

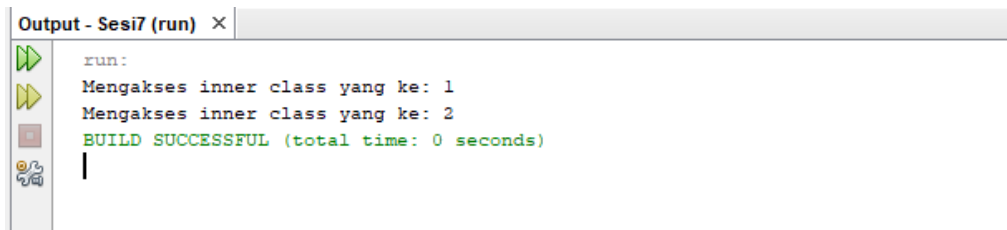
Nama : Yulva Cintakandida  
NIM : 20210040079  
Kelas : TI21A  
Matkul : Pemrograman Berorientasi Objek

## Sesi 7

1.

```
10 |  * @author HP 14S
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 |  }
```

- a. Pada kode program diatas terdapat “Outer” sebagai outer class atau kelas luar yang mempunyai “Inner” yang merupakan inner class. Class “Outer” mempunyai method displayFromMethod yang nantinya akan membuat objek dari inner classnya kemudian memanggil method print dari objek si kelas tersebut. Pada class Main kemudian dibuat objek kelas outer Bernama “out”, lalu method displayFromMethod dari objek “out”. Untuk output yang dihasilkan seperti berikut :



```
Output - Sesi7 (run) x
run:
Mengakses inner class yang ke: 1
Mengakses inner class yang ke: 2
BUILD SUCCESSFUL (total time: 0 seconds)
```

- b. Menghasilkan 2 file dengan ekstensi .class yaitu : Main.class dan Outer.class

## 2. Program yang menggunakan variabel (++number)

```
10  |  * @author HP 14S
11  |  */
12  |  public class Outer {
13  |      int number = 0;
14  |
15  |      void outerMethod() {
16  |          class Inner {
17  |              public void print() {
18  |                  System.out.println("Mengakses inner class yang ke: " + (++number));
19  |              }
20  |          }
21  |
22  |          Inner inner = new Inner();
23  |          inner.print();
24  |      }
25  |  }
26  |
27  |  class OuterAccess {
28  |      public static void main(String[] args) {
29  |          Outer out = new Outer();
30  |          out.outerMethod();
31  |          out.outerMethod();
32  |          out.outerMethod();
33  |      }
34  |  }
```

Output - Sesi7 (run) X

```
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)
```

## Program yang menggunakan variabel (number++)

```
10  |  * @author HP 14S
11  |  */
12  |  public class Outer {
13  |      int number = 0;
14  |
15  |      void outerMethod() {
16  |          class Inner {
17  |              public void print() {
18  |                  System.out.println("Mengakses inner class yang ke: " + (number++));
19  |              }
20  |          }
21  |
22  |          Inner inner = new Inner();
23  |          inner.print();
24  |      }
25  |  }
26  |
27  |  class OuterAccess {
28  |      public static void main(String[] args) {
29  |          Outer out = new Outer();
30  |          out.outerMethod();
31  |          out.outerMethod();
32  |          out.outerMethod();
33  |      }
34  |  }
```

Output - Sesi7 (run) X

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

- Penulisan simbol ++ dibelakang variabel (`++number`) disebut dengan post-increment, sedangkan bila ditulis didepan variabel (`number++`) disebut pre-increment. Posisi simbol ++ ini menentukan kapan nilai variabel akan ditambah. Bila ditulis didepan, maka akan ditambah terlebih dahulu, sedangkan bila ditulis dibelakang akan ditambah belakangan.
- Ketika sudah dirun akan menghasilkan 3 file dengan ekstensi .class yaitu :Outer.class, Outer\$1Inner.class, dan OuterAccess.class.