

Modul Praktikum SPRITE

1. Tujuan

Setelah mempelajari bab ini, siswa diharapkan dapat:

- Memahami dan menggunakan *class sprite*
- Menggambar dan menyusun *sprite*
- Manipulasi graphics dengan menggunakan method yang ada pada *class sprite*
- Dapat membuat animasi dengan *class sprite*.

2. Latar Belakang

Salah satu faktor yang membuat keberhasilan game adalah grafis yang unggul. Sebagian besar objek pada sebuah game dikategorikan sebagai grafis khusus yang disebut sprites. Sebuah sprite dapat berupa *bullet*, *monster*, karakter utama, musuh, kekuatan spesial, kunci dan pintu. Pada umumnya *Sprite* adalah grafis animasi. Grafis animasi dibuat dari *sprite* yang sama namun berbeda penampakkannya. Kumpulan sprite biasanya mengacu sebagai sebuah kumpulan *frame*. *Frame* ini dapat dibuat secara terurut ataupun tidak terurut. Umumnya *sprites* ditampilkan secara terurut untuk memudahkan pengkodean.

3. Percobaan

Percobaan 1: Membuat ContohSprite1

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

public class ContohSprite1 extends MIDlet {
    private Display display;
    public void startApp() {
        try {
            display = Display.getDisplay(this);
```

Modul Praktikum SPRITE

```
// Memanggil class ContohGameCanvas dengan membuat obyek
gameCanvas

ContohGameCanvas gameCanvas = new ContohGameCanvas();
gameCanvas.start();
display.setCurrent(gameCanvas);
} catch (Exception ex) {System.out.println(ex); }
}

public Display getDisplay() {
    return display;
}

public void pauseApp() {}
public void destroyApp(boolean unconditional) { }
public void exit() {
    destroyApp(false);
    notifyDestroyed();
}

class ContohGameCanvas extends GameCanvas implements Runnable {
    private boolean isPlay; // Game berjalan berulang-ulang ketikan
isPlay bernilai true
    private long delay; // memberikan nilai thread secara konsisten
    private int currentX, currentY; // Membuat posisi awal 'X' dan 'Y'
    private int width; // untuk lebar layar/screen
    private int height; // untuk lebar layar/screen
    private Image image, imageTemp; //Membuat obyek Image
    private Sprite sprite; //Membuat sprite transparan
    private Sprite nonTransparentSprite; //Membuat sprite tidak transparan
    // Constructor dan inialisasi
    public ContohGameCanvas() throws Exception {
        super(true);
        width = getWidth();
        height = getHeight();
        currentX = width / 2;
        currentY = height / 2;
```

Modul Praktikum SPRITE

```
        delay = 20;
        try{
            // Load Images untuk Sprites
            image = Image.createImage("/transparent.png");
            imageTemp = Image.createImage("/nontransparent.png");
        }catch (Exception ioex) { System.out.println(ioex); }
        sprite = new Sprite (image,32,32);
        nonTransparentSprite = new Sprite (imageTemp,32,32);
    }

    // membuat otomatis start thread untuk game yang berulang-ulang(loop)
    public void start() {
        isPlay = true;
        Thread t = new Thread(this);
        t.start();
    }

    public void stop() { isPlay = false; }
    // Main Game Loop
    public void run() {
        Graphics g = getGraphics();
        while (isPlay == true) {
            input();
            drawScreen(g);
            try { Thread.sleep(delay); }
            catch (InterruptedException ie) {}
        }
    }

    // Method untuk handle inputan dari keypad
    private void input() {
        int keyStates = getKeyStates();
        //sprite.setFrame(0);

        // Ke Kiri
```

Modul Praktikum SPRITE

```
        if ((keyStates == 4)) { // & LEFT_PRESSED) != 0) {
            currentX = currentX - 1;
            sprite.setFrame(1);
        }
        // Ke Kanan
        else if ((keyStates == 32)) { // & RIGHT_PRESSED) != 0 )
            if ( currentX + 5 < width) {
                currentX = currentX + 1;
                sprite.setFrame(3);
            }
            // Ke Atas
            if ((keyStates == 2)) { // { UP_PRESSED) != 0) {
                currentY = currentY - 1;
                sprite.setFrame(2);
            }
            // Ke Bawah
            else if ((keyStates == 64)) { // & DOWN_PRESSED) != 0)
                if ( currentY + 10 < height) {
                    currentY = currentY + 1;
                    sprite.setFrame(4);
                }
            }
        }

        // Method untuk Display Graphics
        private void drawScreen(Graphics g) {
            // g.setColor(0xffffffff);
            g.setColor(228, 77, 3);
            g.fillRect(0, 0, getWidth(), getHeight());
            g.setColor(0x0000ff);
            // display untuk sprites
            sprite.setPosition(currentX, currentY);
            sprite.paint(g);
            nonTransparentSprite.paint(g);
            flushGraphics();
        }
    }
```

Modul Praktikum SPRITE

```
}  
  
}  
  
}
```

Percobaan 2 : Membuat ContohSprite2

```
import javax.microedition.midlet.*;  
import javax.microedition.lcdui.*;  
import javax.microedition.lcdui.game.*;  
  
public class ContohSprite2 extends MIDlet {  
    private Display display;  
    public void startApp() {  
        try {  
            display = Display.getDisplay(this);  
            // Memanggil class ContohGameCanvas dengan membuat obyek gameCanvas  
            ContohGameCanvas gameCanvas = new ContohGameCanvas();  
            gameCanvas.start();  
            display.setCurrent(gameCanvas);  
        } catch (Exception ex) {System.out.println(ex); }  
    }  
  
    public Display getDisplay() { return display; }  
    public void pauseApp() {}  
    public void destroyApp(boolean unconditional) { }  
    public void exit() {  
        destroyApp(false);  
        notifyDestroyed();  
    }  
  
    class ContohGameCanvas extends GameCanvas implements Runnable {  
        private boolean isPlay; // Game berjalan berulang-ulang ketikan  
isPlay bernilai true
```

Modul Praktikum SPRITE

```
private long delay;          // memberikan nilai thread secara konsisten
private int currentX, currentY; // Membuat posisi awal 'X' dan 'Y'
private int width;           // untuk lebar layar/screen
private int height;          // untuk lebar layar/screen
private Image image, imageTemp; //Membuat obyek Image
private Sprite sprite;       //Membuat sprite transparan
private Sprite nonTransparentSprite; //Membuat sprite tidak transparan

// Constructor dan inialisasi
public ContohGameCanvas() throws Exception {
    super(true);
    width = getWidth();
    height = getHeight();
    currentX = width / 2;
    currentY = height / 2;
    delay = 20;
    try{
        // Load Images untuk Sprites
        image = Image.createImage("/panah.png");
        imageTemp = Image.createImage("/non_trans_panah.jpg");
    }catch (Exception ioex) { System.out.println(ioex); }
    sprite = new Sprite (image,20,20);
    nonTransparentSprite = new Sprite (imageTemp,20,20);
    }

    // membuat otomatis start thread untuk game yang berulang-
ulang(loop)
    public void start() {
        isPlay = true;
        Thread t = new Thread(this);
        t.start();
    }

    public void stop() { isPlay = false; }
```

Modul Praktikum SPRITE

```
// Main Game Loop
public void run() {
    Graphics g = getGraphics();
    while (isPlay == true) {
        input();
        drawScreen(g);
        try { Thread.sleep(delay); }
        catch (InterruptedException ie) {}
    }
}

// Method untuk menghandle inputan dari keypad
private void input() {
    int keyStates = getKeyStates();
    //sprite.setFrame(0);

    // Ke Kiri
    if ((keyStates & LEFT_PRESSED) != 0) {
        currentX = currentX - 1;
        sprite.setFrame(3);
    }
    // Ke Kanan
    else if ((keyStates & RIGHT_PRESSED) != 0 )
        if ( currentX + 5 < width) {
            currentX = currentX + 1;
            sprite.setFrame(2);
        }
    // Ke Atas
    if ((keyStates & UP_PRESSED) != 0) {
        currentY = currentY - 1;
        sprite.setFrame(1);
    }
    // Ke Bawah
```

Modul Praktikum SPRITE

```
        else if ((keyStates & DOWN_PRESSED) !=0)
            if ( currentY + 10 < height) {
                currentY = currentY + 1;
                sprite.setFrame(0);
            }
    }

    // Method untuk Display Graphics
    private void drawScreen(Graphics g) {
        //g.setColor(0xffffffff);
        g.setColor(228,77,3);
        g.fillRect(0, 0, getWidth(), getHeight());
        g.setColor(0x0000ff);
        // display untuk sprites
        sprite.setPosition(currentX,currentY);
        sprite.paint(g);
        nonTransparentSprite.paint(g);

        flushGraphics();
    }
}
```