|  |
| --- |
| **POKOK BAHASAN 4** |
| **THREADS** |

**LEMBAR KERJA DAN TUGAS**

1. Mengaplikasikan mekanisme Thread dengan menggunakan fungsi run()

public class SimpleThread extends Thread{

public SimpleThread(String nama) {

super(nama);

}

public void run(){

for (int i=0 ;i<10; i++) {

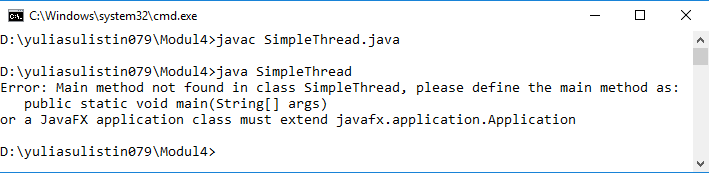
System.out.println(i+""+getName());

}

System.out.println("SELESAI "+getName());

}

}



public class TwoThreadTest {

public static void main (String argv [] ){

SimpleThread t1,t2;

t1 = new SimpleThread ("Surabaya");

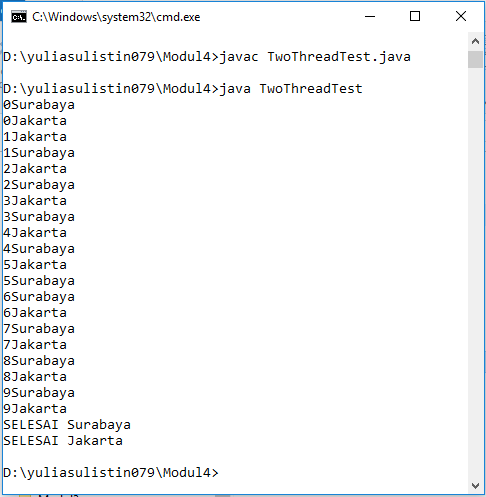
t2 = new SimpleThread ("Jakarta");

t1.start ();

t2.start ();

}

}



1. Melihat concurrency (multitasking) pada program dengan menambahkan

method sleep() untuk aktual Thread (current Thread). Method sleep() hanya dapat dilakukan dengan menangkap signal interrupt dengan mekanisme try & cacth

public class SimpleThread1 extends Thread {

public SimpleThread1 (String nama) {

super (nama);

}

public void run (){

for (int i=0;i<10;i++){

System.out.println(i+" "+getName());

try{

sleep((int)Math.random()\*10000);

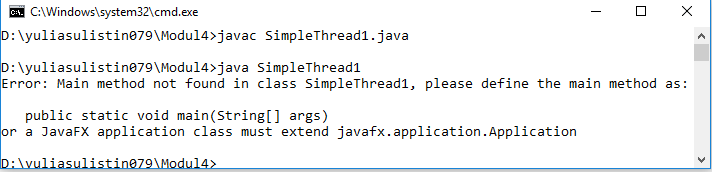
}catch(Exception e){}

}

System.out.println("SELESAI "+getName());

}

}



public class TwoThreadTest1 {

public static void main (String argv [] ){

SimpleThread t1,t2;

t1 = new SimpleThread ("Surabaya");

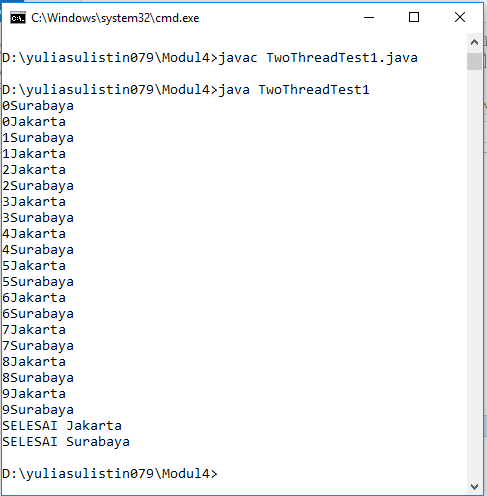
t2 = new SimpleThread ("Jakarta");

t1.start ();

t2.start ();

}

}



1. Mendemonstrasikan animasi yang dilakukan dengan menjalankan Thread

(dalam animasi ini diperlukan file image T1.gif s/d T17.gif)

import java.awt.\*;

public class TThread extends Thread{

Component oby;

int delay;

public TThread(Component oby, int delay){

this.oby=oby;

this.delay=delay;

}

public void run() {

while (true) {

try{

oby.repaint();

sleep(delay);

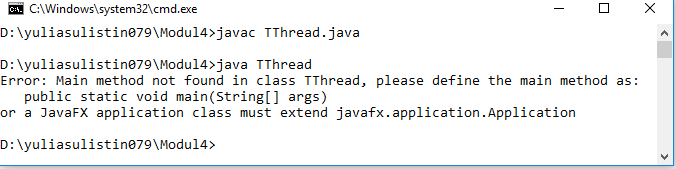
}

catch (Exception e){}

}

}

}



import java.applet.\*;

import java.awt.\*;

public class Animasi extends Applet {

int count, lastcount;

Image gambar[];

TThread timer;

Button btnStart;

Button btnStop;

public void init() {

gambar = new Image[17];

lastcount=17;

count=0;

MediaTracker mt=new MediaTracker(this);

for (int i=0; i<lastcount; i++) {

gambar[i]=getImage(getCodeBase(),"T"+(i+1)+".gif");

mt.addImage(gambar[i],0);

}

mt.checkAll(true);

btnStop=new Button("Stop");

add(btnStop);

btnStart=new Button("Start");

add(btnStart);

show();

}

public void start() {

if (timer==null) {

timer=new TThread(this,75);

timer.start();

showStatus("Start...");

}

}

public void stop() {

if (timer!=null) {

showStatus("Stop...");

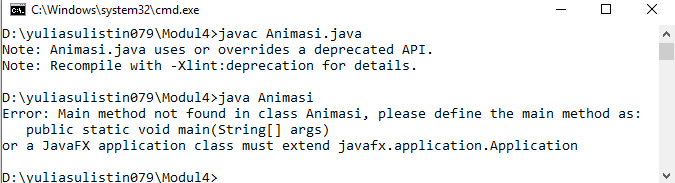
timer.stop();

timer=null;

}

}

}



Setelah mengcompile kedua kelas tersebut, yaitu class TThread dan class Animasi, lalu membungkusnya kedalam file animasi.html. selanjutnya mengeksplore file tersebut dengan internet browser.

<html>

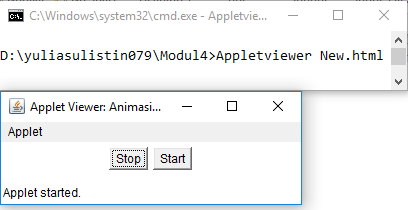
<head>Contoh Animasi</head>

<title>Contoh Animasi</title>

<Applet code = Animasi.class width=300 height=200>

</applet>

</html>



1. Mengimplementasikan Runnable interface

import java.applet.\*;

import java.awt.\*;

import java.lang.\*;

public class Animasi1 extends Applet implements java.lang.Runnable{

int count, lastcount;

Image gambar[];

Button btnStart;

Button btnStop;

Thread timer;

public void init(){

gambar=new Image[17];

lastcount=17;

count=0;

MediaTracker mt=new MediaTracker(this);

for (int i=0; i<lastcount; i++){

gambar[i]=getImage(getCodeBase(),"T"+(i+1)+".gif");

mt.addImage(gambar[i],0);

}

mt.checkAll(true);

btnStop=new Button("Stop");

add(btnStop);

btnStart=new Button("Start");

add(btnStart);

show();

}

public void start(){

if(timer==null){

timer=new Thread(this);

timer.start();

showStatus("Start..");

}

}

public void stop(){

if (timer!=null){

showStatus("Stop..");

timer.stop();

timer=null;

} }

public boolean action(Event e, Object o){

if(e.target==btnStop){

stop();

return true;

}

if(e.target==btnStart){

start();

return true;

}

return false;

}

public void paint(Graphics g){

g.drawImage (gambar[count++],70,70,null);

if(count==lastcount)

count=0;

showStatus("Gambar no"+(count+1));

setLayout(new FlowLayout(FlowLayout.CENTER,5,5));

setBackground(java.awt.Color.lightGray);

//setSize(0,0);

}

public void run(){

while (isActive()){

try{

repaint();

Thread.sleep(100);

}

catch(Exception e){

showStatus("Exception :"+e);

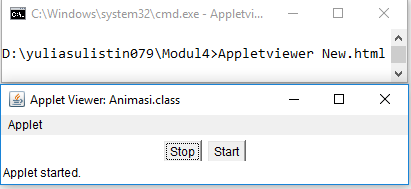
}

}

showStatus("Not Active");

timer=null;

}}



**TUGAS**

1. Buatlah class dalam java yang mengaktifkan tiga thread sekaligus. Tunjukkan

bahwa ketiga thread tersebut bekerja bersama-sama dengan mencetak output ke layar monitor.

class Tugas1 extends Thread{

public Tugas1(String nama){

super(nama);

}

public void run(){

for (int i=0;i<10;i++){

System.out.println(i+" "+getName());

try{

sleep((int)Math.random()\*10000);

}catch(Exception t){}

}

System.out.println("SELESAI "+getName());

}

}

class Tugas1K1 extends Thread{

public Tugas1K1(String nama) {

super(nama);

}

public void run(){

for (int i=0 ;i<10; i++){

System.out.println(i+""+getName());

}

System.out.println("SELESAI "+getName());

}

}

class Tugas1Pembahasan4 {

public static void main (String [] args){

Tugas1 t1=new Tugas1(" Yulia ");

Tugas1K1 t2=new Tugas1K1(" Sulistin");

Tugas1 t3=new Tugas1(" 7 Pagi A");

t1.start();

t2.start();

t3.start();

}

}

