

## TUGAS SESI 7 PEMROGRAMAN BERORIENTASI OBJEK

Nama : Dave Maulana Ferros

NIM : 20210040114





Matkul : Pemrograman Berorientasi Objek

Kelas : TI21E

```
4 |  * and open the template in the editor.
5 |  */
6 |  package pbo_sesi7;
7 |
8 |  /**
9 |   *
10 |   * @author Dave Maulana Ferros
11 |   */
12 |  public class Outer {
13 |      int number = 0;
14 |
15 |      private class Inner{
16 |          public void print(){
17 |              System.out.println("Mengakses inner class yang ke: " + (++number));
18 |          }
19 |      }
20 |
21 |      void displayFromMethod(){
22 |          Inner in = new Inner();
23 |          in.print();
24 |      }
25 |  }
26 |
27 |  class Main{
28 |      public static void main(String[] args) {
29 |          Outer out = new Outer();
30 |          out.displayFromMethod();
31 |          out.displayFromMethod();
32 |      }
33 |  }
34 |
```

- a. Dari kode diatas, diketahui bahwa class **Outer** merupakan *outer class*, dan class **Inner** merupakan bagian dari *inner class*, class Inner juga menerapkan hak akses private, selanjutnya method **displayfromMethod()** merupakan pembuatan instance dari inner class, method ini juga dipanggil dari method utama atau **Main**.

b. Class yang dihasilkan setelah mengeksekusi file

 Main.class	11/25/2022 7:58 PM	CLASS File	1 KB
 Outer\$1.class	11/25/2022 7:58 PM	CLASS File	1 KB
 Outer\$Inner.class	11/25/2022 7:58 PM	CLASS File	2 KB
 Outer.class	11/25/2022 7:58 PM	CLASS File	1 KB

## 2. a.

- Jika kode diganti menjadi `number++` maka yang akan terjadi adalah




```
run:
Mengakses inner class yang ke: 0
Mengakses inner class yang ke: 1
Mengakses inner class yang ke: 2
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Jika kode diganti menjadi `++number` maka hasilnya adalah

```
run:
Mengakses inner class yang ke: 1
Mengakses inner class yang ke: 2
Mengakses inner class yang ke: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

Posisi simbol `++` menentukan kapan nilai variabel akan ditambah. Jika simbol `++` ditulis di depan, maka akan ditambah terlebih dahulu. Sedangkan bila ditulis di belakang maka akan ditambah setelahnya.

## b. Class yang dihasilkan setelah mengeksekusi file

 Outer\$1Inner.class	11/25/2022 8:29 PM	CLASS File	1 KB
 Outer.class	11/25/2022 8:29 PM	CLASS File	1 KB
 OuterAccess.class	11/25/2022 8:29 PM	CLASS File	1 KB