

Modul Praktikum Timer

1. Tujuan

- Memahami dan menggunakan class Timer dan TimerTask
- Mengolah waktu dengan Timer dan TimerTask
- Menentukan waktu yang sesuai untuk penjadwalan
- Dapat membuat animasi dari Timer dan TimerTask

2. Latar Belakang

Class Timer merupakan salah satu pengembangan dari versi Java 1.3, dalam lingkup J2SE berupa pembentukan class-class yang mempermudah pengaturan penjadwalan proses agar dapat dieksekusi melalui pengaksesan fungsi dasar thread. Pembaharuan diatas juga dapat dimanfaatkan oleh pengembang pemrograman berbasis mobile (J2ME) karena class baru diatas juga telah menjadi bagian dalam MIDP (*Mobile Information Device Profile*).

3. Percobaan

Percobaan 1: Membuat Class MidletSplashTimer

```
import java.util.Timer;
import java.util.TimerTask;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;

public class MidletSplashTimer extends MIDlet {
    private SplashTimerCanvas ttCanvas;
    private Display display;
    private Timer timer;
    private TimerTask timerTask;

    public MidletSplashTimer() { }

    protected void startApp() {
        display = Display.getDisplay(this);
        ttCanvas = new SplashTimerCanvas(this);
        display.setCurrent(ttCanvas);
    }
}
```

Modul Praktikum Timer

```
protected void pauseApp() { }

protected void destroyApp(boolean unconditional) { }

void stopTimer() {
    timer.cancel();
    timerTask.cancel();
}

void startTimer() {
    timer = new Timer();
    timerTask = new TimerTask() {
        public void run() {
            display.setCurrent(new SuccesCanvas());
        }
    };
    timer.schedule(timerTask, 3000);
}
```

Percobaan 2 : Membuat Class SplashTimerCanvas

```
import javax.microedition.lcdui.*;

public class SplashTimerCanvas extends Canvas {
    private MidletSplashTimer midletTT;
    int w, h, hImg;
    private Image img;

    public SplashTimerCanvas(MidletSplashTimer midlet) {
        w = getWidth();
        h = getHeight();
        midletTT = midlet;
        try {
            img = Image.createImage("/jeni.png");
        } catch (java.io.IOException e) { }
        hImg = img.getHeight();
    }
}
```

Modul Praktikum Timer

```
protected void paint(Graphics g) {
    g.setColor(255, 255, 255);
    g.fillRect(0, 0, w, h);
    g.drawImage(img, w/2, h/2, Graphics.HCENTER | Graphics.VCENTER);
}

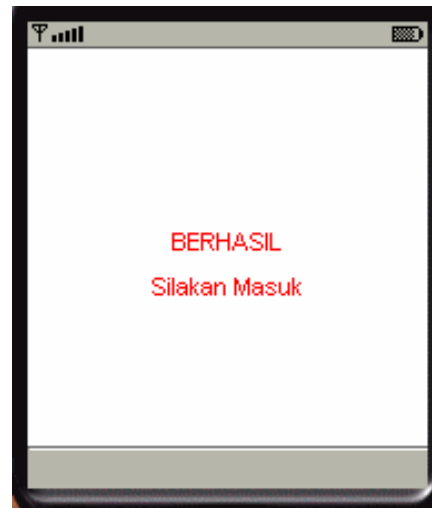
protected void hideNotify() {
    midletTT.stopTimer();
}

protected void showNotify() {
    midletTT.startTimer();
}
}

class SuccesCanvas extends Canvas {
    protected void paint(Graphics g) {
        g.setColor(255, 255, 255);
        g.fillRect(0, 0, getWidth(), getHeight());
        g.setColor(255, 0, 0);
        g.drawString("BERHASIL", getWidth()/2, (getHeight()/2)-10, Graphics.TOP
            | Graphics.HCENTER);
        g.drawString("Silakan Masuk", getWidth()/2, (getHeight()/2)+10,
            Graphics.TOP | Graphics.HCENTER);
    }
}
```

Modul Praktikum Timer

Hasil :



Percobaan 3 : Membuat Class MidletCounterDown

```
import java.util.Timer;
import java.util.TimerTask;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;

public class MidletCounterDown extends MIDlet {
    private CounterDownCanvas ttCanvas;
    private Display display;
    private int count;
    private Timer timer;
    private TimerTask timerTask;

    public MidletCounterDown() {
        count = 0;
    }

    protected void startApp() {
        display = Display.getDisplay(this);
        ttCanvas = new CounterDownCanvas(this);
        display.setCurrent(ttCanvas);
    }
}
```

Modul Praktikum Timer

```
protected void pauseApp() { }

protected void destroyApp(boolean unconditional) { }

private void timerRun() {
    count++;
    ttCanvas.update();
    ttCanvas.repaint(ttCanvas.w/2, ttCanvas.h/2 + ttCanvas.hImg/2, 20,
        20);
}

void stopTimer() {
    timer.cancel();
    timerTask.cancel();
    display.setCurrent(new SuccesCanvas());
}

void startTimer() {
    timer = new Timer();
    timerTask = new TimerTask() {
        public void run() {
            timerRun();
        }
    };
    timer.schedule(timerTask, 1000, 1000);
}
```

Modul Praktikum Timer

Percobaan 4 : Membuat Class CounterDownCanvas

```
import javax.microedition.lcdui.*;

public class CounterDownCanvas extends Canvas {
    private MidletCounterDown midletTT;
    int points;
    int w, h, hImg;
    private Image img;

    public CounterDownCanvas(MidletCounterDown midlet) {
        points = 3;
        w = getWidth();
        h = getHeight();
        midletTT = midlet;
        try {
            img = Image.createImage("/jeni.png");
        } catch (java.io.IOException e) { }
        hImg = img.getHeight();
    }

    protected void paint(Graphics g) {
        g.setColor(255, 255, 255);
        g.fillRect(0, 0, w, h);
        g.drawImage(img, w/2, h/2, Graphics.HCENTER | Graphics.VCENTER);
        g.setColor(0, 0, 255);
        g.drawString(""+ points, w/2, h/2 + hImg/2, Graphics.TOP |
                    Graphics.LEFT);
    }

    protected void hideNotify() {
        midletTT.stopTimer();
    }

    protected void showNotify() {
        midletTT.startTimer();
    }

    protected void update() {
        points--;
        if(points < 1) {
```

Modul Praktikum Timer

```
        hideNotify();
    }
}

class SuccesCanvas extends Canvas {
    protected void paint(Graphics g) {
        g.setColor(255, 255, 255);
        g.fillRect(0, 0, getWidth(), getHeight());
        g.setColor(255, 0, 0);
        g.drawString("BERHASIL", getWidth()/2, (getHeight()/2)-10,
                    Graphics.TOP | Graphics.HCENTER);
        g.drawString("Silakan Masuk", getWidth()/2, (getHeight()/2)+10,
                    Graphics.TOP | Graphics.HCENTER);
    }
}
```

Hasil :

