

Nama : La Ode Muhammad Gazali
NIM : 222212696
Kelas : 2KS2

MODUL 12 PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (Threading)

Penugasan

Laporkan hasil praktikum berikut dengan hasil penugasan dan penjelasannya ke Dosen dalam bentuk file pdf dengan format nama <<nim>>_modul12.

1. Lengkapi kode semua objek
2. Tangkapan layar hasil kode yang dilengkapi
3. Tangkapan layar hasil running

Penyelesaian

Pada praktikum kali ini akan dibuat sebuah projek bary dengan nama Latihan Thread. Kemudian modifikasi LatihanThread.java menjadi seperti berikut:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package latihan.thread;
6
7  /**
8   *
9   * @author U53R
10  */
11 public class LatihanThread {
12     /**
13      * @param args the command line arguments
14      */
15     public static void main(String[] args) {
16         // TODO code application logic here
17         Thread t = new Thread(new Runnable() {
18             @Override
19             public void run() {
20                 for (int i=0; i<10;i++){
21                     System.out.println("Thread t: "+ i );
22                 }
23             }
24         });
25         t.start();
26         for (int i=0; i<10;i++){
27             System.out.println("Main Thread: "+ i );
28         }
29     }
30 }
```

Ketika dijalankan maka outputnya sebagai berikut

```
run:
Thread t: 0
Thread t: 1
Thread t: 2
Thread t: 3
Thread t: 4
Thread t: 5
Thread t: 6
Thread t: 7
Thread t: 8
Thread t: 9
Main Thread: 0
Main Thread: 1
Main Thread: 2
Main Thread: 3
Main Thread: 4
Main Thread: 5
Main Thread: 6
Main Thread: 7
Main Thread: 8
Main Thread: 9
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ketika `t.start()` dipanggil, penjadwalan oleh JVM atau sistem operasi memungkinkan Thread `t` untuk mulai eksekusi segera dan menyelesaikan tugasnya tanpa interupsi Main Thread, meskipun tidak menunggu Thread `t` secara eksplisit, mendapatkan kembali kontrol setelah Thread `t` menyelesaikan tugasnya.

Terdapat 2 cara untuk mengimplementasikan Thread pada Java. Yang pertama adalah dengan meng-implement interface `Runnable`, dan yang lainnya adalah dengan meng-extend kelas `Thread`.

Buatlah kelas java baru dengan nama `ThreadA.java`, salinlah kode berikut ini

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
4   */
5   package latihan.thread;
6
7   /**
8    *
9    * @author U53R
10  */
11  public class ThreadA implements Runnable{
12      @Override
13      public void run() {
14          for (int i=0; i<10;i++){
15              System.out.println("Thread A: "+ i );
16          }
17      }
18  }
```

Kemudian buatlah kelas baru dengan nama ThreadB.java dengan kode berikut:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
4   */
5   package latihan.thread;
6
7   /**
8    *
9    * @author U53R
10   */
11  public class ThreadB extends Thread{
12      @Override
13      public void run(){
14          for (int i=0; i<10;i++){
15              System.out.println("Thread B: "+ i );
16          }
17      }
18  }
```

Kemudian modifikasilah kelas utama LatihanThread menjadi seperti berikut:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5   package latihan.thread;
6
7   /**
8    *
9    * @author U53R
10   */
11  public class LatihanThread {
12      /**
13       * @param args the command line arguments
14       */
15      public static void main(String[] args) {
16          // TODO code application logic here
17
18          Thread t = new Thread(new Runnable(){
19              @Override
20              public void run() {
21                  for (int i=0; i<10;i++){
22                      System.out.println("Thread t: "+ i );
23                  }
24              }
25          });
26          t.start();
27          Thread t2 = new Thread(new ThreadA());
28          t2.start();
29
30          Thread t3 = new ThreadB();
31          t3.start();
32
33          for (int i=0; i<10;i++){
34              System.out.println("Main Thread: "+ i );
35          }
36      }
37  }
```

```
run:
Thread t: 0
Main Thread: 0
Thread A: 0
Thread B: 0
Thread A: 1
Main Thread: 1
Thread t: 1
Main Thread: 2
Thread A: 2
Thread B: 1
Thread A: 3
Main Thread: 3
Thread t: 2
Main Thread: 4
Thread A: 4
Main Thread: 5
Thread B: 2
Main Thread: 6
Thread A: 5
Thread A: 6
Thread A: 7
Thread A: 8
Thread A: 9
Main Thread: 7
Main Thread: 8
Thread t: 3
Main Thread: 9
Thread B: 3
Thread B: 4
Thread B: 5
Thread B: 6
Thread B: 7
Thread B: 8
Thread B: 9
Thread t: 4
Thread t: 5
Thread t: 6
Thread t: 7
Thread t: 8
Thread t: 9
BUILD SUCCESSFUL (total time: 0 seconds)
```

Terlihat bahwa keempat thread yaitu A,B,C dan main Thread berjalan secara **bersamaan**