

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

MODUL 10

INTERFACE



DISUSUN OLEH :

NAMA : BIMA TRIADMAJA

NIM : L200210137

KELAS : E

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

TAHUN 2022/2023

10.2.1. Percobaan

Buat interface dengan nama ActivityLampu seperti pada program 4 berikut ini!

```
1 public interface ActivityLampu {
2     public static final int LAMPU_HIDUP=1;
3     public static final int LAMPU_MATI=0;
4     public abstract void matikanLampu();
5     public abstract void hidupkanLampu(); 6
6 }
```

Program 4. Interface ActivityLampu dengan method dan variabelnya

Selanjutnya buat class Lampu yang merupakan implementasi dari interface IntLampu seperti Program 5 berikut ini!

```
1 public class Lampu implements ActivityLampu{
2     public int statusLampu;
3
4     @Override
5     public void matikanLampu() {
6         if(statusLampu==0){
7             System.out.println("Lampu sudah dalam
8 kondisi mati");
9         }else if(statusLampu==1){
10            statusLampu=-1;
11            System.out.println("Lampu sudah
12 dimatikan");
13        }
14    }
15
16    @Override
17    public void hidupkanLampu() {
18        if(statusLampu==1){
19            System.out.println("Lampu sudah
20 dinyalakan\n***");
21        }else {
22            statusLampu=+1;
23            System.out.println("Lamu sudah dalam
24 kondisi menyala");
25        }
26    }
27
28    public int setSaklar(int saklar){
29        return statusLampu = saklar;
30    }
31 }
```

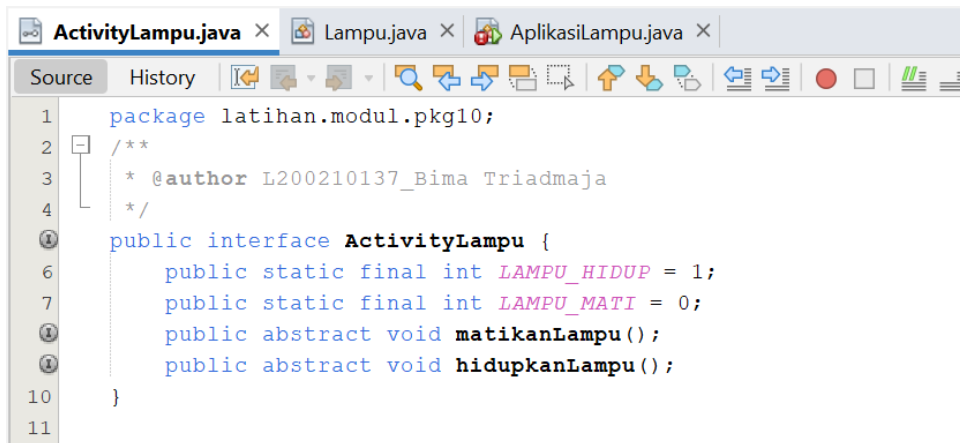
Program 5. Class Lampu mengimplementasikan interface

```
1 public class AplikasiLampu {
2     public static void main(String[] args) {
3         Lampu lampu = new Lampu();
4         Scanner sc = new Scanner(System.in);
5         lampu.statusLampu = lampu.setSaklar(0);
6         System.out.println("Status Lampu =
7 "+lampu.statusLampu+"\nketikkan");
8         System.out.println("1 untuk menyalakan
9 lampu\n0 untuk mematikan lampu");
10
11         if(lampu.setSaklar(sc.nextInt())==0){
12             lampu.matikanLampu();
13         }else {
14             lampu.hidupkanLampu();
15         }
16     }
17 }
```

Program 6. Fungsi main () untuk menjalankan program interface

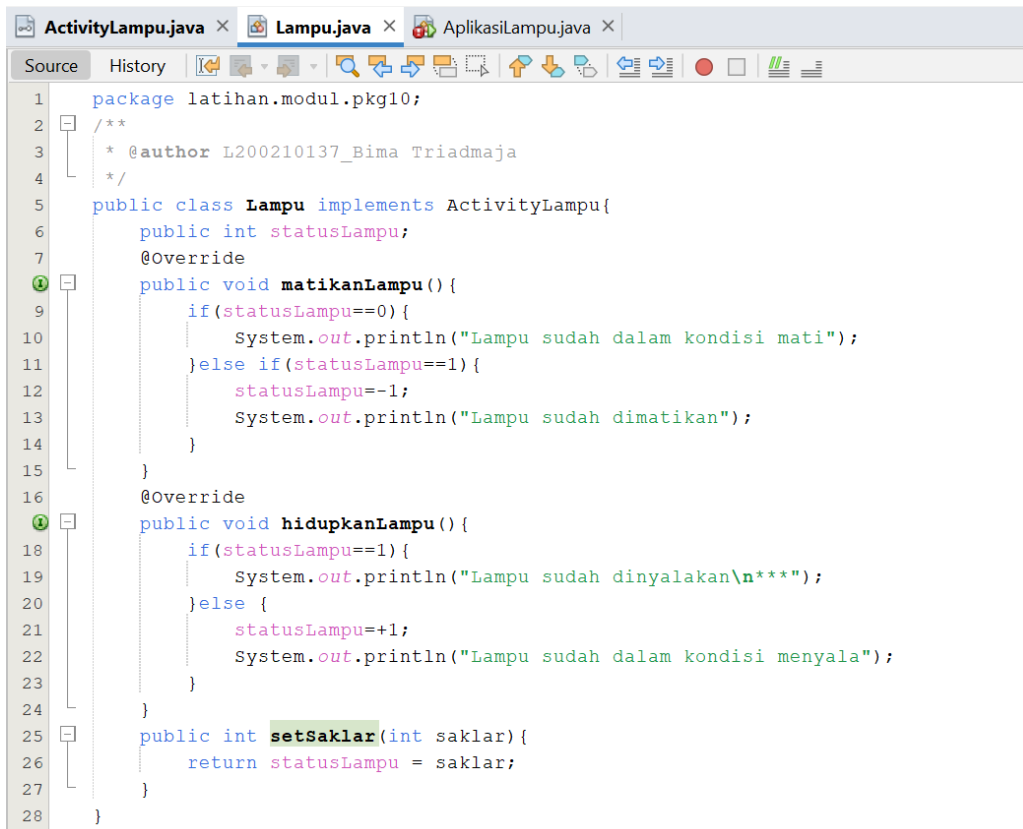
❖ Hasil Kegiatan :

Class ActivityLampu.java (interface) :



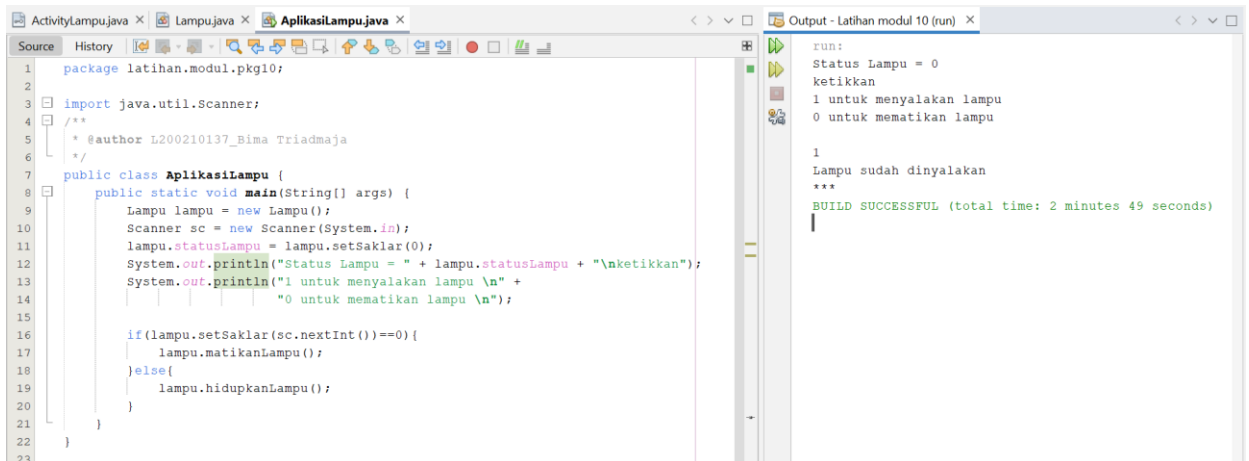
```
1 package latihan.modul.pkg10;
2 /**
3  * @author L200210137_Bima Triadmaja
4  */
5 public interface ActivityLampu {
6     public static final int LAMPU_HIDUP = 1;
7     public static final int LAMPU_MATI = 0;
8     public abstract void matikanLampu();
9     public abstract void hidupkanLampu();
10 }
11
```

Class Lampu.java (implement dari ActivityLampu) :



```
1 package latihan.modul.pkg10;
2 /**
3  * @author L200210137_Bima Triadmaja
4  */
5 public class Lampu implements ActivityLampu{
6     public int statusLampu;
7     @Override
8     public void matikanLampu(){
9         if(statusLampu==0){
10             System.out.println("Lampu sudah dalam kondisi mati");
11         }else if(statusLampu==1){
12             statusLampu=-1;
13             System.out.println("Lampu sudah dimatikan");
14         }
15     }
16     @Override
17     public void hidupkanLampu(){
18         if(statusLampu==1){
19             System.out.println("Lampu sudah dinyalakan\n***");
20         }else {
21             statusLampu+=1;
22             System.out.println("Lampu sudah dalam kondisi menyala");
23         }
24     }
25     public int setSaklar(int saklar){
26         return statusLampu = saklar;
27     }
28 }
```

Class AplikasiLampu.java (aplikasi dari Lampu) dan Output :



The screenshot shows an IDE with two tabs: 'AplikasiLampu.java' and 'Output - Latihan modul 10 (run)'. The source code in the left pane is as follows:

```
1 package latihan.modul.pkg10;
2
3 import java.util.Scanner;
4
5 /**
6  * @author L200210137_Bima Triadmaja
7  */
8 public class AplikasiLampu {
9     public static void main(String[] args) {
10         Lampu lampu = new Lampu();
11         Scanner sc = new Scanner(System.in);
12         lampu.setStatusLampu = lampu.setSaklar(0);
13         System.out.println("Status Lampu = " + lampu.statusLampu + "\nketikkan");
14         System.out.println("1 untuk menyalakan lampu \n" +
15             "0 untuk mematikan lampu \n");
16
17         if (lampu.setSaklar(sc.nextInt()) == 0) {
18             lampu.matikanLampu();
19         } else {
20             lampu.hidupkanLampu();
21         }
22     }
23 }
```

The output in the right pane shows the execution results:

```
run:
Status Lampu = 0
ketikkan
1 untuk menyalakan lampu
0 untuk mematikan lampu

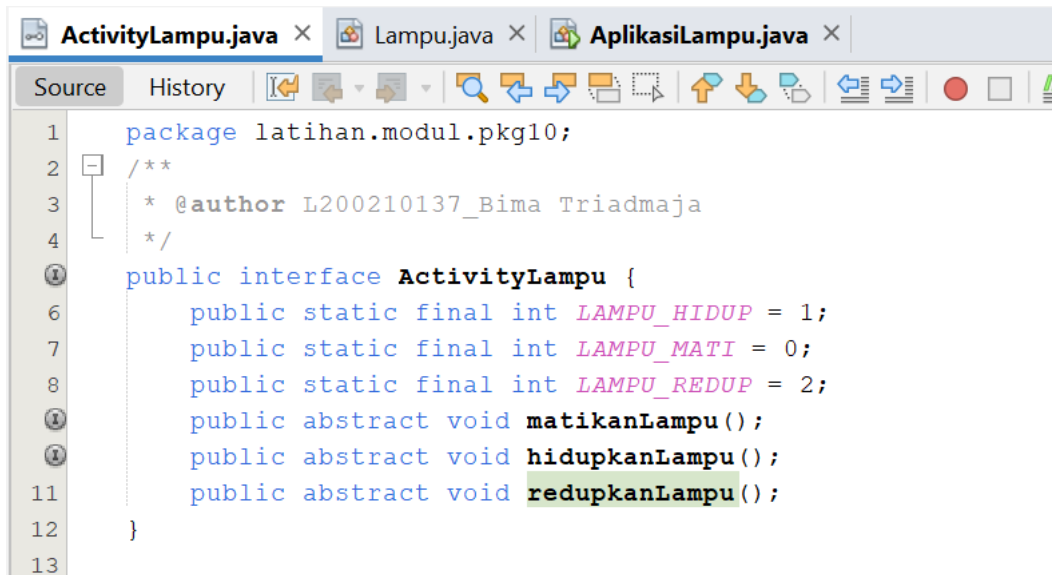
1
Lampu sudah dinyalakan
***
BUILD SUCCESSFUL (total time: 2 minutes 49 seconds)
```

10.3. TUGAS

Modifikasi class Lampu di atas dengan menambahkan satu variabel static final LAMPU_REDUP, dan tambahkan method untuk meredupkan lampu. Selanjutnya buat class dengan fungsi main () untuk menjalankannya!

❖ Hasil Kegiatan :

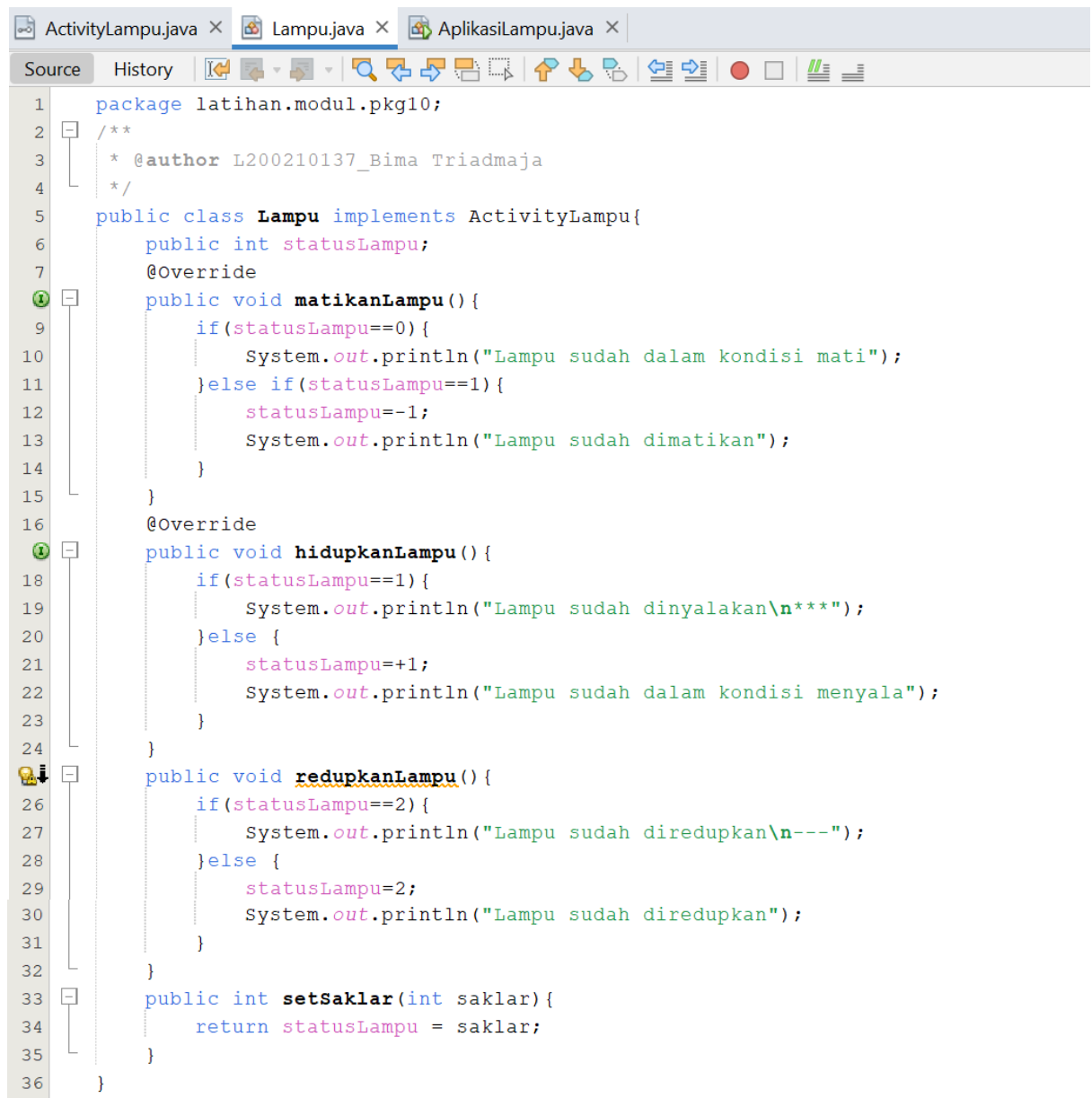
Class ActivityLampu.java (interface) :



The screenshot shows an IDE with a single tab: 'ActivityLampu.java'. The source code is as follows:

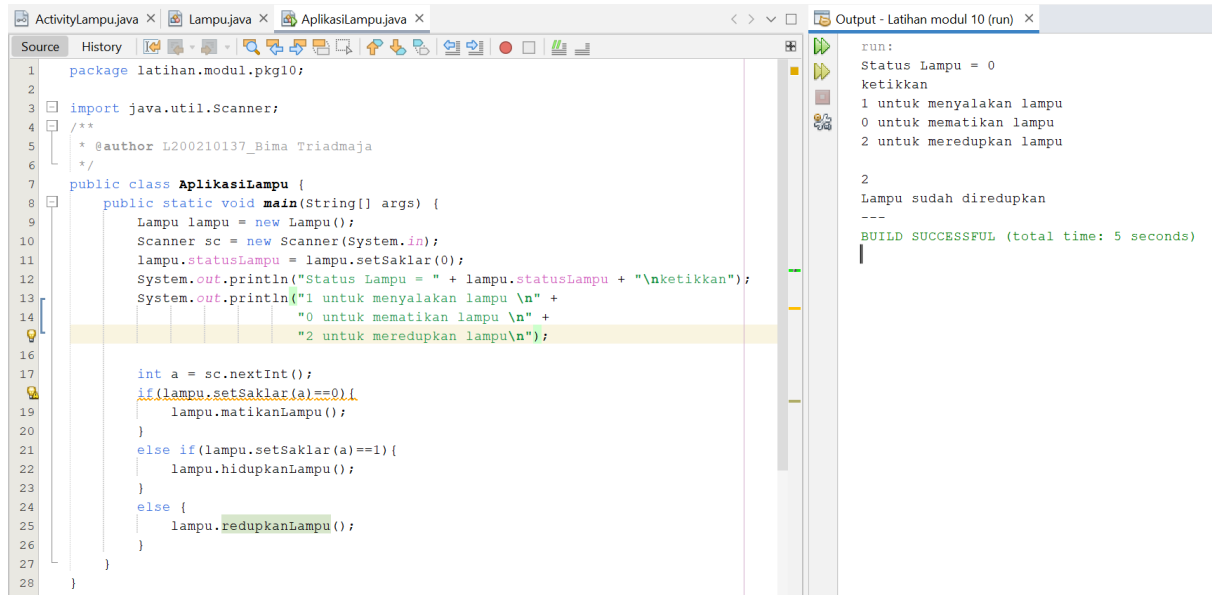
```
1 package latihan.modul.pkg10;
2
3 /**
4  * @author L200210137_Bima Triadmaja
5  */
6 public interface ActivityLampu {
7     public static final int LAMPU_HIDUP = 1;
8     public static final int LAMPU_MATI = 0;
9     public static final int LAMPU_REDUP = 2;
10     public abstract void matikanLampu();
11     public abstract void hidupkanLampu();
12     public abstract void redupkanLampu();
13 }
```

Class Lampu.java (implement dari ActivityLampu) :



```
1 package latihan.modul.pkg10;
2 /**
3  * @author L200210137_Bima Triadmaja
4  */
5 public class Lampu implements ActivityLampu{
6     public int statusLampu;
7     @Override
8     public void matikanLampu() {
9         if(statusLampu==0){
10             System.out.println("Lampu sudah dalam kondisi mati");
11         }else if(statusLampu==1){
12             statusLampu=-1;
13             System.out.println("Lampu sudah dimatikan");
14         }
15     }
16     @Override
17     public void hidupkanLampu() {
18         if(statusLampu==1){
19             System.out.println("Lampu sudah dinyalakan\n***");
20         }else {
21             statusLampu=+1;
22             System.out.println("Lampu sudah dalam kondisi menyala");
23         }
24     }
25     public void redupkanLampu() {
26         if(statusLampu==2){
27             System.out.println("Lampu sudah diredupkan\n--");
28         }else {
29             statusLampu=2;
30             System.out.println("Lampu sudah diredupkan");
31         }
32     }
33     public int setsaklar(int saklar){
34         return statusLampu = saklar;
35     }
36 }
```

Class AplikasiLampu.java (aplikasi dari Lampu) dan Output :



The screenshot displays an IDE with two tabs: 'AplikasiLampu.java' and 'Output - Latihan modul 10 (run)'. The 'Source' tab shows the Java code for the 'AplikasiLampu' class, which uses a Scanner to read user input and control a lamp's status. The 'Output' tab shows the program's execution results, including the initial status, instructions for using the program, and the final status after input.

```
1 package latihan.modul.pkg10;
2
3 import java.util.Scanner;
4
5 /**
6  * @author L200210137_Bima Triadmaja
7  */
8 public class AplikasiLampu {
9     public static void main(String[] args) {
10         Lampu lampu = new Lampu();
11         Scanner sc = new Scanner(System.in);
12         lampu.statusLampu = lampu.setSaklar(0);
13         System.out.println("Status Lampu = " + lampu.statusLampu + "\nketikkan");
14         System.out.println("1 untuk menyalakan lampu \n" +
15             "0 untuk mematikan lampu \n" +
16             "2 untuk meredupkan lampu\n");
17
18         int a = sc.nextInt();
19         if (lampu.setSaklar(a)==0){
20             lampu.matikanLampu();
21         }
22         else if (lampu.setSaklar(a)==1){
23             lampu.hidupkanLampu();
24         }
25         else {
26             lampu.redupkanLampu();
27         }
28     }
29 }
```

run:
Status Lampu = 0
ketikkan
1 untuk menyalakan lampu
0 untuk mematikan lampu
2 untuk meredupkan lampu

2
Lampu sudah diredukan

BUILD SUCCESSFUL (total time: 5 seconds)