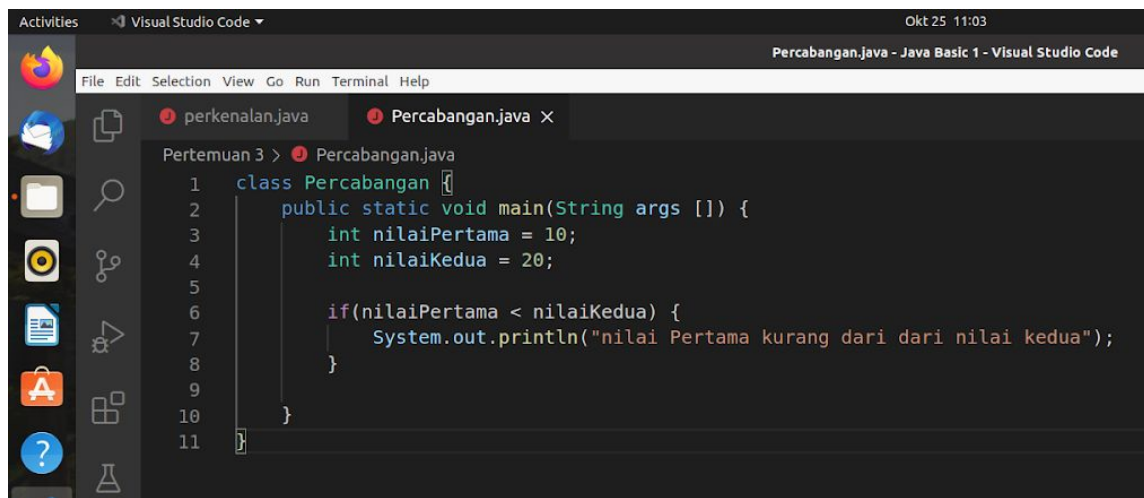


# Modul Pemrograman Berbasis Objek (PBO)

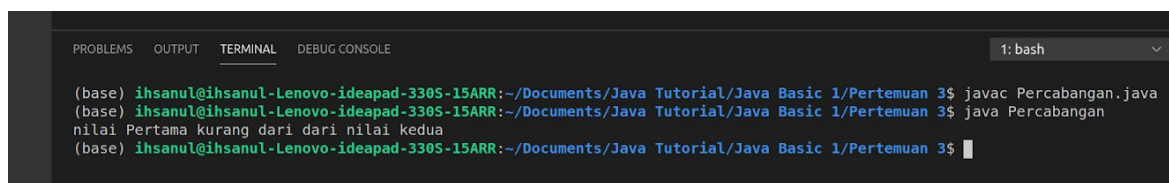
## Java Basic - Percabangan dan Perulangan Part 1

Pada modul kali ini kita akan belajar mengenai percabangan terlebih dahulu baru perulangan. Seperti biasa kita mulai dari kodingan terlebih dahulu

Buka text editor teman teman dan tulis kode dibawah ini dalam suatu file baru bernama percabangan.java lalu jalankan



```
1 class Percabangan {
2     public static void main(String args []) {
3         int nilaiPertama = 10;
4         int nilaiKedua = 20;
5
6         if(nilaiPertama < nilaiKedua) {
7             System.out.println("nilai Pertama kurang dari dari nilai kedua");
8         }
9     }
10 }
11 }
```



```
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$ javac Percabangan.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$ java Percabangan
nilai Pertama kurang dari dari nilai kedua
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$
```

Pada gambar diatas kita telah melakukan percabangan. Percabangan adalah bagaimana kita membuat suatu kondisi dimana jika kita memenuhi kondisi tersebut maka kondisi tersebut akan dijalankan.

Contohnya :

Andi sedang kebingungan ingin mengikuti organisasi web atau data science. Jika andi mengikuti organisasi web, maka andi akan jago web. Sedangkan jika andi mengikuti data science maka dia dapat menganalisa data. Akhirnya andi pun memilih untuk mengikuti organisasai web dan dia pun jago web.

Berdasarkan contoh diatas, kita mengetahui bahwa terdapat 2 cabang dalam pemilihan keputusan andi. Dimana andi harus memutuskan apakah ia akan mengikuti web atau data science dan apa yang akan dia dapatkan jika ia memilih salah satu dari 2 hal tersebut.

Jika kita menuliskan kondisinya maka akan menjadi seperti ini :

Kondisi 1 : Jika ANDI mengikuti organisasi web, maka andi akan jago web

Kondisi 2 : Jika Andi mengikuti organisasi data science, maka andi akan dapat menganalisa data.

Pernyataan : andi mengikuti organisasi web

Maka kondisi yang terpenuhi adalah kondisi 1, dimana andi akan jago web

Mari kita lihat kodingan diatas kembali,

```
if(nilaiPertama < nilaiKedua) {  
    System.out.println("nilai Pertama kurang dari dari nilai kedua");  
}
```

Pada kodingan diatas, kita menuliskan kondisi

Kondisi : nilaiPertama kurang dari nilaiKedua

Jika kondisi terpenuhi, maka akan menampilkan tulisan “nilai pertama kurang dari nilai kedua”

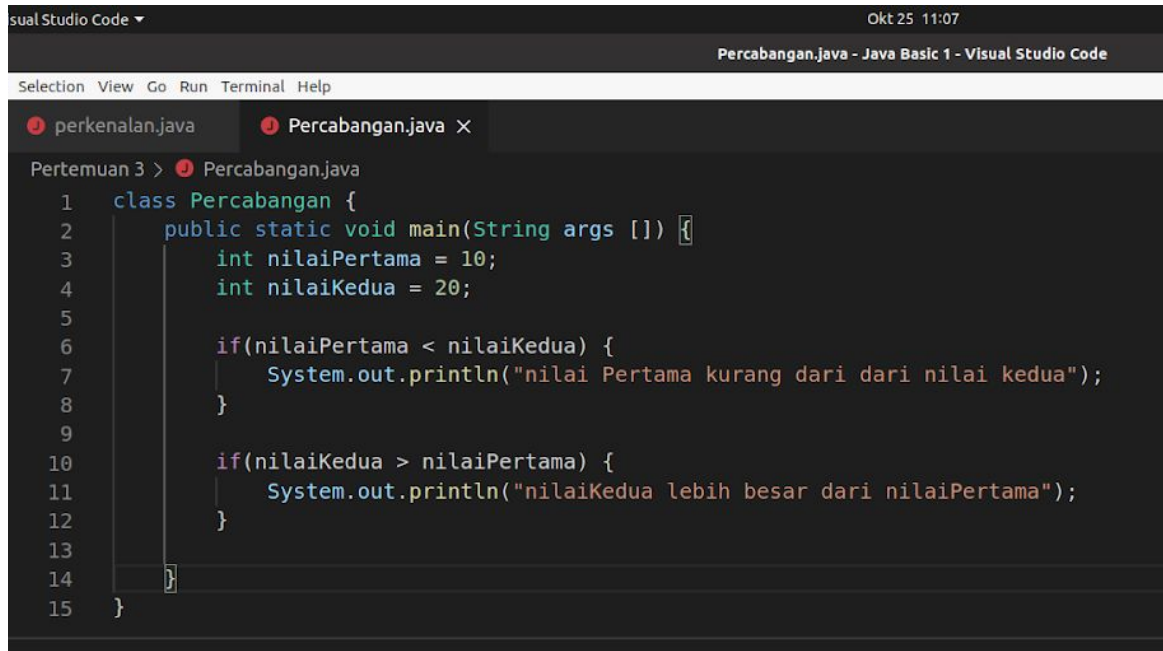
Mari kita lihat kodingan nya secara penuh

```
class Percabangan {  
    public static void main(String args []) {  
        int nilaiPertama = 10;  
        int nilaiKedua = 20;  
  
        if(nilaiPertama < nilaiKedua) {  
            System.out.println("nilai Pertama kurang dari dari nilai kedua");  
        }  
    }  
}
```

Dapat kita lihat bahwa nilaiPertama = 10, nilaiKedua=20, kondisi terpenuhi maka tulisan “nilai pertama kurang dari nilai kedua” akan muncul dilayar terminal/komputer teman teman.

Pada kodingan diatas, kita hanya membuat 1 kondisi. Lalu, bagaimana jika kita menginginkan beberapa kondisi ?

Mari, kita lanjutkan kodingan diatas

A screenshot of the Visual Studio Code editor. The top bar shows 'Visual Studio Code' on the left and 'Okt 25 11:07' on the right. Below the top bar is a menu bar with 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help'. The editor has two tabs: 'perkenalan.java' and 'Percabangan.java'. The 'Percabangan.java' tab is active, showing a Java class named 'Percabangan'. The code is as follows:

```
1 class Percabangan {
2     public static void main(String args []) {
3         int nilaiPertama = 10;
4         int nilaiKedua = 20;
5
6         if(nilaiPertama < nilaiKedua) {
7             System.out.println("nilai Pertama kurang dari dari nilai kedua");
8         }
9
10        if(nilaiKedua > nilaiPertama) {
11            System.out.println("nilaiKedua lebih besar dari nilaiPertama");
12        }
13
14    }
15 }
```

Pada kodingan diatas, kita telah membuat 2 kondisi yaitu

Kondisi 1 : nilaiPertama lebih kecil dari nilaiKedua

Kondisi 2 : nilaiKedua lebih besar dari nilaiPertama

Jika kondisi 1 terpenuhi, maka akan muncul tulisan “nilai pertama kurang dari nilai kedua”

Jika kondisi 2 terpenuhi, maka akan muncul tulisan “nilai kedua lebih besar dari nilai pertama

Lalu, kondisi mana yang akan dijalankan ? Kedua kondisi tersebut akan dijalankan

Kenapa ? Karena 2 kondisi tersebut terpenuhi

nilaiPertama = 10, nilaiKedua=20, maka benar bahwa nilai pertama kurang dari nilai kedua, dan benar juga bahwa nilai kedua lebih besar dari nilai pertama

dan inilah hasil di dalam terminal

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  1: bash
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$ javac Percabangan.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$ java Percabangan
nilai Pertama kurang dari dari nilai kedua
nilaiKedua lebih besar dari nilaiPertama
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$
```

Lalu, bagaimana jika ingin membuat kondisi, dimana jika 1 kondisi terpenuhi maka kondisi lain tidak dijalankan ?

Mari kita berkenalan dengan IF-Else, buat kodingan baru pada file bernama percabangan2.java

```
Okt 25 11:25
Percabangan2.java - Java Basic 1 - Visual Studio Code

perkenalan.java  Percabangan.java  Percabangan2.java x
Percabangan2.java
1  class Percabangan2 {
2      public static void main(String args []) {
3          int nilaiPertama = 10;
4          int nilaiKedua = 20;
5          int nilaiKetiga = 5;
6
7          if(nilaiPertama < nilaiKedua) {
8              System.out.println("nilai Pertama kurang dari dari nilai kedua");
9          }
10         else if(nilaiKedua > nilaiPertama) {
11             System.out.println("nilaiKedua lebih besar dari nilaiPertama");
12         }
13         else if(nilaiKetiga < nilaiPertama) {
14             System.out.println("nilaiKetiga lebih kecil dari nilaiPertama");
15         }
16         else {
17             System.out.println("Nilai setara semua");
18         }
19     }
20 }
21 }

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  2: bash
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1$ javac Percabangan2.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1$ java Percabangan2
nilai Pertama kurang dari dari nilai kedua
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1$
```

Pada kodingan diatas, terdapat 3 kondisi, yaitu  
Kondisi 1 : nilaiPertama lebih kecil dari nilaiKedua  
Kondisi 2 : nilaiKedua lebih besar dari nilaiPertama  
Kondisi 3 : nilaiKetiga lebih kecil dari nilaiPertama

Jika, merujuk pada ke 3 kondisi diatas, maka semua kondisi terpenuhi. Namun, karena kita menggunakan else if pada kodingan diatas, maka hanya 1 kondisi yang dijalankan, yaitu kondisi yang sudah memenuhi diawal, dan kondisi lain nya tidak akan dijalankan, berbeda dengan kodingan diawal yang menggunakan if saja tetapi tidak menggunakan else if.

Untuk else if sendiri, jika terdapat beberapa kondisi, maka jika kondisi pertama tidak terpenuhi maka dia akan ke kondisi ke 2, jika kondisi ke 2 terpenuhi, maka kondisi 2 akan dijalankan dan kondisi ke 3 tidak akan dijalankan. Begitupun, jika kondisi pertama sudah terpenuhi, maka kondisi kedua 2 dan ke 3 tidak akan dijalankan. Lalu, bagaimana jika semua kondisi tidak terpenuhi ? Maka kita bisa bikin default kondisi, yaitu jika semua kondisi tidak memenuhi, maka kita akan menjalankan 1 kondisi default yaitu else

Contoh nya :

Kondisi 1 : If(fikri adalah mahasiswa nf) maka fikri berkuliah dikampus nf

Kondisi 2 : else if(fikri bukan mahasiswa nf) maka fikri tidak berkuliah di nf

Kondisi 3 : else if(fikri bukan mahasiswa) maka fikri tidak berkuliah dimanapun

Kondisi default : else ( ) fikri adalah seorang mahasiswa

Jika kondisi 1 terpenuhi, maka kondisi 2,3, dan default tidak akan dijalankan, tapi jika tidak terpenuhi, maka dialihkan ke kondisi 2. Jika kondisi 2 terpenuhi kondisi 3 dan default tidak akan dijalankan. Tetapi, jika kondisi 2 tidak terpenuhi, maka kondisi 3 akan dijalankan. Lalu jika kondisi 3 tidak terpenuhi, maka kondisi default lah yang akan dijalankan

Mari, kita lihat kodingan kembali

```
6
7     if(nilaiPertama < nilaiKedua) {
8         System.out.println("nilai Pertama kurang dari dari nilai kedua");
9     }
10    else if(nilaiKedua > nilaiPertama) {
11        System.out.println("nilaiKedua lebih besar dari nilaiPertama");
12    }
13    else if(nilaiKetiga < nilaiPertama) {
14        System.out.println("nilaiKetiga lebih kecil dari nilaiPertama");
15    }
16    else {
17        System.out.println("Nilai setara semua");
18    }
19 }
```

Kondisi 1 : nilaiPertama lebih kecil dari nilaiKedua

Kondisi 2 : nilaiKedua lebih besar dari nilaiPertama

Kondisi 3 : nilaiKetiga lebih kecil dari nilaiPertama

Kondisi default : Nilai setara semua

Seperti penjelasan diatas, jika kondisi 1 terpenuhi, maka kondisi 2,3, dan default tidak akan dijalankan. Jika kondisi 1 tidak terpenuhi, maka kondisi dialihkan ke kondisi 2. Jika kondisi 2 terpenuhi, maka kondisi 3 dan default tidak akan dijalankan, begitupun seterusnya. Jika kondisi 1, 2, dan 3 tidak terpenuhi maka kondisi default lah yang akan dijalankan.

Selain IF-else, terdapat cara lain untuk melakukan percabangan. Yaitu menggunakan switch.

Seperti biasa, mari kita buat kodingan baru bernamaPercabanganKetiga.java, lalu jalankan kodingan tersebut

```
Visual Studio Code
PercabanganKetiga.java

Selection View Go Run Terminal Help

PercabanganKetiga.java
PercabanganKetiga.java Percabangan2.java Percab...

Pertemuan 3 > PercabanganKetiga.java
1  import java.util.Scanner;
2
3  class PercabanganKetiga {
4
5      public static void main(String args []) {
6          String text;
7          Scanner input = new Scanner(System.in);
8
9          System.out.print("Masukan nama anda = ");
10         String nama = input.nextLine();
11
12         switch (nama) {
13             case "Nurul":
14                 text = "Halo nama saya nurul";
15                 break;
16             case "Fikri":
17                 text = "Halo nama saya Fikri";
18                 break;
19             case "Andi":
20                 text = "Halo nama saya Andi";
21                 break;
22             default:
23                 text = "Halo, saya tidak punya nama";
24                 break;
25         }
26
27         System.out.println(text);
28     }
29 }
```

```
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$ javac PercabanganKetiga.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$ java PercabanganKetiga
Masukan nama anda = Nurul
Halo nama saya nurul
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$ java PercabanganKetiga
Masukan nama anda = Fikri
Halo nama saya Fikri
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$ java PercabanganKetiga
Masukan nama anda = Andi
Halo nama saya Andi
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$ java PercabanganKetiga
Masukan nama anda = randi
Halo, saya tidak punya nama
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 3$
```

Pada kodingan diatas, kita membuat suatu kondisi menebak nama. Dimana jika nama yang ditebak nurul maka text nya akan berisi “halo nama saya nurul”, begitupun jika nama yang ditebak fikri, maka text yang akan muncul adalah “halo nama saya fikri” dan begitupun untuk andi juga seperti itu. Jika nama yang ditebak tidak ada dalam pilihan maka akan muncul “halo,saya tidak punya nama”

Notes :

- Fungsi break pada kodingan diatas adalah, jika kondisi terpenuhi maka tidak akan dicari kondisi lain.
- fungsi import digunakan untuk menggunakan fungsi yang disediakan oleh java/orang lain, yang tidak tersedia secara default jika tidak mengimport fungsi tersebut. Dalam contoh kasus diatas, yang di import adalah fungsi `java.util.Scanner` yang nantinya digunakan untuk menerima inputan langsung dari user bukan ditulis manual pada kodingan
- `new Scanner` digunakan untuk menginisialisasi variable yang akan digunakan sebagai fungsi penerima inputan. Pada contoh diatas variable yang digunakan bernama `input`
- `input.nextLine`, `input` adalah variabel yang digunakan untuk menyimpan `Scanner`. Otomatis variabel `input` mempunyai kemampuan menggunakan fungsi dari `Scanner` yaitu menerima inputan. `.nextLine` sendiri, kita gunakan untuk memberi tahu inputan yang akan di input user adalah `String`. Hasil inputan yang di input user disimpan dalam variabel `nama`.