

Modul Praktikum TiledLayer

1. Tujuan

- Menggambar dan Mengetahui fungsi TiledLayer
- Membuat object dari class TiledLayer
- Menyusun graphics dari TiledLayer dan di masukkan dalam LayerManager
- Manipulasi graphics dengan menggunakan method yang ada pada class TiledLayer

2. Latar Belakang

Setelah kita mempelajari class LayerManager, kita akan mempelajari bagaimana pembuatan TiledLayer pada GameCanvas dan menyusunnya hingga seperti graphics background. Kita akan membuat contoh pembuatan dan menyusun graphics dengan menggunakan TiledLayer.

Disini peserta wajib mengikuti pembuatan class-class dan memodifikasi class yang telah dibuat agar peserta lebih memahami kinerja atau fungsi-fungsi yang telah dibahas di modul 7 (JENI 4).

3. Percobaan

Percobaan 1: Membuat Class MidletTiledLayer

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

public class MidletTiledLayer extends MIDlet implements CommandListener {
    private Display display;
    private Command cmExit;
    private TiledLayerCanvas tlCanvas;

    public MidletLayerManager() {
        display = Display.getDisplay(this);
        cmExit = new Command("Keluar", Command.EXIT, 1);
        try {
            tlCanvas = new TiledLayerCanvas();
            tlCanvas.addCommand(cmExit);
            tlCanvas.setCommandListener(this);
        }
    }
}
```

Modul Praktikum TiledLayer

```
        tlCanvas.start();
    } catch (Exception ex) {
        System.out.println(ex);
    }
}

public void startApp() {
    display.setCurrent(tlCanvas);
}

public void pauseApp() { }

public void destroyApp(boolean unconditional) { }

public void exit() {
    destroyApp(true);
    notifyDestroyed();
}

public void commandAction(Command c, Displayable s) {
    if(c == cmExit) {
        exit();
    }
}
};

public void commandAction(Command c, Displayable s) {
    if(c == cmExit) {
        exit();
    }
}
};
```

Modul Praktikum TiledLayer

Percobaan 2 : Membuat Class TiledLayerCanvas

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

public class TiledLayerCanvas extends GameCanvas implements Runnable {
    private long delay;
    private int width;
    private int height;
    private LayerManager layerManager;
    // TiledLayer
    private TiledLayer hasilBackground;

    public TiledLayerCanvas() throws Exception {
        super(true);
        width = getWidth();
        height = getHeight();
        delay = 20;

        hasilBackground = initBackground();
        layerManager = new LayerManager();
        layerManager.append(hasilBackground);
    }

    public void start() {
        Thread t = new Thread(this);
        t.start();
    }

    public void run() {
        Graphics g = getGraphics();
        drawScreen(g);
        try { Thread.sleep(delay); }
        catch (InterruptedException ie) { }
    }

    private void drawScreen(Graphics g) {
        g.setColor(255, 255, 255);
        g.fillRect(0, 0, getWidth(), getHeight());
        g.setColor(0, 0, 255);
        layerManager.paint(g, 5, 5);
    }
}
```

Modul Praktikum TiledLayer

```
flushGraphics();
}

private TiledLayer initBackground() throws Exception {
    Image pecahanGambar = Image.createImage("/tangga.png");
    TiledLayer pecahanLayer = new TiledLayer(10,10, pecahanGambar,32,32);

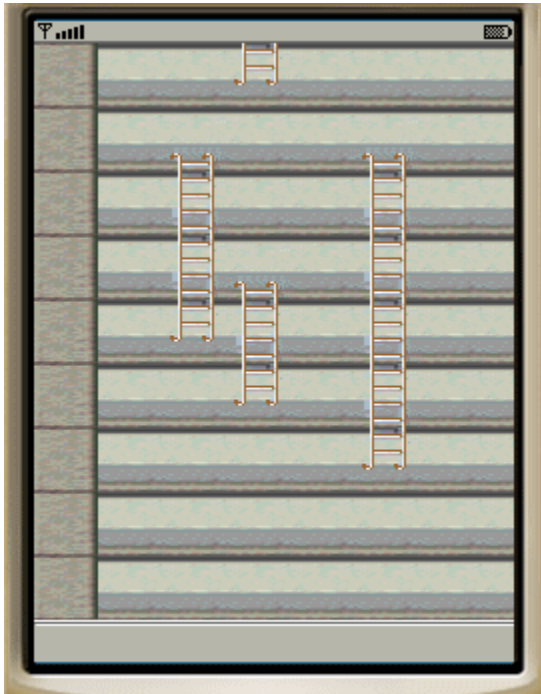
    // memanipulasi pecahan gambar
    int[] map = {
        5, 1, 1, 4, 1, 1, 1, 1, 1, 6,
        5, 1, 3, 1, 1, 3, 1, 1, 1, 6,
        5, 1, 2, 1, 1, 2, 1, 1, 1, 6,
        5, 1, 2, 3, 1, 2, 1, 1, 1, 6,
        5, 1, 4, 2, 1, 2, 1, 1, 1, 6,
        5, 1, 1, 4, 1, 2, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 4, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 6
    };

    // menyusun supaya pecahan gambar sesuai dengan map
    for (int i=0; i < map.length; i++) {
        int column = i % 10;
        int row = (i - column) / 10;
        pecahanLayer.setCell(column,row,map[i]);
    }

    return pecahanLayer;
}
};
```

Modul Praktikum TiledLayer

Hasil :



Modul Praktikum TiledLayer

Percobaan 3 : Membuat Class MidletAnimasiTiledLayer

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

public class MidletAnimasiTiledLayer extends MIDlet implements
CommandListener {
    private Display display;
    private Command cmExit;
    private AnimasiTiledLayerCanvas lmCanvas;

    public MidletAnimasiTiledLayer() {
        display = Display.getDisplay(this);
        cmExit = new Command("Keluar", Command.EXIT, 1);

        try {
            atlCanvas = new LayerManagerCanvas();
            atlCanvas.addCommand(cmExit);
            atlCanvas.setCommandListener(this);
            atlCanvas.start();
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    public void startApp() {
        display.setCurrent(atlCanvas);
    }

    public void pauseApp() { }

    public void destroyApp(boolean unconditional) { }

    public void exit() {
        destroyApp(true);
        notifyDestroyed();
    }

    public void commandAction(Command c, Displayable s) {
```

Modul Praktikum TiledLayer

```
        if(c == cmExit) {  
            exit();  
        }  
    }  
};
```

Percobaan 4 : Membuat Class AnimasiTiledLayerCanvas

```
import javax.microedition.lcdui.*;  
import javax.microedition.lcdui.game.*;  
  
public class AnimasiTiledLayerCanvas extends GameCanvas implements Runnable {  
    private boolean gerak;  
    private long delay;  
    private int width;  
    private int height;  
    private int animasi;  
    private boolean pilihTile;  
    private LayerManager layerManager;  
    private TiledLayer hasilBackground;  
  
    public AnimasiTiledLayerCanvas() throws Exception {  
        super(true);  
        width = getWidth();  
        height = getHeight();  
        delay = 500;  
  
        hasilBackground = rancangBackground();  
        layerManager = new LayerManager();  
        layerManager.append(hasilBackground);  
    }  
}
```

Modul Praktikum TiledLayer

```
public void start() {
    gerak = true;
    Thread t = new Thread(this);
    t.start();
}

public void stop() { gerak = false; }

public void run() {
    Graphics g = getGraphics();
    while (gerak == true) {
        drawScreen(g);
        try { Thread.sleep(delay); }
        catch (InterruptedException ie) { }
    }
}

private void drawScreen(Graphics g) {
    g.setColor(255, 255, 255);
    g.fillRect(0, 0, getWidth(), getHeight());
    g.setColor(0, 0, 255);
    // Menentukan Pecahan untuk ditampilkan
    if (pilihTile) {
        hasilBackground.setAnimatedTile(animasi,3);
    } else {
        hasilBackground.setAnimatedTile(animasi,4);
    }
    // menjadikan pilihTitle di set agar false supaya menjalankan
    (animasi,4)
    pilihTile = !pilihTile;
    layerManager.paint(g,5,5);
}
```

Modul Praktikum TiledLayer

```
flushGraphics();
}

private TiledLayer initBackground() throws Exception {
    Image pecahanGambar = Image.createImage("/tangga.png");
    TiledLayer pecahanLayer = new TiledLayer(10,10, pecahanGambar,32,32);

    // memanipulasi pecahan gambar
    int[] map = {
        5, 1, 1, 4, 1, 1, 1, 1, 1, 6,
        5, 1, 3, 1, 1, 3, 1, 1, 1, 6,
        5, 1, 2, 1, 1, 2, 1, 1, 1, 6,
        5, 1, 2, 3, 1, 2, 1, 1, 1, 6,
        5, 1, 4, 2, 1, 2, 1, 1, 1, 6,

        5, 1, 1, 4, 1, 2, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 4, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 6,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 6
    };

    // menyusun supaya pecahan gambar sesuai dengan map
    for (int i=0; i < map.length; i++) {
        int column = i % 10;
        int row = (i - column) / 10;
        pecahanLayer.setCell(column,row,map[i]);
    }

    // membuat animasiTile
    animasi = pecahanLayer.createAnimatedTile(5);
    // Menentukan koordinat Sel gambar dengan index
```

Modul Praktikum TiledLayer

```
    pecahanLayer.setCell(1,1,animasi);  
    return pecahanLayer;  
}  
};
```

Hasil :

Animasi antara potongan atas pada tangga
dengan potongan bawah pada tangga

