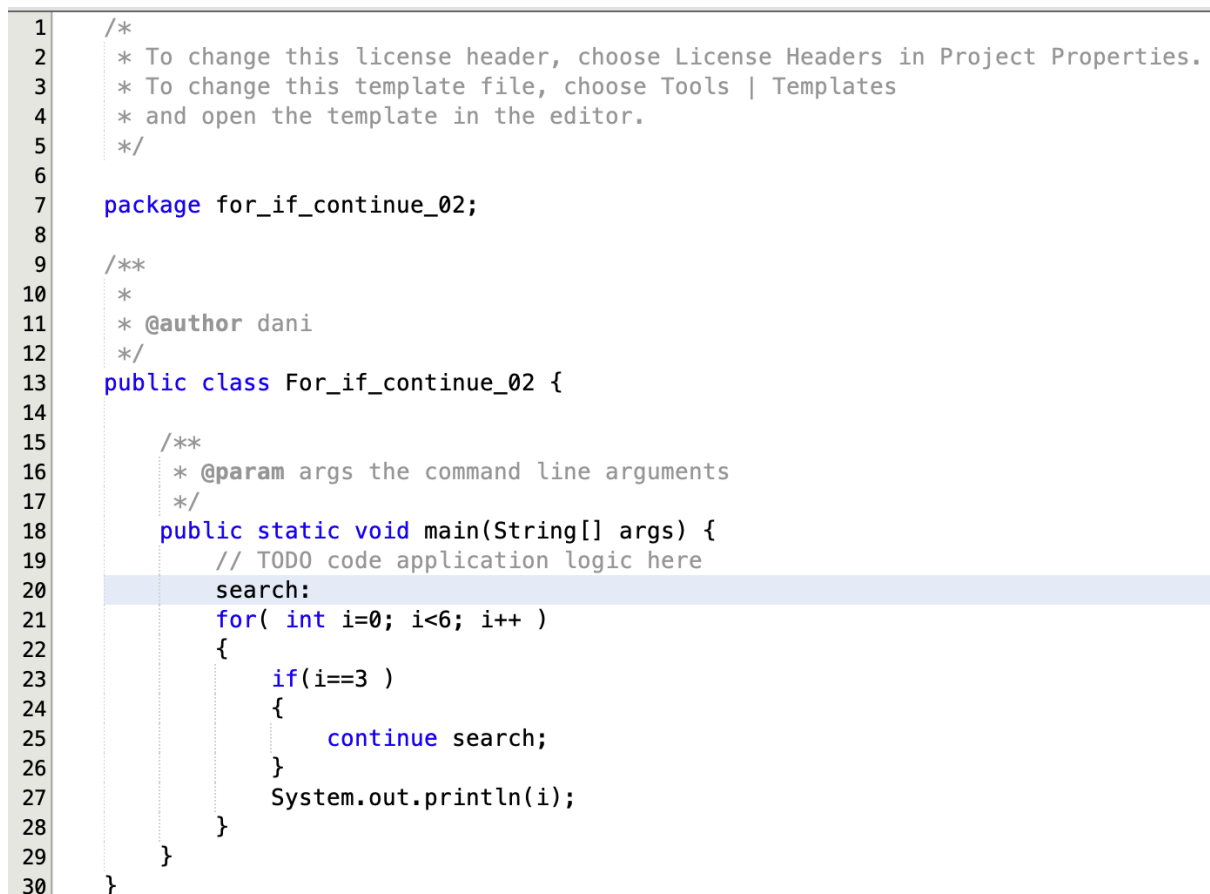


```
Output - for_if_continue (run-single) x
compile-single:
run-single:
0
1
2
4
5
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktek 10 : Screen capture tampilan output program dibuat seperti tampilan diatas pada e-learning!

Percabangan dalam Looping dengan Continue Label

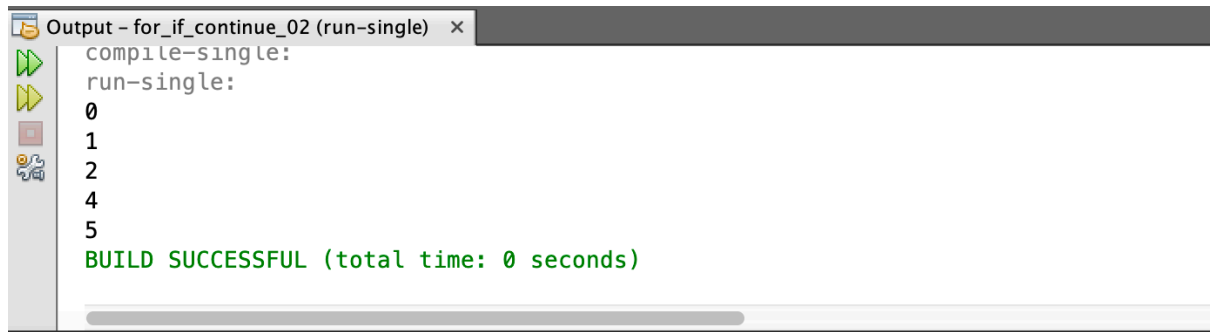
Buatlah program berikut ini :



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  package for_if_continue_02;
8
9  /**
10   *
11   * @author dani
12   */
13  public class For_if_continue_02 {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          search:
21          for( int i=0; i<6; i++ )
22          {
23              if(i==3 )
24              {
25                  continue search;
26              }
27              System.out.println(i);
28          }
29      }
30  }
```

Praktek 11 : Screen capture program yang telah dibuat seperti tampilan diatas pada e-learning!

Jalankan program yang telah dibuat sebelumnya !



The screenshot shows an IDE's output window with a tab titled "Output - for_if_continue_02 (run-single) x". On the left side of the window, there are four icons: a green double arrow, a yellow double arrow, a red square, and a magnifying glass. The output text is as follows:

```
compile-single:  
run-single:  
0  
1  
2  
4  
5  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktek 12 : Screen capture tampilan output progam dibuat seperti tampilan diatas pada e-learning!