_____MultiThread_____

Nama : Muhamad Wahyu Saputra

Kelas : XII RPL B

Absen : 34

LATIHAN 1

demopackage.java | tryClass.java | tryClass2.java | ReadWrite1.java | CobaThreadRunnable.java | CobaThread.java | latihan6.java

Source | History

```
1  package multithread;
2  public class tryClass {
3      public static void main(String[] args) {
4          thread1 T1 = new thread1();
5          thread2 T2 = new thread2();
6          T1.start();
7          T2.start();
8      }
9  }
10 class thread1 extends Thread{
11     @Override
12     public void run(){
13         while(true){
14             System.out.println("thread 1 running");
15             try{
16                 Thread.sleep(1000);
17             }catch (InterruptedException IE){
18                 System.out.println("Thread interupted");
19             }
20         }
21     }
22 }
23 .
24 class thread2 extends Thread{
25     @Override
26     public void run(){
27         for(int i = 0; i < 5; i++){
28             System.out.println("Thread 2 running " + i);
29             try{
30                 Thread.sleep(500);
31             }catch(InterruptedException IE){
32                 System.out.println("Thread interrupted");
33             }
34         }
35         System.out.println("Thread 2 completed");
36     }
37 }
```

Output - MULTITHREAD (run)

```
run:
thread 1 running
Thread 2 running 0
Thread 2 running 1
Thread 2 running 2
thread 1 running
Thread 2 running 3
Thread 2 running 4
thread 1 running
Thread 2 completed
thread 1 running
thread 1 running
thread 1 running
thread 1 running
thread 1 running
BUILD STOPPED (total time: 7 seconds)
```

LATIHAN 2

```java
package multithread;
public class tryClass2 {
    public static void main(String[] args) {
        thread11 th1 = new thread11();
        thread22 th2 = new thread22();
        Thread T1 = new Thread(th1);
        Thread T2 = new Thread(th2);
        T1.start();
        T2.start();
    }

}

class thread11 implements Runnable{
    @Override
    public void run(){
        while(true){
            System.out.println("thread 1 running");
            try{
                Thread.sleep(1000);
            }catch (InterruptedException IE){
                System.out.println("Thread interupted");
            }
        }
    }
}

 class thread22 implements Runnable{
     @Override
     public void run(){
         for(int i = 0; i < 5; i++){
             System.out.println("Thread 2 running " + i);
             try{
                 Thread.sleep(500);
             }catch(InterruptedException IE){
                 System.out.println("Thread interrupted");
             }
         }
         System.out.println("Thread 2 completed");
     }
 }
```

```
run:
thread 1 running
Thread 2 running 0
Thread 2 running 1
Thread 2 running 2
thread 1 running
Thread 2 running 3
Thread 2 running 4
thread 1 running
Thread 2 completed
thread 1 running
thread 1 running
thread 1 running
thread 1 running
thread 1 running
thread 1 running
thread 1 running
BUILD STOPPED (total time: 10 seconds)
```

LATIHAN 3

```java
package multithread;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ReadWrite1 implements Runnable {

public static int countwriter = 0;
public static int countreader = 0;

public static final int READER = 0;
public static final int WRITER = 1;

private int type;
private int id;
private String strid;

public ReadWrite1(int type){
    this.type = type;
    this.strid = (type == READER) ? ("reader "+ (this.id = ++countreader)) :  ("Writer " + (this.id = ++countwriter));
}

    public static void main(String[] args) {
        (new Thread(new ReadWrite1(READER))).start();
        (new Thread(new ReadWrite1(READER))).start();
        (new Thread(new ReadWrite1(WRITER))).start();
    }
```

```
    public void run(){
        if (type ==READER){
            read();
        }else if(type == WRITER){
            write();
        }
    }


    public void read(){
        System.out.println(strid + " : start!");
        while(true){
            System.out.println(strid +" : Reading");
            try{
                Thread.sleep(1000);
            }catch(InterruptedException ex){
                Logger.getLogger(ReadWritel.class.getName()).log(Level.SEVERE,null,ex);
            }
        }
    }


    public void write(){
        System.out.println(strid + " : start!");
        while(true){
            System.out.println(strid +" : writing");
            try{
                Thread.sleep(1000);
            }catch(InterruptedException ex){
                Logger.getLogger(ReadWritel.class.getName()).log(Level.SEVERE,null,ex);
            }
        }
    }

}
```

```
Output - MULTITHREAD (run)
reader 1 : Reading
Writer 1 : writing
reader 2 : Reading
Writer 1 : writing
reader 1 : Reading
reader 2 : Reading
reader 1 : Reading
reader 2 : Reading
Writer 1 : writing
reader 2 : Reading
Writer 1 : writing
reader 1 : Reading
Writer 1 : writing
reader 1 : Reading
reader 2 : Reading
```

MULTITHREAD (run) 56:43 INS

LATIHAN 4

Source History

```
package multithread;
public class CobaThreadRunnable {
    public static void main(String[] args) {
        Thread buku1 = new Thread(new Buku("Buku 1"));
        Thread buku2 = new Thread(new Buku("Buku 2"));
        buku1.start();
        buku2.start();
    }
}
class Buku implements Runnable {
    String nama;
    //konstruktor
    public Buku(String id){
        nama = id;
    }
    public void run(){
        for(int i = 1; i < 9; i++ ){
            try{
                Thread.currentThread().sleep(1000);
            }catch (InterruptedException ie){
                System.out.println("Terinterupsi");
            }
            System.out.println("Thread " + nama + " iposisi " + 1);
        }
    }
}
```

```
Output - MULTITHREAD (run)
run:
Thread Buku 1 iposisi 1
Thread Buku 2 iposisi 1
Thread Buku 2 iposisi 2
Thread Buku 1 iposisi 2
Thread Buku 2 iposisi 3
Thread Buku 1 iposisi 3
Thread Buku 2 iposisi 4
Thread Buku 1 iposisi 4
BUILD SUCCESSFUL (total time: 4 seconds)
```

```java
package multithread;
public class CobaThread {
    public static void main(String[] args) {
        Topi topi1 = new Topi("Topi-1");
        Topi topi2 = new Topi("Topi-2");
        topi1.start();
        topi2.start();
    }
}
    class Topi extends Thread{
        //konstruktor
        public Topi (String id){
            super (id);
        }
        //mendefinisikan sendiri run
        @Override
        public void run() {
            String nama = getName();
            for(int i = 1; i < 5; i++){
                try{
                    sleep(1000);//tunggu sedetik
                }catch(InterruptedException ie){
                    System.out.println("Terinterupsi");
                }
                System.out.println("Thread " + nama + " :posisi " + i);
            }
        }
    }
```

```
Output - MULTITHREAD (run)
run:
Thread Topi-2 :posisi 1
Thread Topi-1 :posisi 1
Thread Topi-1 :posisi 2
Thread Topi-2 :posisi 2
Thread Topi-1 :posisi 3
Thread Topi-2 :posisi 3
Thread Topi-2 :posisi 4
Thread Topi-1 :posisi 4
BUILD SUCCESSFUL (total time: 4 seconds)
```

LATIHAN 6

Silahkan membuat program multithread dengan mengacu contoh latihan program diatas

```java
package multithread;
import static java.lang.Thread.sleep;
public class latihan6 {
    public static void main(String[] args) {
        Thread kitab1 = new Thread(new Kitab("Kitab Ta'lim"));
        Thread kitab2 = new Thread(new Kitab("Buku Tarikh"));
        Kopyah kopyah1 = new Kopyah("kopyah putih-1");
        Kopyah kopyah2 = new Kopyah("kopyah hitam-2");
        kopyah1.start();
        kopyah2.start();
        kitab1.start();
        kitab2.start();
    }
}
  class Kopyah extends Thread{
        //konstruktor
        public Kopyah (String id){
            super (id);
        }
        //mendefinisikan sendiri run

        @Override
        public void run() {
            String nama = getName();
            for(int i = 1; i < 5; i++){
                try{
                    sleep(1000);//tunggu sedetik
                }catch(InterruptedException ie){
                    System.out.println("Terinterupsi");
                }
                System.out.println("Thread " + nama + " :posisi" + i);
                System.out.println("----------------------------");
            }
        }
    }
```

```
36
37
38  class Kitab implements Runnable {
39      String nama;
40      //konstruktor
41      public Kitab(String id){
42          nama = id;
43      }
44
45      public void run(){
46          for(int i = 1; i < 5; i++ ){
47              try{
48                  Thread.currentThread().sleep(1000);
49              }catch (InterruptedException ie){
50                  System.out.println("Terinterupsi");
51              }
52              System.out.println("Thread " + nama + " iposisi " + i);
53          }
54      }
55
56  }
57
58
```

```
Output - MULTITHREAD (run)
run:
Thread kopyah putih-1 :posisi1
Thread kopyah hitam-2 :posisi1
---------------------------
---------------------------
Thread Kitab Ta'lim iposisi 1
Thread Buku Tarikh iposisi 1
Thread kopyah hitam-2 :posisi2
Thread Buku Tarikh iposisi 2
Thread Kitab Ta'lim iposisi 2
Thread kopyah putih-1 :posisi2
---------------------------
---------------------------
Thread Buku Tarikh iposisi 3
Thread Kitab Ta'lim iposisi 3
Thread kopyah putih-1 :posisi3
---------------------------
Thread kopyah hitam-2 :posisi3
---------------------------
Thread Buku Tarikh iposisi 4
Thread Kitab Ta'lim iposisi 4
Thread kopyah putih-1 :posisi4
Thread kopyah hitam-2 :posisi4
---------------------------
---------------------------
BUILD SUCCESSFUL (total time: 4 seconds)
```

==========()==========()==========

==========----------==========

----------(----------)----------
