

1. Tujuan

- Memahami event handling level rendah dalam MIDP
- Menggambar dan menampilkan teks, gambar, garis, kotak, dan sudut
- Menentukan warna, huruf, dan coretan untuk operasi menggambar
- Memahami dan menggunakan class Canvas dan Graphic
- Menggambar grafik berskala

2. Latar Belakang

DPada bab sebelumnya, kita telah membahas tentang bagaimana cara membuat user interface level tinggi seperti list, form, dan field input. Mereka bersifat user interface level tinggi dan programmer tidak perlu khawatir tentang menggambar pixel layar atau mengatur posisi teks pada layar. Semua program telah menetapkan jenis komponen dan label elemen. Sistem tersebut akan menangani gambar pada layar, scrolling dan layout.

Pada bab ini, kita akan berhadapan langsung dengan layar. Kita akan mempelajari class Canvas, dimana akan menjadi pendukung dari proses menggambar kita. Kita juga akan menyelidiki ke dalam class Graphic, dimana memiliki metode untuk menggambar garis, kotak, sudut, dan teks. Kita juga akan membahas huruf, warna dan gambar.

3. Percobaan

Percobaan 1: Menggambar Teks

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

public class HelloJENI extends MIDlet {
    private Display display;
    HelloCanvas1 canvas;

    public void startApp() {
        if(display == null){
            canvas = new HelloCanvas1(this);
            display = Display.getDisplay(this);
        }
        display.setCurrent(canvas);
    }
}
```



```
public void pauseApp() {
}

public void destroyApp(boolean unconditional) {
}
protected void Quit(){
    destroyApp(true);
    notifyDestroyed();
}
}

class HelloCanvas1 extends Canvas implements CommandListener {
    private Command exitCommand = new Command("Exit",Command.EXIT,0);
    private HelloJENI jeni;

    public HelloCanvas1(HelloJENI jeni){
        this.jeni = jeni;
        addCommand(exitCommand);
        setCommandListener(this);
    }
    protected void paint (Graphics g){
        g.setColor(255,255,255);
        g.fillRect(0,0,getWidth(),getHeight());
        g.setColor(255,0,0);
        g.drawString("JENI",getWidth()/2,getHeight()/2,Graphics.TOP |
Graphics.HCENTER );
        g.setColor(0,0,255);
        Font font = g.getFont();
        g.drawString("JAVA EDUCATION NETWORK
INDONESIA",getWidth()/2,getHeight()/2+font.getHeight() ,Graphics.TOP |
Graphics.HCENTER);
    }
    public void commandAction(Command c, Displayable d){
        if(c == exitCommand){
            jeni.Quit();
        }
    }
}
}
```



>>> Java Education Network Indonesia

Hasil :





Percobaan 2 : Penggambaran Garis dengan Warna

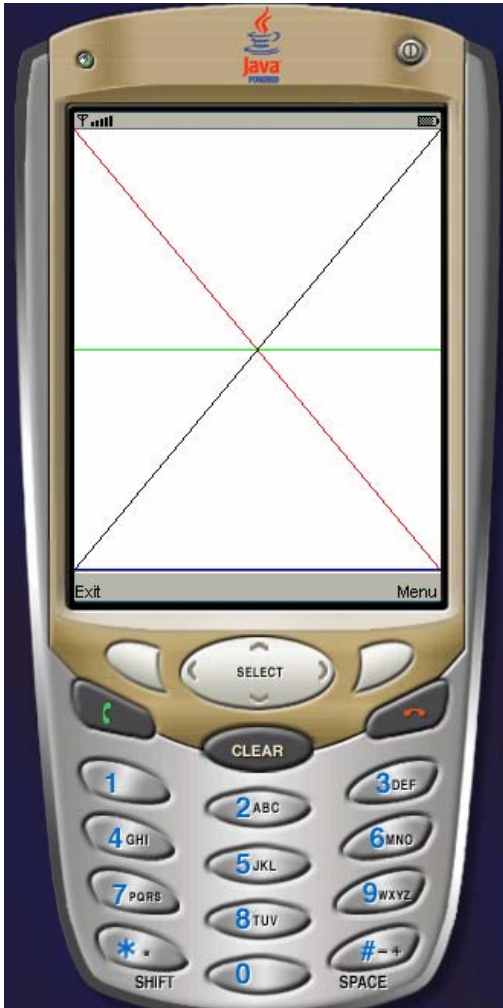
```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class GarisWarna extends MIDlet {
    private Display display;
    HelloCanvas2 canvas;
    public void startApp() {
        if (display == null){
            canvas = new HelloCanvas2(this);
            display = Display.getDisplay(this);
        }
        display.setCurrent(canvas);
    }
    public void pauseApp() {
    }
    public void destroyApp(boolean unconditional) {
    }
    protected void Quit(){
        destroyApp(true);
        notifyDestroyed();
    }
}

class HelloCanvas2 extends Canvas implements CommandListener {
    private Command exitCommand = new Command("Exit",Command.EXIT,0);
    private GarisWarna line;
    public HelloCanvas2(GarisWarna line){
        this.line = line;
        addCommand(exitCommand);
        setCommandListener(this);
    }
    protected void paint(Graphics g){
        g.setColor(255,255,255);
        g.fillRect(0,0,getHeight(),getWidth());
        g.setColor(255,0,0);
        g.drawLine(0,0,getWidth()-1,getHeight()-1);
        g.setColor(0,255,0);
        g.drawLine(0,getHeight()/2,getWidth()-1,getHeight()/2);
        g.setColor(0,0,255);
        g.drawLine(0,getHeight()-1,getWidth()-1,getHeight()-1);
        g.setColor(0,0,0);
        g.drawLine(0,getHeight()-1,getWidth()-1,0);
    }
    public void commandAction(Command c, Displayable d){
        if(c == exitCommand)
        {
            line.Quit();
        }
    }
}
}
```



>>> Java Education Network Indonesia

Hasil :





Percobaan 3 : Menggambar Segi Empat

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class Kotak extends MIDlet {
    private Display display;
    HelloCanvas3 canvas;
    public void startApp() {
        if (display == null){
            canvas = new HelloCanvas3(this);
            display = Display.getDisplay(this);
        }
        display.setCurrent(canvas);
    }
    public void pauseApp() {
    }
    public void destroyApp(boolean unconditional) {
    }
    protected void Quit(){
        destroyApp(true);
        notifyDestroyed();
    }
}
class HelloCanvas3 extends Canvas implements CommandListener {
    private Command exitCommand = new Command("Exit",Command.EXIT,7);
    private Kotak segiempat;
    public HelloCanvas3(Kotak segiempat){
        this.segiempat = segiempat;
        addCommand(exitCommand);
        setCommandListener(this);
    }
    protected void paint (Graphics g){
        g.setColor(255,255,255);
        g.fillRect(0,0,getWidth(),getHeight());
        //menggambar kotak bewarna kuning
        g.setColor(0,0,0);
        g.drawRect(8,8,64,32);
        g.setColor(255,255,0);
        g.fillRect(8,8,64,32);
        //menggambar berbagai macam jenis kotak
        g.setColor(0,0,0);
        g.drawRect(8,50,88,44);
        g.drawRoundRect(108,50,88,44,18,18);
        g.fillRect(8,120,88,44);
        g.fillRoundRect(108,120,88,44,18,18);
    }
}
```

```
}  
public void commandAction(Command c, Displayable d){  
    if (c == exitCommand){  
        segiempat.Quit();  
    }  
}  
}
```

Hasil :





Percobaan 4 : Menggambar Sudut

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class Sudut extends MIDlet {
    private Display display;
    HelloCanvas4 canvas;
    public void startApp() {
        if (display == null){
            canvas = new HelloCanvas4(this);
            display = Display.getDisplay(this);
        }
        display.setCurrent(canvas);
    }
    public void pauseApp() {
    }
    public void destroyApp(boolean unconditional) {
    }
    protected void Quit(){
        destroyApp(true);
        notifyDestroyed();
    }
}
class HelloCanvas4 extends Canvas implements CommandListener {
    private Command exitCommand = new Command("Exit",Command.EXIT,1);
    private Sudut angle;
    public HelloCanvas4(Sudut angle){
        this.angle = angle;
        addCommand(exitCommand);
        setCommandListener(this);
    }
    protected void paint (Graphics g){
        g.setColor(255,255,255);
        g.fillRect(0,0,getWidth(),getHeight());
        g.setColor(255,0,0);
        g.drawArc(18,18,50,50,0,360);
        g.setColor(0,255,0);
        g.drawArc(40,40,100,120,0,180);
        g.setColor(0,0,255);
        g.fillArc(100,200,80,100,0,90);
    }
    public void commandAction(Command c, Displayable d){
        if (c == exitCommand){
            angle.Quit();
        }
    }
}
}
```




>>> Java Education Network Indonesia

Hasil :





Percobaan 5: Menulis Hello World dengan Canvas

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

public class HelloCanvasMIDlet extends MIDlet {
    private Display display;
    HelloCanvas canvas;
    Command exitCommand = new Command("Exit",Command.EXIT,0);

    public void startApp() {
        if (display == null){
            canvas = new HelloCanvas(this,"Hello World");
            display = Display.getDisplay(this);
        }
        display.setCurrent(canvas);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
    protected void Quit(){
        destroyApp(true);
        notifyDestroyed();
    }
}

class HelloCanvas extends Canvas implements CommandListener {
    private Command exitCommand = new Command("Exit",Command.EXIT,0);
    private Command menuCommand = new Command("Menu",Command.OK,0);
    private HelloCanvasMIDlet midlet;
    private String text;

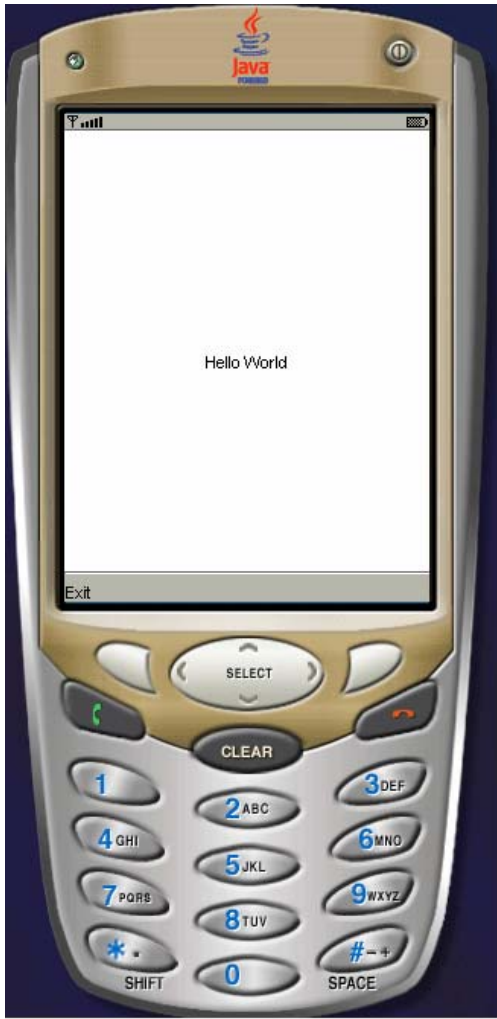
    public HelloCanvas(HelloCanvasMIDlet midlet,String text){
        this.midlet = midlet;
        this.text = text;
        addCommand(exitCommand);
        setCommandListener(this);
    }

    protected void paint(Graphics g) {
        g.setColor(255,255,255);
        g.fillRect(0,0,getWidth(),getHeight());
        g.setColor(0,0,0);
        g.drawString(text,getWidth()/2,getHeight()/2,Graphics.TOP | Graphics.HCENTER);
    }
}
```



```
public void commandAction(Command c, Displayable d){  
    if(c == exitCommand){  
        midlet.Quit();  
    }  
}
```

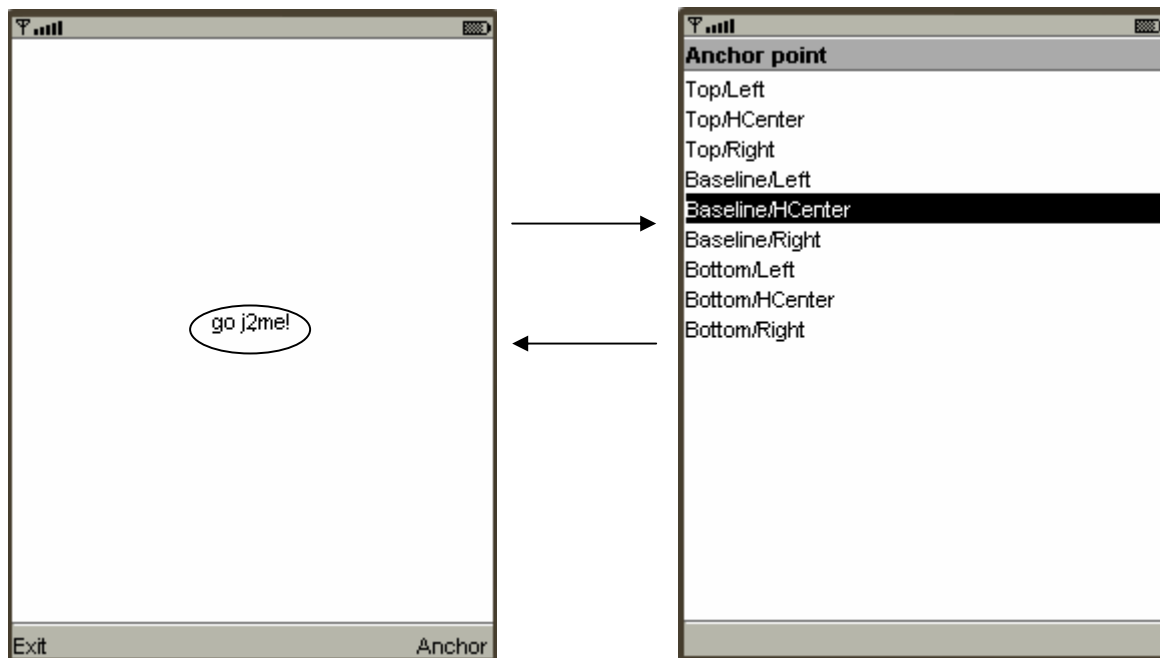
Hasil :



4. Latihan

Test Anchor Point

Buatlah sebuah demonstrasi sederhana mengenai Anchor Point pada J2ME. Ada dua interface yang mendukung program ini yaitu String "go j2me" yang digambarkan pada sebuah Canvas dan sebuah List mengenai semua titik anchor point yang didukung oleh MIDP. Sebagai default, Anda akan mendapatkan String tersebut diikuti dengan sebuah . (dot/titik) sebagai penanda anchor point dengan posisi BASELINE|HCENTER. Pada saat Anda memutuskan untuk mengubah posisi anchor point dari text tersebut, maka Anda akan mengklik command Anchor, kemudian memilih dari salah satu list letak titik anchor point yang Anda inginkan, sehingga text pada interface pertama tersebut akan digambarkan berbeda sesuai dengan titik anchor point yang telah Anda pilih. (Perhatikan gambar berikut ini)





Jawaban:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class Text extends MIDlet{
    private Display display;
    private TextCanvas canvas;
    private AnchorPtList anchorPt;
    private int anchorPoint = Graphics.BASELINE | Graphics.HCENTER;
    public Text(){
        display = Display.getDisplay(this);
        canvas = new TextCanvas(this);
        anchorPt = new AnchorPtList("Anchor point", List.IMPLICIT, this);
    }
    protected void startApp() {
        showCanvas();
    }
    protected void pauseApp() { }
    protected void destroyApp( boolean unconditional ) { }
    public void showCanvas() {
        display.setCurrent(canvas);
    }
    public void showList(){
        display.setCurrent(anchorPt);
    }
    public int getAnchorPoint(){
        return anchorPoint;
    }
    public void setAnchorPoint(int anchorPoint){
        this.anchorPoint = anchorPoint;
    }
    public void exitMIDlet() {
        destroyApp(true);
        notifyDestroyed();
    }
}

class TextCanvas extends Canvas implements CommandListener{
    private Command cmExit;
    private Command cmGetAnchorPt;
    private Text midlet;
    public TextCanvas(Text midlet) {
        this.midlet = midlet;
        cmExit = new Command("Exit", Command.EXIT, 1);
        cmGetAnchorPt = new Command("Anchor", Command.SCREEN, 2);
        addCommand(cmExit);
        addCommand(cmGetAnchorPt);
        setCommandListener(this);
    }
}
```

```
}
protected void paint(Graphics g){
    int xcenter = getWidth() / 2,
        ycenter = getHeight() / 2;
        g.setColor(255, 255, 255);
    g.fillRect(0, 0, getWidth(), getHeight());
    g.setColor(0, 0, 0);
    g.setFont(Font.getFont(Font.FACE_SYSTEM, Font.STYLE_PLAIN, Font.SIZE_MEDIUM));
    g.drawLine(xcenter, ycenter, xcenter, ycenter);
    g.drawString("Test Anchor Point", xcenter, ycenter, midlet.getAnchorPoint());
}

public void commandAction(Command c, Displayable d) {
    if (c == cmExit)
        midlet.exitMIDlet();
    else if (c == cmGetAnchorPt){
        // panggil method untuk menampilkan list
        midlet.showList();
    }
}

class AnchorPtList extends List implements CommandListener{
    private Text midlet;
    public AnchorPtList(String title, int listType, Text midlet) {
        super(title, listType);
        this.midlet = midlet;
        append("Top/Left", null);
        append("Top/HCenter", null);
        append("Top/Right", null);
        append("Baseline/Left", null);
        append("Baseline/HCenter", null);
        append("Baseline/Right", null);
        append("Bottom/Left", null);
        append("Bottom/HCenter", null);
        append("Bottom/Right", null);
        setCommandListener(this);
    }
    public void commandAction(Command c, Displayable s) {
        switch (getSelectedIndex()) {
            case 0:
                midlet.setAnchorPoint(Graphics.TOP | Graphics.LEFT);
                break;
            case 1:
                midlet.setAnchorPoint(Graphics.TOP | Graphics.HCENTER);
                break;
            case 2:
                midlet.setAnchorPoint(Graphics.TOP | Graphics.RIGHT);
                break;
            case 3:
```



```
        midlet.setAnchorPoint(Graphics.BASELINE | Graphics.LEFT);
        break;
    case 4:
        midlet.setAnchorPoint(Graphics.BASELINE | Graphics.HCENTER);
        break;
    case 5:
        midlet.setAnchorPoint(Graphics.BASELINE | Graphics.RIGHT);
        break;
    case 6:
        midlet.setAnchorPoint(Graphics.BOTTOM | Graphics.LEFT);
        break;
    case 7:
        midlet.setAnchorPoint(Graphics.BOTTOM | Graphics.HCENTER);
        break;
    case 8:
        midlet.setAnchorPoint(Graphics.BOTTOM | Graphics.RIGHT);
        break;
    default:
        midlet.setAnchorPoint(Graphics.BASELINE | Graphics.HCENTER);
    }
    midlet.showCanvas();
}
```