

TUGAS SESI 7

PEMROGRAMAN BERORIENTASI OBJEK

Nama : Tiara Lelita

Kelas : TI21E

NIM : 20210040091

1. Kode Program yang sudah saya coba

```
1
2 public class Outer {
3     int number=0;
4
5     private class Inner{
6         public void print() {
7             System.out.println("Mengakses Inner class yang ke: " + (++number));
8         }
9     }
10
11     void displayfromMethod() {
12         Inner in = new Inner();
13         in.print();
14     }
15 }
16
17 class Main{
18
19     public static void main(String[] args){
20         Outer out = new Outer();
21         out.displayfromMethod();
22         out.displayfromMethod();
23     }
24 }
```

Outer > Inner > print >

Notifications Output x

User - C:\Users\User x Sesi7 (run) x

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

- Pada kode diatas yang sudah saya coba terdapat class outer yaitu public class Outer, dimana disana juga terdapat class inner yang hak aksesnya private, disana juga ada displayFromMethod() adalah metode dimana terdapat pembuatan instance inner class, dan metode ini dipanggil dari metode utama.
 - Hasil eksekusi dari kode program diatas menghasilkan dua class yaitu Mengakses inner class yang ke : 1 dan Mengakses inner class yang ke : 2. Untuk lebih jelasnya bisa dilihat pada gambar diatas
- #### 2. Kode Program yang sudah saya coba

```
1
2 public class Outer {
3     int number=0;
4
5     private class Inner{
6         public void print(){
7             System.out.println("Mengakses Inner class yang ke: " + (number++));
8         }
9     }
10
11     void displayfromMethod(){
12         Inner in = new Inner();
13         in.print();
14     }
15
16 }
17
18 class Main{
19
20     public static void main(String[] args){
21         Outer out = new Outer();
22         out.displayfromMethod();
23         out.displayfromMethod();
24     }
25 }
```

Outer > Inner > print >

Notifications Output x

User - C:\Users\User x Sesi7 (run) x

run:
Mengakses Inner class yang ke: 0
Mengakses Inner class yang ke: 1
BUILD SUCCESSFUL (total time: 0 seconds)

- a. Dari gambar diatas terlihat hasil eksekusi yang berbeda dari hasil yang pertama, mungkin ini terjadi karena kode ++number diganti dengan number++. Nah ternyata setelah saya menelusuri berbagai sumber ++number (prefix) dan number++ (postfix)
Jadi posisi simbol ++ menentukan kapan nilai variabel akan ditambah.
Bila simbol ++ ditulis didepan maka akan ditambah dulu, sedangkan bila ditulis dibelakang akan di tambah belakangan
- b. Sehingga eksekusi dari kode program diatas menghasilkan dua class yaitu Mengakses inner class yang ke : 0 dan Mengakses inner class yang ke : 1. Untuk lebih jelasnya bisa dilihat pada gambar diatas