

LAPORAN HASIL PRAKTIKUM

Teknik Pemrograman

Laporan ini disusun untuk memenuhi Tugas Mata Kuliah Praktikum Teknik Pemrograman



disusun oleh:

Berliana Elfada (211524004)

PROGRAM STUDI D4 TEKNIK INFORMATIKA

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

POLITEKNIK NEGERI BANDUNG

Kasus 1

```
10  L  */
11      public class barang {
12
13          String kode_barang;
14          String nama_barang;
15          private int stok;
16
17      public String getNama_barang() {
18          return nama_barang;
19      }
20      public void setNama_barang(String nama_barang) {
21          this.nama_barang = nama_barang;
22      }
23      public int getStok() {
24          return stok;
25      }
26      public void setStok(int stok) {
27          this.stok = stok;
28      }
29      public void tambahStok(int tambah) {
30          stok += tambah; //method aritmatika untuk menambah stok
31      }
32      public barang(String kode, String nama, int stk) {
33          kode_barang = kode;
34          nama_barang = nama;
35          stok = stk;
36      }
37  }
```

barang.java

```
9      * @author Berliana Elfada
10     */
11     public class inventori {
12         barang[] barangs;
13         void initbarang() {
14             barangs = new barang[2];
15             barangs[0] = new barang ("001", "Baju", 10);
16             barangs[1] = new barang ("002", "Celana", 20);
17         }
18         void showbarang() {
19             System.out.println(barangs[0].getNama_barang() + "(" + barangs[0].getstok() + ")");
20             System.out.println(barangs[1].getNama_barang() + "(" + barangs[1].getstok() + ")");
21         }
22         void pengadaan() {
23             initbarang();
24             barangs[0].tambahStok(10);
25             barangs[1].tambahStok(20);
26             showbarang();
27         }
28         public static void main(String[] args) {
29             inventori beli = new inventori();
30             beli.pengadaan();
31         }
32     }
33 }
```

inventori.java

output :

```
run:
Baju(20)
Celana(40)
BUILD SUCCESSFUL (total time: 0 seconds)
```

pada kasus pertama ini, program berjalan untuk menampilkan nama, dan jumlah barang dengan menggunakan konsep enkapsulasi pada variable yang disediakan.

Kasus 2

```
9      * @modified Berliana Elfada
10     */
11     public class Item {
12         private String name;
13         private Item() {
14             name = "Ipin";
15         }
16
17         public String getName() {
18             return name;
19         }
20         public void setName(String name) {
21             this.name = name;
22         }
23         public Item(String name) {
24             System.out.println(this.name = new Item().name);
25         }
26     }
27
28
```

Item.java

```
5     package week4;
6
7     /**
8      *
9      * @modified Berliana Elfada
10     */
11     public class UpinIpin {
12         public static void main(String[] args) {
13             Item name = new Item("upin");
14         }
15     }
```

UpinIpin.java

Output :

```
run:
Ipin
BUILD SUCCESSFUL (total time: 0 seconds)
```

pada kasus kedua ini, program menjalankan code agar output menghasilkan "Ipin" dengan menggunakan constructor dan dengan memberikan suatu nilai pada variable tersebut yang sifatnya private.

Kasus 3

```
8      *
9      * @modified Berliana Elfada
10     */
11     public class KelasSatu {
12     {
13         System.out.println(11);
14     }
15     static{
16         System.out.println(2);
17     }
18     public KelasSatu(int i){
19         System.out.println(3);
20     }
21     public KelasSatu(){
22         System.out.println(4);
23     }
24 }
25
```

KelasSatu.java

```
5     package week4;
6
7     /**
8      *
9      * @modified Berliana Elfada
10    */
11    public class KelasDua {
12    {
13        System.out.println(5);
14    }
15    public static void main(String[] args){
16        System.out.println(6);
17        KelasSatu Satu = new KelasSatu();
18        KelasSatu Dua = new KelasSatu(10);
19    }
20 }
21
```

KelasDua

Output :

```
run:
6
2
11
4
11
3
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pada program kasus ketiga ini, hasil output tampil dengan urutannya masing-masing, yaitu 6 di baris pertama karena ada pada main method. Kemudian angka 2 karena berada pada static block yang akan dieksekusikan pertama setelah main method dan hanya dilakukan sekali. Lalu, angka 11 di sana instance dari KelasSatu dan calling objek satu. Kemudian angka 4, merupakan calling method KelasSatu. Angka 11 diprint kembali karena instance dari KelasSatu dan calling objek dua. Dan yang terakhir 3, calling method KelasSatu(10).