

```

CanvasMidlet.java
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;

public class CanvasMIDlet extends MIDlet implements CommandListener {
    private Display display;
    private ContohSplash cvsSplash;
    private ContohCanvas cvsLogin;

    public CanvasMIDlet() {
        display = Display.getDisplay(this);
    }

    public void tampilKanvasLogin()
    {
        cvsLogin = new ContohCanvas(this);
        display.setCurrent(cvsLogin);
    }

    public void tampilSplashScreen()
    {
        cvsSplash = new ContohSplash(this);
        display.setCurrent(cvsSplash);
        cvsSplash.setFullScreenMode(true);
        cvsSplash.start();
    }

    public void keluarAplikasi()
    {
        destroyApp(false); notifyDestroyed();
    }

    public void destroyApp(boolean arg0) {
    }

    public void pauseApp() {
    }

    public void startApp() {
        tampilSplashScreen();
    }

    public void commandAction(Command arg0, Displayable arg1) {
    }
}

```

ContohSplash.java

```
import java.util.Timer;
import java.util.TimerTask;
import javax.microedition.lcdui.Canvas;
import javax.microedition.lcdui.Graphics;
import javax.microedition.lcdui.Image;

public class ContohSplash extends Canvas {

    private CanvasMIDlet midlet;
    Image sp;

    private Timer timer;

    private TimerTask timerTask = new TimerTask(){
        public void run(){
            midlet.tampilKanvasLogin();
        }
    };

    public ContohSplash(CanvasMIDlet midlet) {
        this.midlet = midlet;

        setFullScreenMode(true);
        try{
            sp = Image.createImage("/splash.png");
        }
        catch(Exception e){
        }

        timer = new Timer();
    }

    protected void paint(Graphics g) {
        g.drawImage(sp,0,0,Graphics.TOP|Graphics.LEFT);
    }

    public void start(){
        timer.schedule(timerTask, 3000);
    }

    protected void keyPressed(int keyCode){
        midlet.tampilKanvasLogin();
        timer.cancel();
    }

}
```

```

ContohCanvas.java
import javax.microedition.lcdui.Canvas;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Graphics;
import javax.microedition.lcdui.Image;

public class ContohCanvas extends Canvas implements CommandListener {
    private CanvasMIDlet midlet;
    // menyimpan data string utk user dan pass
    String strUsername = new String("2005250025");
    String strPassword = new String();
    String strPasswordMask = new String();

    final int p_USER = 0; // cursor berada pada posisi username
    final int p_PASS = 1; // cursor berada pada posisi password
    final int p_LOGIN = 2; // cursor berada pada posisi button Login
    final int p_EXIT = 3; // cursor berada pada posisi button EXIT

    int posisi = p_USER; // posisi awal diset pada username

    int awalX = 30;
    int awalY = 150;

    private Command cmdHapus = new Command("Hapus", Command.BACK,0);
    private Image logo;

    public ContohCanvas(CanvasMIDlet midlet) {

        this.midlet = midlet;
        // create gambar logo
        try {
            logo = Image.createImage("/JavaDockLogo.png");
        }
        catch (Exception e){
        }

        addCommand(cmdHapus);
        setCommandListener(this);
    }

    protected void paint(Graphics g) {
        /* background color*/
        g.setColor(255,255,255);
        g.fillRect(0,0,g.getClipWidth(),g.getClipHeight());

        /* GAMBAR */
        g.drawImage(logo,40,50, Graphics.TOP|Graphics.LEFT);

        /* string */
        g.setColor(0,0,0);
        g.drawString("Username", 10 +awalX, 2 +awalY, Graphics.TOP|Graphics.LEFT);
        g.drawString("Password", 10 +awalX, 26+awalY, Graphics.TOP|Graphics.LEFT);

        g.drawString(":", 75 +awalX, 2+awalY, Graphics.TOP|Graphics.LEFT);
        g.drawString(":", 75 +awalX, 26+awalY, Graphics.TOP|Graphics.LEFT);

        /* kotak isi */
        g.setColor(0,200,255);
        g.fillRect(90 +awalX,2+awalY,90,13);
        g.fillRect(90 +awalX,26+awalY,50,13);

        /* text di dalam kotak isi */
        g.setColor(255,255,255);
        if (strUsername.length() > 0)
            g.drawString(strUsername, 90 +awalX,2+awalY,Graphics.TOP|Graphics.LEFT);
        if (strPassword.length() > 0)
            g.drawString(strPasswordMask, 90 +awalX,26+awalY,Graphics.TOP|Graphics.LEFT);

        /* bingkai */
        if (posisi <= p_PASS){
            g.setColor(50,50,0);
            g.drawRect(awalX,(24 * posisi) +awalY,180,16);
            //g.drawRect(12,(24 * 0) +28,103,16);
            //g.drawRect(12,(24 * 1) +28,103,16);
        }

        /* button */
        if (posisi <= p_PASS) {
            g.setColor(100,50,0);
            g.fillRoundRect(70, 210, 80, 25, 15, 45);
            g.fillRoundRect(70, 210+30, 80, 25, 15, 45); // utk LOGIN // utk EXIT
        }

        else if (posisi == p_LOGIN ) {
            g.setColor(0,0,255);
            g.fillRoundRect(70, 210, 80, 25, 15, 45); // utk LOGIN
            g.setColor(100,50,0);
            g.fillRoundRect(70, 210+30, 80, 25, 15, 45); // utk EXIT
        }

        else if (posisi == p_EXIT){
            g.setColor(0,0,255);
            g.fillRoundRect(70, 210+30, 80, 25, 15, 45); // utk EXIT
            g.setColor(100,50,0);
        }
    }
}

```

```

        g.fillRect(70, 210, 80, 25, 15, 45); // utk LOGIN
    }
    /* tulisan pada button */
    g.setColor(255,255,255);
    g.drawString("L O G I N", 40 + awalX, 65+awalY, Graphics.TOP|Graphics.LEFT);
    g.drawString(" E X I T", 40 + awalX, 65+30+awalY, Graphics.TOP|Graphics.LEFT);
}

protected void keyPressed(int keyCode)
{
    if (keyCode >= 35)
    {
        if (posisi == p_USER)
        {
            if (strUsername.length() < 10)
            {
                char ch = (char) keyCode;
                strUsername = strUsername + ch;
                repaint();
            }
        }
        else if (posisi == p_PASS)
        {
            if (strPassword.length() < 4)
            {
                char ch = (char) keyCode;
                strPassword = strPassword + ch;
                strPasswordMask = strPasswordMask + '*';
                repaint();
            }
        }
    }
    else
    {
        int arrow = getGameAction(keyCode);
        if (arrow == UP)
        {
            if (posisi > p_USER)
            {
                posisi--;
                repaint();
            }
        }
        else if (arrow == DOWN)
        {
            if (posisi < p_EXIT)
            {
                posisi++;
                repaint();
            }
        }
        else if (arrow == FIRE)
        {
            if (posisi == p_LOGIN)
            {
                // misal : midlet.tampilFormInput(); dst
            }
            else if (posisi == p_EXIT)
            {
                midlet.keluarAplikasi();
            }
        }
    }
}

public void commandAction(Command c, Displayable d) {
    if (c == cmdHapus )
    {
        if (posisi == p_USER)
        {
            strUsername = strUsername.substring(0, strUsername.length()-1);
            repaint();
        }
        else if (posisi == p_PASS)
        {
            strPassword = strPassword.substring(0, strPassword.length()-1);
            strPasswordMask = strPasswordMask.substring(0, strPasswordMask.length()-1);
            repaint();
        }
    }
}
}
}
}

```

