

TUGAS SISTEM TERSEBAR MULTITHREAD PROGRAMMING

Diajukan untuk memenuhi tugas Sistem Tersebar



Disusun Oleh:

Muhammad Raihan Nur Azmii (1193020)

D3 Teknik Informatika 3A

**PROGRAM STUDI DIPLOMA III JURUSAN TEKNIK INFORMATIKA
POLITEKNIK POS INDONESIA**

2022

A. Praktikum

1. Tulis dan jalankan Contoh 6.1 dan Contoh 6.2 pada modul ini dan pahami tiap barisnya!
2. Program 1
 - a) Buatlah sebuah class Kirim yang merupakan turunan dari class thread dimana dalam kelas ini terdapat prosedur yang pertama kali dijalankan. Keluaran dari proses merupakan bilangan ganjil dan ditampilkan ke layar yang di ulang sebanyak 10 kali.
 - b) Lalu buatlah class Terima yang juga merupakan turunan dari class thread dimana dalam kelas ini terdapat prosedur yang pertama kali dijalankan. Keluaran dari proses merupakan bilangan genap dan ditampilkan ke layar yang di ulang sebanyak 10 kali.
 - c) Buatlah kelas utama dengan nama ThreadUtama yang didalamnya menjalankan kelas terima dan kelas kirim secara bersama sama
3. Buatlah class-class seperti pada program 1 hanya saja class Kirim dan class Terima mengimplementasikan interface Runnable
4. Tulis dan jalankan Contoh 6.3, Contoh 6.4, Contoh 6.5, dan Contoh 6.6 pada modul ini dan pahami tiap barisnya!

B. Teori

1. Buatlah program yang mampu menjumlahkan 3 buah Array bertipe integer berikut:

Array 1: 6, 9, 1, 2, 3, 5

Array 2: 7, 11, 6, 4, 3, 1

Array 3: 5, 4, 3, 2, 1, 12

----- +

Hasil = 18, 24, 8, 7, 18

Dengan aturan terdapat 6 thread dimana thread 1-6 tugasnya menjumlahkan array sesuai dengan indexnya

Misalkan:

Thread 1 = 6 + 7 + 5 = 18

Thread 2 = 9 + 11 + 4 = 24

...dst

Total: 75

2. Buatlah program java dengan menggunakan thread dengan hasil output menceritakan kisah sebagai berikut: Disuatu rumah sakit terdapat 1 dokter, 2 perawat dan 1 kasir. Datanglah 10 pasien kemudian diperiksa oleh dokter. Setelah 10 pasien diperiksa dokter, perawat 1 datang dan merawat 5 pasien dan perawat 2 merawat sisa pasien lain. Setelah dirawat, pasien langsung membayar biaya perawatan dikasir.

JAWABAN

A. Praktikum

1. Praktikum 6.1 dan 6.2

- Contoh 6.1, membuat aplikasi multithread dengan kelas thread

➤ Program

```
1  ...4 lines
5  package Praktikum6;
6
7      public class ExtendThread extends Thread {
8
9          int threadNumber;
10
11     public ExtendThread (int num) {
12         threadNumber = num;
13     }
14
15     //Run method is executed when thread first started
16     public void run () {
17         System.out.println ("I'm thread number: " + threadNumber);
18         try {
19             Thread.sleep(5000);
20         } catch (InterruptedException ex) {
21             System.out.println (threadNumber + " is finished");
22         }
23         System.out.println (threadNumber + " is finished!");
24     }
25     public static void main (String[] args) {
26         System.out.println ("Creating thread 1");
27         Thread t1 = new ExtendThread (1) ;
28
29         System.out.println ("Creating thread 2");
30         Thread t2 = new ExtendThread (2);
31
32         //start both threads
33         t1.start ();
34         t2.start ();
35     }
36 }
```

➤ Output

```
Creating thread 1
Creating thread 2
I'm thread number: 1
I'm thread number: 2
1 is finished!
2 is finished!
BUILD SUCCESSFUL (total time: 8 seconds)
```

- Contoh 6.2

- Program

```
1  ...4 lines
5  package Praktikum6;
6
7      public class RunnableThread implements Runnable{
8          // Run method is executed when thread first started
9      public void run () {
10         System.out.println("I am an instance of the java.lang. Runnable interface");
11     }
12
13     public static void main (String args []) {
14         System.out.println ("Creating runnable object");
15
16         Runnable run = (Runnable) new RunnableThread();
17
18         System.out.println("Creating First Thread");
19         Thread t1 = new Thread (run);
20
21         System.out.println("Creating Second Thread");
22         Thread t2 = new Thread (run);
23
24         System.out.println("Starting Both Threads");
25         t1.start();
26         t2.start();
27     }
28 }
```

- Output

```
run:
Creating runnable object
Creating First Thread
Creating Second Thread
Starting Both Threads
I am an instance of the java.lang. Runnable interface
I am an instance of the java.lang. Runnable interface
BUILD SUCCESSFUL (total time: 1 second)
```

2. Praktikum Program 1

- Contoh buat class kirim looping ganjil

➤ Program

```
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 Praktikum6;
6
7   /**...4 lines */
11  public class Kirim extends Thread{
12      public static void main(String args[]) throws java.io.IOException{
13          int count = 1;
14          System.out.println ("Bilangan Ganjil");
15          for (int i = 0; i < 10; i++){
16              System.out.print(count + " ");
17              count = count + 2;
18              try{
19                  Thread.sleep(500);
20              } catch (InterruptedException ie){
21              }
22          }
23      }
24  }
```

➤ Output

```
run:
Bilangan Ganjil
1 3 5 7 9 11 13 15 17 19 BUILD SUCCESSFUL (total time: 5 seconds)
```

- Contoh buat class terima looping Genap

➤ Program

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit t
4  */
5  package Praktikum6;
6
7  /**
8   *
9   * @author ASUS
10  */
11  public class Terima {
12      public static void main(String args[]) throws java.io.IOException{
13          int count = 0;
14          System.out.println ("Bilangan Genap");
15          for (int i = 0; i < 10; i++){
16              System.out.print(count + " ");
17              count = count + 2;
18              try{
19                  Thread.sleep(500);
20              } catch (InterruptedException ie){
21              }
22          }
23      }
24  }
```

➤ Output

```
run:
Bilangan Genap
0 2 4 6 8 10 12 14 16 18 BUILD SUCCESSFUL (total time: 5 seconds)
|
```

- Thread utama menjalankan kelas terima dan kelas kirim

➤ Program

```

4  */
5  package Praktikum6;
6
7  /**...4 lines */
8
9  class ThreadUtama {
10     protected String title = "Thread Utama";
11     public void kirim() {
12         int count = 1;
13         System.out.println ("Bilangan Ganjil");
14         for (int i = 0; i < 10; i++){
15             System.out.print(count + " ");
16             count = count + 2;
17             try{
18                 Thread.sleep(500);
19             } catch (InterruptedException ie){
20             }
21         }
22     }
23     public void terima() {
24         int count = 0;
25         System.out.println ("\nBilangan Genap");
26         for (int i = 0; i < 10; i++){
27             System.out.print(count + " ");
28             count = count + 2;
29             try{
30                 Thread.sleep(500);
31             } catch (InterruptedException ie){
32             }
33         }
34     }
35 }
36
37 class Main extends ThreadUtama {
38     private String random = "Praktikum Program 1";
39
40     for (int i = 0; i < 10; i++){
41         System.out.print(count + " ");
42         count = count + 2;
43         try{
44             Thread.sleep(500);
45         } catch (InterruptedException ie){
46         }
47     }
48 }
49
50 class Main extends ThreadUtama {
51     private String random = "Praktikum Program 1";
52     public static void main(String[] args) {
53         Main myThreadUtama = new Main();
54
55         System.out.println(myThreadUtama.title + " - " + myThreadUtama.random);
56         myThreadUtama.kirim();
57         myThreadUtama.terima();
58     }
59 }

```

➤ Output

```

run:
Thread Utama - Praktikum Program 1
Bilangan Ganjil
1 3 5 7 9 11 13 15 17 19
Bilangan Genap
0 2 4 6 8 10 12 14 16 18 BUILD SUCCESSFUL (total time: 10 seconds)

```

3. Praktikum membuat class class seperti pada program 1 mengimplementasikan runnable

- Runnable Kirim

➤ Program

```
5
6 package Praktikum6;
7
8 /**
9  *
10  * @author ASUS
11  */
12 public class RunnableTerima extends Thread{
13     public void run () {
14         int count = 0;
15         System.out.println ("Ganjil");
16         for (;;) {
17             System.out.print(count + " ");
18             count = count + 2;
19             try{
20                 Thread.sleep(500);
21             } catch (InterruptedException ie) {
22             }
23         }
24     }
25
26     public static void main(String args[]) throws java.io.IOException{
27         // Create and start counting thread
28         Thread counter = new RunnableKirim ();
29         counter.start ();
30         System.out.println ("Press any enter to stop the thread counting");
31         System.in.read();
32         counter.stop();
33     }
34 }
```

➤ Output

Ketika kita tekan enter maka akan berhenti

```
run:
Press any enter to stop the thread counting
Ganjil
1 3 5 7 9 11 13 15
BUILD SUCCESSFUL (total time: 4 seconds)
```


- Runnable Terima

- Program

```
5 package Praktikum6;
6
7 /**
8  *
9  * @author ASUS
10 */
11 public class RunnableKirim extends Thread{
12     public void run () {
13         int count = 1;
14         System.out.println ("Ganjil");
15         for (;;) {
16             System.out.print(count + " ");
17             count = count + 2;
18             try{
19                 Thread.sleep(500);
20             } catch (InterruptedException ie) {
21             }
22         }
23     }
24
25     public static void main(String args[]) throws java.io.IOException{
26         // Create and start counting thread
27         Thread counter = new RunnableKirim ();
28         counter.start ();
29         System.out.println ("Press any enter to stop the thread counting");
30         System.in.read();
31         counter.stop();
32     }
33 }
34
```

- Output

```
run:
Press any enter to stop the thread counting
Ganjil
1 3 5 7 9 11 13
BUILD SUCCESSFUL (total time: 4 seconds)
|
```

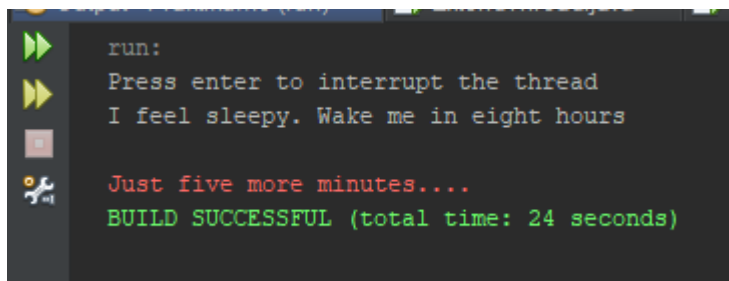
4. Praktikum 6.3, 6.4, 6.5, 6.6

- Contoh 6.3

➤ Program InterruptingThread

```
4 package Praktikum6;
5
6
7 public class InterruptingThread extends Thread {
8     public void run () {
9         System.out.println ("I feel sleepy. Wake me in eight hours");
10        try {
11            Thread.sleep (1000 * 60 * 60 * 8);
12            System.out.println ("That was a nice nap");
13        } catch (InterruptedException ie) {
14            System.err.println ("Just five more minutes....");
15        }
16    }
17
18    public static void main (String args[]) throws java.io.IOException {
19
20        Thread sleepy = new InterruptingThread ();
21
22        sleepy.start ();
23
24        System.out.println ("Press enter to interrupt the thread");
25        System.in.read ();
26
27        sleepy.interrupt ();
28    }
29 }
```

➤ Output



```
run:
Press enter to interrupt the thread
I feel sleepy. Wake me in eight hours
Just five more minutes....
BUILD SUCCESSFUL (total time: 24 seconds)
```

- Contoh 6.4

- Program StoppingThread

```
5 package Praktikum6;
6
7 public class StoppingThread extends Thread {
8
9     // Run method is executed when thread first started
10    public void run () {
11        int count = 1;
12        System.out.println ("I can count. Watch me go!");
13        for (;;) {
14            // Print count and increment it
15            System.out.print(count++ + " ");
16            // Sleep for half a second
17            try {
18                Thread.sleep(500);
19            } catch (InterruptedException ie) {}
20        }
21    }
22
23
24    public static void main(String args[]) throws java.io.IOException{
25        // Create and start counting thread
26        Thread counter = new StoppingThread ();
27        counter.start ();
28        // Prompt user and wait for input
29        System.out.println ("Press any enter to stop the thread counting");
30        System.in.read();
31        // Interrupt the thread
32        counter.stop();
33    }
34 }
```

- Output

```
run:
Press any enter to stop the thread counting
I can count. Watch me go!
1 2 3 4 5 6 7 8 9 10 11 12 13 14
BUILD SUCCESSFUL (total time: 8 seconds)
```

- Contoh 6.5

- Program SuspendingThread

```
1  ...4 lines
5  package Praktikum6;
6
7  /**...4 lines */
11 public class SuspendingThread extends Thread {
12     // Run method is executed when thread first started
13     public void run () {
14         int count = 1;
15         System.out.println ("I can count. Watch me go!");
16         for (;;) {
17             System.out.print (count++ + " ");
18             try {
19                 Thread.sleep (500);
20             } catch (InterruptedException ie) {}
21         }
22     }
23
24
25     public static void main(String args[]) throws java.io.IOException {
26         Thread counter = new SuspendingThread ();
27         counter.start();
28         System.out.println ("Press any enter to SUSPEND the thread counting");
29         System.in.read();
30
31         counter.suspend ();
32         System.out.println ("Press any enter to RESUME the thread counting");
33         System.in.read();
34         counter.resume ();
35
36         System.out.println ("Press any enter to STOP the thread counting");
37         System.in.read();
38         counter.stop ();
39     }
40 }
```

- Output

```
run:
Press any enter to SUSPEND the thread counting
I can count. Watch me go!
1 2 3 4 5 6 7 8
9 Press any enter to RESUME the thread counting

Press any enter to STOP the thread counting
10 11 12 13
BUILD SUCCESSFUL (total time: 9 seconds)
|
```

- Contoh 6.6

- Program WaitingThreadDead

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to cha
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this tem
4  */
5  package Praktikum6;
6
7  /**
8   *
9   * @author ASUS
10  */
11  public class WaitingThreadDead extends Thread {
12      // Run method is executed when thread first started
13      public void run () {
14          System.out.println ("This thread feels a little ill....");
15          // Sleep for five seconds
16          try {
17              Thread.sleep (5000) ;
18          } catch (InterruptedException ie) {
19              ;
20          }
21      }
22
23      public static void main(String args[]) throws java.lang.InterruptedException {
24          // Create and start dying thread
25          Thread dying = new WaitingThreadDead ();
26          dying.start ();
27          // Prompt user and wait for input
28          System.out.println ("Waiting for thread death");
29          // Wait till death
30          dying.join ();
31          System.out.println("Thread has died");
32      }
33  }
```

- Output

```
run:
Waiting for thread death
This thread feels a little ill....

Thread has died
BUILD SUCCESSFUL (total time: 6 seconds)
```

B. Praktikum

1. Tugas

- Program yang mampu menambahkan 3 buah array

➤ Program

Class simpleThread

```
1  //
2  import java.util.Arrays;
3  import java.util.logging.Level;
4  import java.util.logging.Logger;
5
6  public class simpleThread extends Thread{
7
8      private String threadName;
9      private int sum;
10     private int num1, num2, num3;
11
12     public simpleThread(String name){
13         super();
14         threadName = name;
15     }
16
17     public void setNum(int n1, int n2, int n3){
18         num1 = n1;
19         num2 = n2;
20         num3 = n3;
21     }
22
23     public void run(){
24         try{
25             sum = num1 + num2 + num3;
26             System.out.printf("%s calculate %d+%d+%d = %d \n", threadName,num1,num2,num3,sum);
27             Thread.sleep(100);
28         } catch (InterruptedException ex) {
29             Logger.getLogger(simpleThread.class.getName()).log(Level.SEVERE, null, ex);
30         }
31     }
32 }
```

Mainclas. ThreadTester

```
5  package Latihan6;
6
7  /**...4 lines */
8  public class threadTester {
9
10     public static void main(String[] args){
11         int[] arr1 = {6,9,1,2,3,5};
12         int[] arr2 = {7,11,6,4,3,1};
13         int[] arr3 = {5,4,3,2,1,12};
14         for (int h = 0;h < arr1.length; h++){
15             System.out.printf("sum[%d] = %d\n",h, arr1[h] + arr2[h] + arr3[h]);
16         }
17
18         for (int i=0; i < arr1.length ; i++){
19             simpleThread st = new simpleThread("Thread "+i);
20             st.setNum(arr1[i], arr2[i], arr3[i]); st.start();
21         }
22     }
23 }
```

➤ Output

```
run:
sum[0] = 18
sum[1] = 24
sum[2] = 10
sum[3] = 8
sum[4] = 7
sum[5] = 18
Thread 0 calculate 6+7+5 = 18
Thread 3 calculate 2+4+2 = 8
Thread 4 calculate 3+3+1 = 7
Thread 2 calculate 1+6+3 = 10
Thread 1 calculate 9+11+4 = 24
Thread 5 calculate 5+1+12 = 18
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Tugas

- Soal cerita

➤ Program

```
1  ...4 lines
5  package Latihan6;
6
7  /**...4 lines */
11 public class ThreadStory extends Thread {
12     int threadNumber;
13
14     public ThreadStory (int num) {
15         threadNumber = num;
16     }
17
18     public void run() {
19         System.out.println("Dokter 1 menangani pasien-"+ threadNumber);
20
21         try {
22             Thread.sleep(1000);
23         } catch (InterruptedException ex) {
24             System.out.println(threadNumber + "Selesai");
25         }
26         for (int i=1; i<=5;i++){
27
28             //System.out.println("perawat2");
29         }
30         try {
31             Thread.sleep(1000);
32         } catch (InterruptedException ex) {
33             System.out.println(threadNumber + "Selesai");
34         }
35         System.out.println("Pasien ke-"+threadNumber+" membayar dikasir 1");
36
37     }
38
39     public static void main(String args[]) {
40         System.out.println("Cerita: ");
```

```

32         } catch (InterruptedException ex) {
33             System.out.println(threadNumber + "Selesai");
34         }
35         System.out.println("Pasien ke-"+threadNumber+" membayar dikasir 1")
36     }
37 }
38
39 public static void main(String args[]) {
40     System.out.println("Cerita: ");
41     Thread t1= new ThreadStory(1);
42     Thread t2= new ThreadStory(2);
43     Thread t3= new ThreadStory(3);
44     Thread t4= new ThreadStory(4);
45     Thread t5= new ThreadStory(5);
46     Thread t6= new ThreadStory(6);
47     Thread t7= new ThreadStory(7);
48     Thread t8= new ThreadStory(8);
49     Thread t9= new ThreadStory(9);
50     Thread t10= new ThreadStory(10);
51     Thread t11= new ThreadPerawat(1);
52     Thread t12= new ThreadPerawat(2);
53     Thread t13= new ThreadPerawat(3);
54     Thread t14= new ThreadPerawat(4);
55     Thread t15= new ThreadPerawat(5);
56     Thread t16= new ThreadPerawat(6);
57     Thread t17= new ThreadPerawat(7);
58     Thread t18= new ThreadPerawat(8);
59     Thread t19= new ThreadPerawat(9);
60     Thread t20= new ThreadPerawat(10);
61
62     //mulai
63     t1.start();
64     t2.start();

```

```

58     Thread t18= new ThreadPerawat(8);
59     Thread t19= new ThreadPerawat(9);
60     Thread t20= new ThreadPerawat(10);
61
62     //mulai
63     t1.start();
64     t2.start();
65     t3.start();
66     t4.start();
67     t5.start();
68     t6.start();
69     t7.start();
70     t8.start();
71     t9.start();
72     t10.start();
73     t11.start();
74     t12.start();
75     t13.start();
76     t14.start();
77     t15.start();
78     t16.start();
79     t17.start();
80     t18.start();
81     t19.start();
82     t20.start();
83
84 }
85 }
86

```


➤ Output

```
run:
Cerita:
Dokter 1 menangani pasien-1
Dokter 1 menangani pasien-2
Dokter 1 menangani pasien-3
Dokter 1 menangani pasien-4
Dokter 1 menangani pasien-6
Dokter 1 menangani pasien-5
Dokter 1 menangani pasien-7
Dokter 1 menangani pasien-8
Dokter 1 menangani pasien-9
Dokter 1 menangani pasien-10
Perawat 1 menangani pasien-1
Perawat 2 menangani pasien-1
Perawat 1 menangani pasien-2
Perawat 2 menangani pasien-2
Perawat 1 menangani pasien-3
Perawat 2 menangani pasien-3
Perawat 1 menangani pasien-4
Perawat 2 menangani pasien-4
Perawat 1 menangani pasien-5
Perawat 2 menangani pasien-5
Perawat 1 menangani pasien-6
Perawat 2 menangani pasien-6
Perawat 1 menangani pasien-7
Perawat 2 menangani pasien-7
Perawat 1 menangani pasien-8
Perawat 2 menangani pasien-8
Perawat 1 menangani pasien-9
Perawat 2 menangani pasien-9
Perawat 1 menangani pasien-10
Perawat 2 menangani pasien-10
Pasien ke-1 membayar dikasir 1
Pasien ke-2 membayar dikasir 1
Pasien ke-3 membayar dikasir 1
Pasien ke-4 membayar dikasir 1
```

```
Perawat 1 menangani pasien-5
Perawat 2 menangani pasien-5
Perawat 1 menangani pasien-6
Perawat 2 menangani pasien-6
Perawat 1 menangani pasien-7
Perawat 2 menangani pasien-7
Perawat 1 menangani pasien-8
Perawat 2 menangani pasien-8
Perawat 1 menangani pasien-9
Perawat 2 menangani pasien-9
Perawat 1 menangani pasien-10
Perawat 2 menangani pasien-10
Pasien ke-1 membayar dikasir 1
Pasien ke-2 membayar dikasir 1
Pasien ke-3 membayar dikasir 1
Pasien ke-4 membayar dikasir 1
Pasien ke-5 membayar dikasir 1
Pasien ke-7 membayar dikasir 1
Pasien ke-6 membayar dikasir 1
Pasien ke-10 membayar dikasir 1
Pasien ke-8 membayar dikasir 1
Pasien ke-9 membayar dikasir 1
BUILD SUCCESSFUL (total time: 4 seconds)
```