

Percabangan

Ada ungkapan “Seribu jalan ke Roma” yang bermakna ada beragam opsi jalan yang bisa dipilih untuk menuju ke Roma. Demikian halnya dengan pemrograman. Terdapat berbagai pilihan dalam pemrograman. Kita memilih salah satu diantaranya. Di tengah-tengah, kita akan menemui problem di mana berjalannya suatu program tergantung dari kondisi yang ditemui. Percabangan bisa kita sebut sebagai *control flow statement*, di mana fungsi utamanya adalah mengontrol alur program.

If-Then

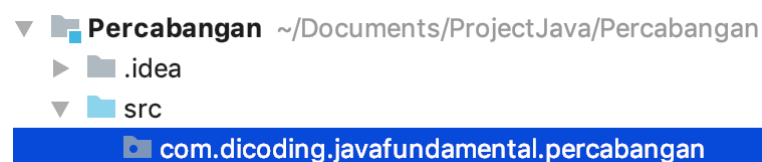
Percabangan if-then adalah *control flow statement* yang paling dasar di dalam pemrograman Java. Ini sangat simpel dengan artian program akan menjalankan baris kode jika kondisinya adalah true.

Codelab If-Then

Mari kita coba percabangan If-Then.

1. Buatlah proyek baru dengan nama Percabangan dengan nama package

`com.dicoding.javafundamental.percabangan` di dalamnya:

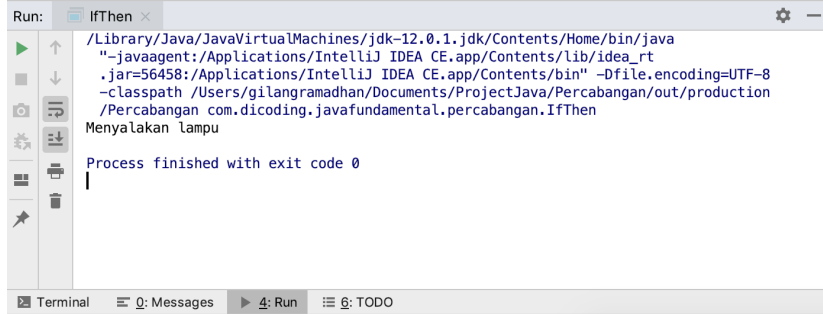


2. Buatlah sebuah kelas baru di dalamnya dengan nama `IfThen`, kemudian tambahkan kode berikut:

```
1. package com.dicoding.javafundamental.percabangan;
2.
3. public class IfThen {
4.     public static void main(String[] args) {
5.         boolean isOn = true;
6.
7.         if (isOn) {
8.             System.out.println("Menyalakan lampu");
9.         }
10.    }
11. }
```

3. Selanjutnya jalankanlah kode di atas pada IDE yang kalian gunakan. Bila sukses, seharusnya Console akan menampilkan output seperti ini.

```
Menyalakan lampu
```



Bedah Code If-Then

Perhatikan kode berikut:

```
1.  if (isOn) {
2.      System.out.println("Menyalakan lampu");
3.  }
```

Kode di atas berarti ketika nilai `isOn` adalah true maka obyek lampu akan memanggil metode `nyala()`. Ini adalah contoh penggunaan percabangan yang sangat simpel, di mana ketika suatu kondisi nilainya true maka kode yang ada di dalam `{ }` (braces) akan bisa dijalankan.

Penggunaan *braces* adalah optional ketika kode yang dijalankan hanyalah 1 baris. Akan tetapi jika terdapat lebih dari 1 baris, maka kode yang di dalamnya harus ada dalam `{ }`. Misalnya, pada contoh kode menyalakan lampu seperti di atas, kita hilangkan tanda `{ }`.

```
1.  void ganti() {
2.      if (isOn)
3.          System.out.println("Menyalakan lampu");
4.  }
```

Dan ketika ada lebih dari 1 baris kode maka kita harus memasukkannya ke dalam `{ }` (braces) seperti ini.

```
1.  void ganti() {
2.      if (isOn) {
3.          System.out.println("Menyalakan lampu");
4.          System.out.println("Menyalakan lampu lain");
5.      }
6.  }
```

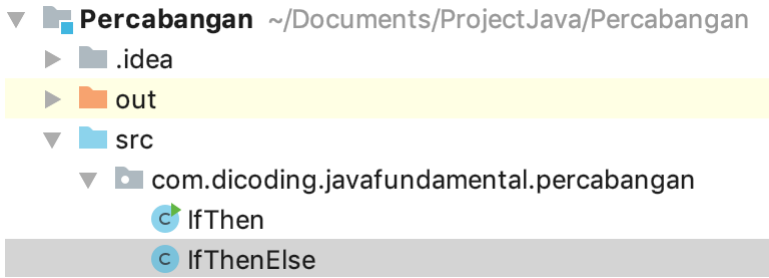
If-Then-Else

Kita juga bisa menambahkan seleksi kondisi baru yaitu else. Ketika semua kondisi di dalam `if` bernilai false maka kode di dalam `else` akan dijalankan.

CodelabIf-Then-Else

Mari kita coba percabangan If-Then-Else.

1. Bukalah kembali proyek Percabangan dan buatlah kelas baru dengan nama `IfThenElse`.



2. Masukkan kode berikut ke dalam kelas `IfThenElse`:

```
1. package com.dicoding.javafundamental.percabangan;
2.
3. public class IfThenElse {
4.     public static void main(String[] args) {
5.         boolean isOn = false;
6.
7.         if (isOn) {
8.             System.out.println("Menyalakan lampu");
9.         } else {
10.            System.out.println("Kondisi tidak terpenuhi...");
11.        }
12.    }
13. }
```

3. Jalankan kode di atas maka hasilnya akan jadi seperti ini:

Kondisi tidak terpenuhi...

Bedah Kode If-Then-Else

Perhatikan kode berikut:

```
1. if (isOn) {
2.     System.out.println("Menyalakan lampu");
3. } else {
4.     System.out.println("Kondisi tidak terpenuhi...");
5. }
```

Kode di atas berarti ketika nilai `isOn` bernilai false maka baris kode yang ada di dalam `else` akan dijalankan. Di sini `else` berfungsi sebagai *secondary option* di mana ketika opsi pertama `if` gagal, maka opsi kedua `else` dijalankan. Sebenarnya `else` bukan hanya sebagai *secondary option* saja. Else lebih cocok dikatakan sebagai *last option* (opsi terakhir).

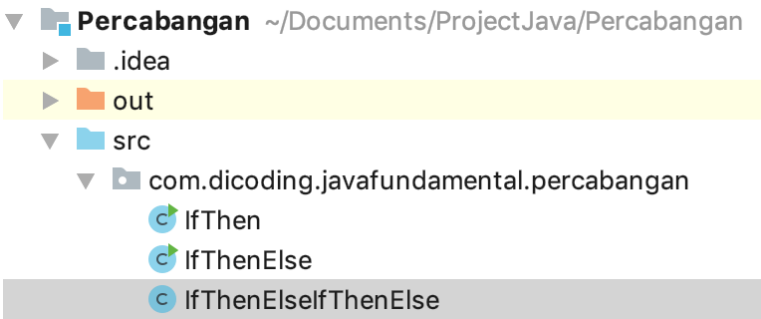
If-Then-Else-Then-Else

Di antara `if` dan `else` , kita bisa menambahkan else if.

Codelab If-Then-Else-Then-Else

Mari kita coba percabangan If-Then-Else-Then-Else.

1. Bukalah kembali proyek Percabangan dan buatlah kelas baru dengan nama `IfThenElseIfThenElse` .

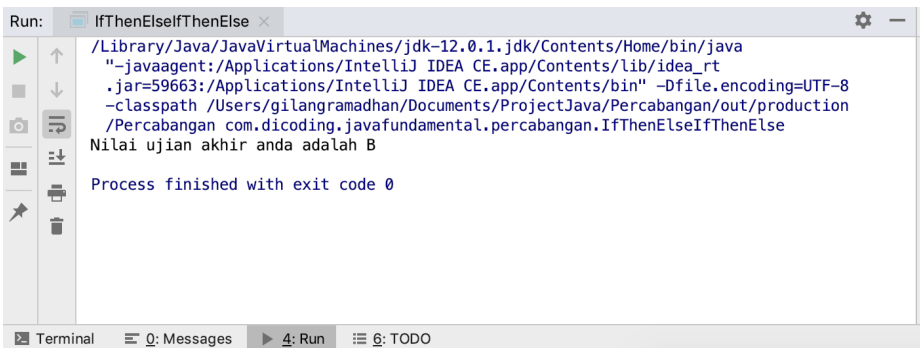


2. Masukkan kode berikut ke dalam kelas `IfThenElse` :

```
1. package com.dicoding.javafundamental.percabangan;
2.
3. public class IfThenElseIfThenElse {
4.     public static void main(String[] args) {
5.         int nilaiUjian = 80;
6.         char indeksPrestasi;
7.
8.         if (nilaiUjian >= 90) {
9.             indeksPrestasi = 'A';
10.        } else if (nilaiUjian >= 80) {
11.            indeksPrestasi = 'B';
12.        } else if (nilaiUjian >= 70) {
13.            indeksPrestasi = 'C';
14.        } else if (nilaiUjian >= 60) {
15.            indeksPrestasi = 'D';
```

3. Jalankan kode di atas maka hasilnya akan jadi seperti ini:

Nilai ujian akhir anda adalah B



Bedah Code If-Then-Else-Then-Else

Perhatikan kode berikut ini:

```
1.  if (nilaiUjian >= 90) {
2.      indeksPrestasi = 'A';
3.  } else if (nilaiUjian >= 80) {
4.      indeksPrestasi = 'B';
5.  } else if (nilaiUjian >= 70) {
6.      indeksPrestasi = 'C';
7.  } else if (nilaiUjian >= 60) {
8.      indeksPrestasi = 'D';
9.  } else if (nilaiUjian >= 50) {
10.     indeksPrestasi = 'E';
11. } else {
12.     indeksPrestasi = 'F';
13. }
```

Kode di atas memiliki beberapa cabang kode yang bisa dijalankan. Tergantung dari nilai variabel integer `nilaiUjian`, kode yang dijalankan akan mengikuti nilainya. Di atas atau sama dengan 90 maka indeks prestasi adalah 'A'; di atas 80 maka 'B'; di atas 70 maka 'C'; di atas 60 maka 'D'; di atas 50 maka 'E'; terakhir di bawah 50 maka indeks prestasinya adalah 'F'.

Bisa disimpulkan bahwa `else` merupakan pilihan terakhir. Ketika `if` dan semua `else-if` nilainya adalah false maka kode di dalam `else` yang akan dijalankan.

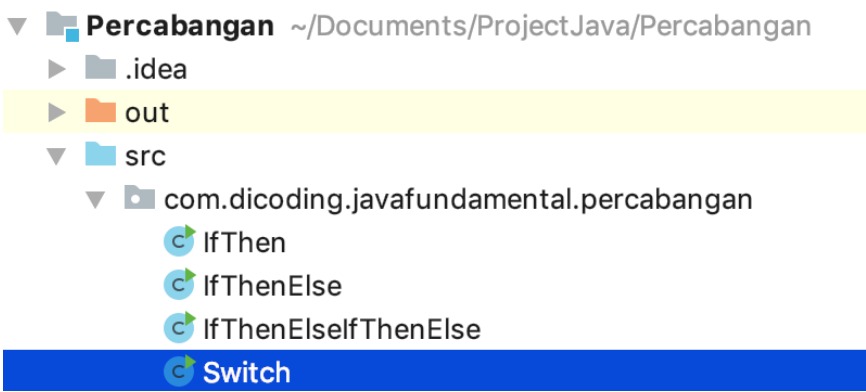
Switch

Control flow lainnya adalah switch, yang menggunakan tipe data primitif seperti int, short, byte, char. Bisa juga dengan menggunakan tipe enumerated, dan string.

Codelab Switch

Mari kita coba percabangan Switch.

1. Bukalah kembali proyek Percabangan dan buatlah kelas baru dengan nama `Switch`.

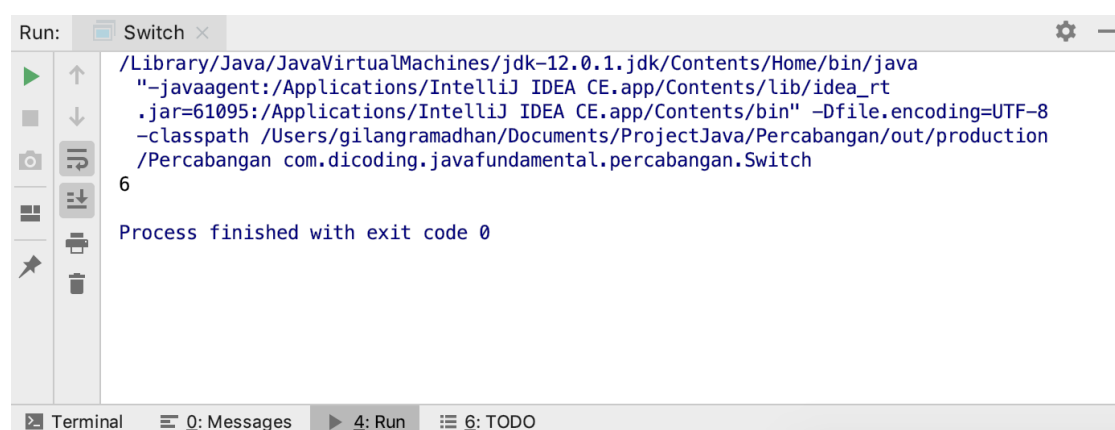


2. Masukkan kode berikut ke dalam kelas `Switch`:

```
1. package com.dicoding.javafundamental.percabangan;
2.
3. public class Switch {
4.     public static void main(String[] args) {
5.         int input = 6;
6.         switch (input) {
7.             case 1:
8.                 System.out.println("1");
9.                 break;
10.            case 2:
11.                System.out.println("2");
12.                break;
13.            case 3:
14.                System.out.println("3");
15.                break;
```

3. Jalankan kode di atas maka hasilnya akan jadi seperti ini:

6



Codelab Switch

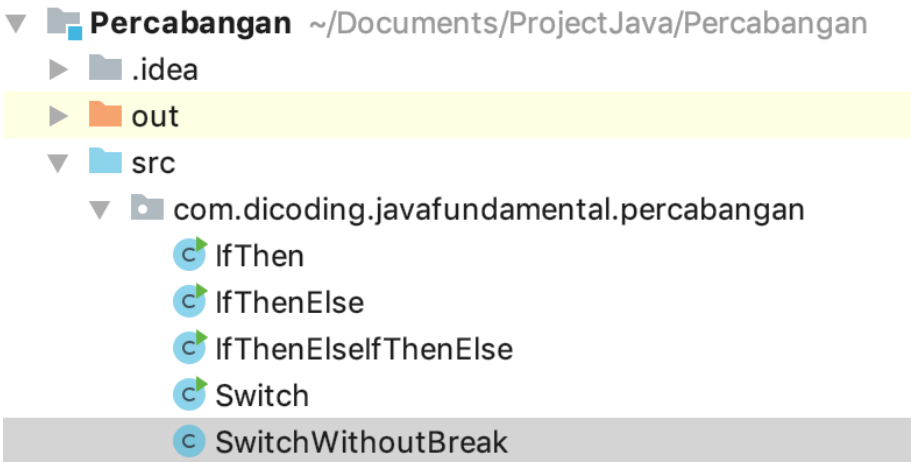
```
1. switch (input) {
2.     case 1:
3.         System.out.println("1");
4.         break;
5.     case 2:
6.         System.out.println("2");
7.         break;
8.     case 3:
9.         System.out.println("3");
10.        break;
11.    case 4:
12.        System.out.println("4");
13.        break;
14.    case 5:
15.        System.out.println("5");
```

Kode di atas berarti variabel *input* memiliki nilai integer 6. Ketika variabel input kita masukkan ke dalam `switch` maka program akan masuk `case` 6. Di dalam `case` 6, program akan menampilkan string “6”, kemudian pemanggilan `break` akan menghentikan proses `switch`.

Tidak berbeda jauh dengan if-then-else bukan? Lalu pertanyaannya, kenapa harus ada break? Bagaimana jika tidak ada break ? Sepertinya kita harus terlebih dahulu menelisik perbedaan dari if-then dan switch.

Codelab Switch Tanpa Break

1. Bukalah kembali proyek Percabangan dan buatlah kelas baru dengan nama `SwitchWithoutBreak`.

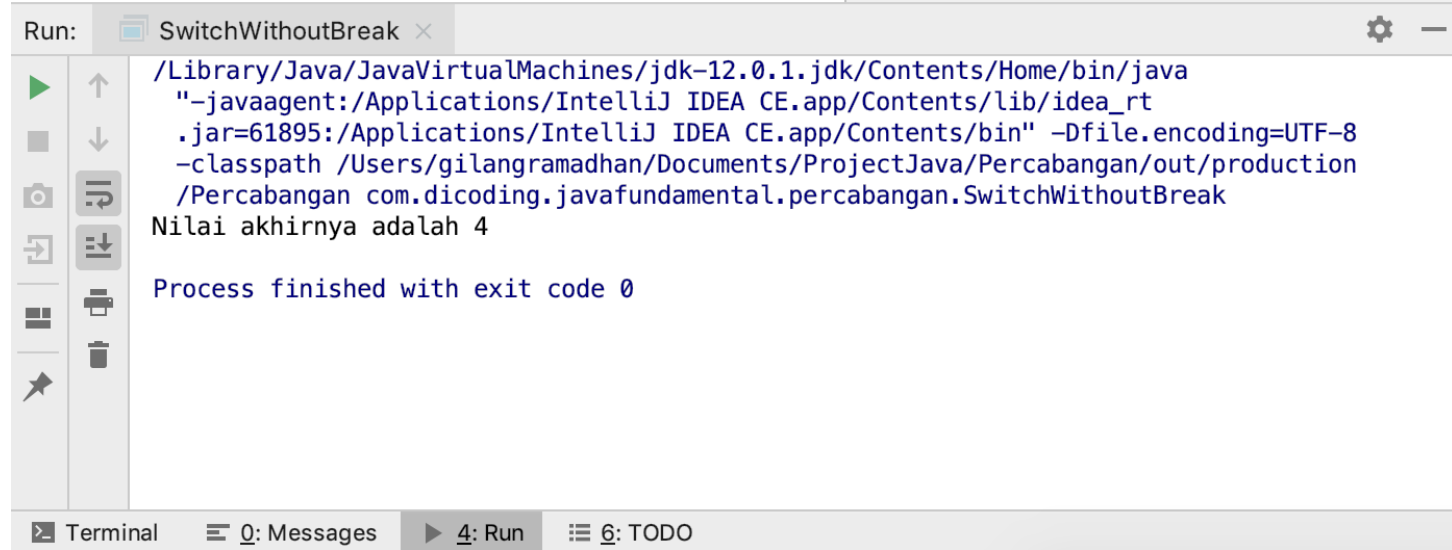


2. Masukkan kode berikut ke dalam kelas `SwitchWithoutBreak`:

```
1. package com.dicoding.javafundamental.percabangan;
2.
3. public class SwitchWithoutBreak {
4.     public static void main(String[] args) {
5.         char input = 'C';
6.         int output = 0;
7.         switch (input) {
8.             case 'A':
9.                 output++;
10.            case 'B':
11.                output++;
12.            case 'C':
13.                output++;
14.            case 'D':
15.                output++;
```

3. Jika kode di atas di jalankan, kira-kira bagaimana ya hasil *output*-nya?
1 atau 4 ?
4. Jika Anda menjawab dengan 4 maka Anda benar. Jikat tidak percaya, jalankanlah kode di atas maka hasilnya akan jadi seperti ini:

Nilai akhirnya adalah 4



Bedah Code Switch Tanpa Break

Bagaimana bisa output yang dihasilkan 4?

Case tanpa **break** akan menjalankan semua kode setelah case yang terpilih. Jadi dengan mengambil contoh kode di atas, dengan nilai inputan adalah char 'C', maka kode program akan masuk ke dalam case 'C'. Di dalamnya nilai variabel output akan ditambah dengan 1 karena pemanggilan operator **++**. Dan karena tidak ada break maka program akan melanjutkan menjalankan case setelahnya yaitu case 'D', case 'E', dan *default*. Dan masing-masing di dalam case ada pemanggilan operator ++, sehingga nilai akhir dari output-nya adalah 4.

Jika masih bingung perhatikan kembali kode di bawah ini yang sudah ditambahkan dengan komen agar lebih jelas.

```
1. package com.dicoding.javafundamental.percabangan;
2.
3. public class SwitchWithoutBreak {
4.     public static void main(String[] args) {
5.         char input = 'C';
6.         int output = 0;
7.         switch (input) {
8.             case 'A':
9.                 output++;
10.            case 'B':
11.                output++;
12.            case 'C':
13.                // Program akan menjalankan kode di dalam case ini
14.                output++; // output = 1
15.        }
```

Setelah mengenal if-then dan switch, lalu statement mana yang kita pilih?

Gunakanlah if-else ketika kondisinya merupakan range dari suatu nilai atau kondisi,. Sebabnya, karena if-else bisa menggunakan operator seperti > atau <. Sedangkan untuk switch, gunakanlah ketika kondisinya hanya satu pilihan seperti integer, char, dan tipe lainnya.

