

Pengembangan High dan Low Level User Interface

1.1. Tujuan:

Mahasiswa dapat membuat form dan menambahkan CustomItem, Textbox dan Gauge object untuk download game serta membuat game StarShipBattle 2004.

1.2. Latihan:

Adapun tujuan-tujuan dari latihan ini yaitu:

1. Untuk membuat CustomItem object yang memajang kata rahasia pada user
2. Untuk membuat class yang menempatkan kata rahasia pada form

Membuat form yang menghadiahi user dengan SecretWordItem class adalah simple CustomItem yang sebelumnya digunakan user dengan password rahasia. User harus mengetik password ke dalam field text dan jika sukses, user diperbolehkan menggerakkan menu utama aplikasi.

1.3. Persiapan:

Untuk menyiapkan latihan ini lakukan setting computer pada lokasi direktori: labdir = _____

Langkah 1— Membuat form dan menambahkan CustomItem, Textbox dan Gauge object untuk download game

Tugas 1 - Membuat CustomItem Object yang didisplay dalam SecretWord untuk user

Copy file pada lokasi direktori dari labdir\Lab4_Exercise2\src ke labdir\Lab5_Exercise1\src dan ikuti step ini:

1. Deklarasikan class baru, disebut SecretWordItem class, yang meluas pada CustomItem class yang didefinisikan di dalam javax.microedition.lcdui package.
2. Deklarasikan String secretWord property
Properti ini sebelumnya adalah word yang di display untuk user.
3. Deklarasikan metode berikut:

- `int getMinContentWidth()` – Kembali ke lebar minimum width didalam isi objek area `CustomItem`.
- `int getMinContentHeight()` – Kembali ke tinggi minimum didalam isi objek `CustomItem`.
- `int getPrefContentHeight(int height)` – Kembali ke objek `CustomItem` berisi tinggi minimum.
- `int getPrefContentWidth(int width)` – Kembali ke objek `CustomItem` berisi lebar minimum.
- `void paint(Graphics g,int w,int h)` – Melukis `secretWord` property pada layar didalam lukisan segiempat.
- `boolean checkWord(String word)` – Bandingkan parameter `word` dengan `secretWord` property
Jika words cocok, bernilai nilai `true` sebaliknya `false`.

Tugas 2 – Membuat class yang menempatkan Secret WordOnaForm

Setelah membuat objek `CustomItem` yang memajang `secret word` untuk user, maka harus membuat class lain yang menempatkan `secret word` pada form dan juga menyediakan fungsi yang memungkinkan user untuk memasukkan word dan mengujinya. Lengkapi step berikut :

1. Desain form yang mempunyai `TextField` pada `SecretWordItem` dan `command` object.

Ketika user menekan `command` object, bandingkan form string ke text yang memasukkan user dengan string ke text display oleh `SecretWordItem` object. Jika text string cocok, lalu dilanjutkan ke menu screen.

- Modifikasi klas `SplashScreen` sehingga yang mendaftarkan adalah `LoginForm` objek dan `MainMenu` objek
- Modifikasi `SplashScreen` objek sehingga yang didisplay ke `LoginForm` objek ketika key ditekan.
- Modifikasi metode `StarShipBattle startApp()` sedemikian sehingga yang membuat `LoginForm` objek dan membuat yang ada untuk `SplashScreen` objek.

Langkah 2— Membuat Game StarShipBattle 2004

Untuk menyiapkan latihan ini lakukan setting computer pada lokasi direktori: labdir = _____

Copy file pada lokasi direktori labdir\Lab5_Exercise1\src ke labdir\Lab5_Exercise2\src dan ikuti step ini:

- Definisikan class baru , disebut Battle, yang subclass dari GameCanvas class dan implementasi Runnable interface.
- Definisikan properties yang mengikuti Battle class berikut:
 - TiledLayer backgrnd – objek sebelum latar belakang image untuk game.
 - LayerManager lmgr – objek ini adalah objek yang mengatur display image.
 - Sprite ship – objek sebelum hero ship
 - Sprite enemy – objek sebelum enemy ship
 - Display deviceDisplay – objek berisi referensi untuk device display
 - Screen mainMenu – objek sebelum referensi untuk mengikuti user kembali ke Menu System.
 - Boolean running – variable kontrol digunakan untuk mengatur game processing Thread
- Definisi mengikuti metode di dalam Battle class:
 - public Battle(Display d, Screen m) – Ini adalah default constructor. Ketika objek Battle dibuat, kamu harus menginisialisasi semua grafik.
 - Public void initSprites – Lengkapi metode dengan tindakan berikut:
 - a. Loads the enemy dan hero images
 - b. Sets ships position didalam lapangan bermain.
 - c. Definisikan segi empat pada enemy ship
 - void stop() – metode ini disebut kejadian yang mengancam game membutuhkan untuk diberhentikan oleh setting variable berjalan ke false.
 - void start() – Lengkapi metode berikut ini:
 - a. Set running variable ke true
 - b. Buat ancaman baru
 - c. Starts the game loop
 - void run() – Ini adalah game looping yang harus selesai pada logika berikut :

- a. Periksa variable kendali dan apabila benar, lanjutkan eksekusi pada loop. Sebaliknya game over dan proses game berhenti.
- b. Query kestates. Jika user menekan tombol petunjuk kiri atau kanan lalu kapal juga bergerak ke kiri atau kanan.
- c. Otomatis bergerak ke kapal musuh sepanjang layar lima pixels screen five pixels
- d. Lihat kemungkinan benturan antar kapal.
- e. Lukis kapal pada display.

Isyarat: Pertimbangkan penempatan yang strategis [sebut/panggil/hubungi] kepada [itu] tidur metoda.

Isyarat: Pertimbangkan penempatan strategi panggilan pada sleep method.

metoda ini digunakan oleh pengulangan/jerat game untuk mengkonversi kapal [itu] untuk dibinasakan gambaran dan kemudian tidur untuk satu [yang] kedua. Setelah benang/ulir start lagi, gambaran dipugar dan digantikan pada startup posisi mereka.

- void destroyShips() – Metode ini digunakan oleh loop game yang dikeonversikan pada kapal untuk mengganti image dan lalu tertidur beberapa detik. Setelah ancaman start lagi, image dimusnahkan dan diganti dengan startup posisi mereka.
- Di dalam kelas MainMenu, definisi property yang sebelum kejadian pada kelas Battle.
- Ubah constructor pada MainMenu untuk membuat kejadian kelas Battle dan menyimpannya di dalam property yang sudah ditetapkan pada step 4.
- Ubah startGame pada MainMenu untuk memanggil Battle.start() dan menempatkan kelas Battle pada device display.

1.4. Penyelesaian Latihan

Gunakan penyelesaian untuk mengecek jawaban anda pada latihan dalam lab.

Bandingkan jawaban anda dengan contoh program.

Langkah 1— Membuat form dan menambahkan CustomItem, Textbox dan Gauge object untuk download game

Program 5-1 adalah contoh SecretWordItem.java

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

public class SecretWordItem extends CustomItem{

    private String secretWord = "Cubs";

    public SecretWordItem(){
        super("PassCode Prompt");
    }

    public int getMinContentWidth(){ return 100; }

    public int getMinContentHeight(){ return 20; }

    public int getPrefContentWidth(int width){
        return getMinContentWidth();
    }

    public int getPrefContentHeight(int height){
        return getMinContentHeight();
    }

    public void paint(Graphics g, int w, int h){

        g.setColor(0,0,255);
        g.fillRect(0,0, w -1, h-1);

        g.setColor(255,255,255);
        g.drawString(secretWord, 50,0, g.TOP|g.HCENTER);

    }
}
```

```

public boolean checkWord(String word){

    boolean result = false;

    if (secretWord.equals(word)){
        result = true;
    }

    return result;
}
}

```

Program 5-2 adalah contoh SplashScreen.java

```

import javax.microedition.lcdui.*;
import java.io.IOException;

public class SplashScreen extends Canvas{

    private Image splashImage;
    private LoginForm login;

    public SplashScreen(){
        loadSplashScreen();
    }

    public void loadSplashScreen(){
        try{
            splashImage = Image.createImage("/titlepage.png");
        }catch(IOException ioe){
            System.out.println("ERR:SplashScreen: Unable to load image");
        }
    }

    public void paint(Graphics g){

        g.fillRect(0,0,getWidth(), getHeight());
        g.drawImage(splashImage, 0,0, Graphics.TOP | Graphics.LEFT);
    }

    public void registerLogin(LoginForm lf){
        login = lf;
    }
}

```

```

        public void keyPressed(int keyCode){
            login.showLoginForm();
        }
    }
}

```

Program 5-3 adalah contoh LoginForm.java

```

import javax.microedition.lcdui.*;

public class LoginForm implements CommandListener{

    private SecretWordItem swi;
    private Form loginForm;
    private TextField userEntry;

    private Display deviceDisplay;
    private MainMenu mainMenu;

    private Command enter;

    public LoginForm(Display d ){
        deviceDisplay = d;

        loginForm = new Form("Login");
        swi = new SecretWordItem();
        userEntry = new TextField("Password", null, 25, TextField.ANY);
        enter = new Command("Enter", Command.OK, 1);

        loginForm.append(swi);
        loginForm.append(userEntry);
        loginForm.addCommand(enter);
        loginForm.setCommandListener(this);
    }
    public void registerMenu(MainMenu menu){
        mainMenu = menu;
    }

    public void showLoginForm(){
        deviceDisplay.setCurrent(loginForm);
    }
}

```

```

public void showMenu(){
    mainMenu.showMenu();
}

public void commandAction(Command c, Displayable d){
    if (swi.checkWord(userEntry.getString())){
        showMenu();
    }else{
        Alert err = new Alert("Invalid Password!", "Invalid Password\n Please try
again", null, AlertType.ERROR);
        err.setTimeout(Alert.FOREVER);
        deviceDisplay.setCurrent(err, loginForm);
    }
}
}

```

Program 5-4 adalah contoh MainMenu.java

```

import javax.microedition.lcdui.*;
import java.io.IOException;

public class MainMenu implements CommandListener{
    private List menu;
    private Display deviceDisplay;

    private Command startGame;
    private Command setOptions;
    private Command exitGame;

    private Gauge loadingGameGauge;
    private Alert loadingAlert;

    private StarShipBattle ssb;
    private UserOptions userOptions;    }
    public void registerMenu(MainMenu menu){
        mainMenu = menu;
    }

    public void showLoginForm(){
        deviceDisplay.setCurrent(loginForm);
    }
}

```

```

public MainMenu(StarShipBattle mid, Display d){
    deviceDisplay = d;

    menu = new List("Game Menu", Choice.IMPLICIT);
    startGame = new Command("Start Game", Command.OK, 1);
    setOptions = new Command("Set Options", Command.OK, 1);
    exitGame = new Command("Exit Application", Command.EXIT, 1);

    menu.addCommand(startGame);
    menu.addCommand(setOptions);
    menu.addCommand(exitGame);

    loadingGameGauge = new Gauge(null,
    false,Gauge.INDEFINITE,Gauge.CONTINUOUS_RUNNING);
    loadingGameGauge.setPreferredSize(-1,-1);

    loadingAlert = new Alert("Loading Game", "Loading StarShip Battle 2004,
    Please standby...", null, AlertType.INFO);

    menu.setCommandListener(this);

    ssb = mid;
    userOptions = new UserOptions(d);
    userOptions.registerMenu(menu);
}
public void commandAction(Command c, Displayable d){

    if(c == startGame){

        startGame();

    }else if(c == setOptions){

        setOptions();

    }else if(c == exitGame){

        exitGame();

    }
}
}

```

```

public void startGame(){
    loadingAlert.setImage(null);
    loadingAlert.setIndicator(loadingGameGauge);
    deviceDisplay.setCurrent(loadingAlert, menu);
}

public void setOptions(){
    userOptions.showUserOpt();
}

public void exitGame(){
    ssb.shutdown();
}

public void showMenu(){
    deviceDisplay.setCurrent(menu);
}
}

```

Program 5-5 adalah contoh StarShipBattle.java

```

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

public class StarShipBattle extends MIDlet{

    private final String profileVersion = "MIDP-2.0";
    private final String configVersion = "CLDC-1.0";

    public StarShipBattle(){
        String pv = getAppProperty("MicroEdition-Profile");
        String cv = getAppProperty("MicroEdition-Configuration");

        if((pv.equals(profileVersion)) && (cv.equals(configVersion))){
            System.out.println("Versions ok, proceed as needed");
            System.out.println("Loading the application...");
        }else{
            System.out.println("Versions incorrect...unable to proceed");
            System.out.println("Unloading the application...");
            shutdown();
        }
    }
}

```

```

public void startApp(){
    System.out.println("in the startApp()");

    Display d = Display.getDisplay(this);
    LoginForm lf = new LoginForm(d);
    MainMenu mm = new MainMenu(this, d);
    SplashScreen ss = new SplashScreen();
    lf.registerMenu(mm);
    ss.registerLogin(lf);
    d.setCurrent(ss);
}

public void pauseApp(){
    System.out.println("in the pauseApp()... ");
}

public void destroyApp(boolean unc){
    System.out.println("in the destroyApp()... ");
}

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

```

Program 5-6 adalah contoh UserOptions.java

```

import javax.microedition.lcdui.*;
import java.io.IOException;

public class UserOptions implements CommandListener{

    private Form inputForm;

    private TextField name;
    private TextField alias;

    private Display deviceDisplay;
    private String nameAlertText = "Your name has changed.";
    private String aliasAlertText = "Your alias has changed.";
    private String saveAlertText = "Data Saved!";
}

```

```

private Image okImage;
private Alert updateAlert;
private Displayable mainMenu;

private Command saveData;

public UserOptions(Display d){

    name = new TextField("UserName", null, 25, TextField.ANY);
    alias = new TextField("alias", null, 25, TextField.ANY);

    inputForm = new Form("User Information");
    inputForm.append(name);
    inputForm.append(alias);

    deviceDisplay = d;

    try{
        okImage = Image.createImage("/dukeok8.png");
    }catch(IOException io){
        System.out.println("ERR:UserOptions:Unable to load image");
    }

    updateAlert = new Alert("Data Changed", null,
okImage,AlertType.CONFIRMATION);
    updateAlert.setTimeout(Alert.FOREVER);
    //inputForm.setItemStateListener(this);

    saveData = new Command("Save", Command.OK, 1);
    inputForm.addCommand(saveData);
    inputForm.setCommandListener(this);

}
public void registerMenu(Displayable mm){
    mainMenu = mm;
}

public void commandAction(Command c, Displayable d){

    if (c == saveData){
        updateAlert.setString(saveAlertText);
        showAlertUpdate();
    }

}
}

```

```
/*
    public void itemStateChanged(Item i){
        if( i == name){
            updateAlert.setString(nameAlertText);
        }else{
            updateAlert.setString(aliasAlertText);
        }
        showAlertUpdate();
    }
*/
private void showAlertUpdate(){
    deviceDisplay.setCurrent(updateAlert, mainMenu);
}

public void showUserOpt(){
    deviceDisplay.setCurrent(inputForm);
}

}
```

Langkah 2— Membuat Game StarShipBattle 2004

Program 5-7 adalah contoh file Battle.java.

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

import java.io.*;

public class Battle extends GameCanvas implements Runnable{

private TiledLayer backgrnd;
private LayerManager lmgr;
private Sprite ship;
private Sprite enemy;

private Display deviceDisplay;
private MainMenu menu;

private boolean running = false;
public Battle(Display s, MainMenu m){
    super(true);
    deviceDisplay = s;
    menu = m;

    lmgr = new LayerManager();
    try{
        Image bkImage = Image.createImage("/background.png");

        backgrnd = new TiledLayer(7,8,bkImage,20,20);
        backgrnd.setPosition(20,12);

        int map[] = { 1,2,3,4,1,2,3,
                    2,4,3,1,4,3,1,
                    4,3,2,1,1,2,4,
                    2,3,4,1,2,3,1,
                    3,4,1,2,3,1,4,
                    2,4,3,1,4,3,1,
                    4,1,2,3,1,3,4,
                    2,3,1,4,2,2,1 };
        for (int i =0; i < map.length; i++){
            int col = i%7;
            int row = (i-col) /7;
            backgrnd.setCell(col,row,map[i]);
        }
    }
```

```

        }catch(IOException ioe){
            System.out.println("ERR:Loading Image");
        }

        initSprites();
        Imgr.append(backgrnd);
        Imgr.insert(ship,0);
        Imgr.insert(enemy,0);
        Graphics g = getGraphics();
        Imgr.paint(g,0,0);
        flushGraphics();
    }
    public void initSprites(){

        try{
            Image shipImage = Image.createImage("/starshipline.png");
            ship = new Sprite(shipImage, 30,30);
            ship.setFrame(1);
            ship.setPosition(70,130);

            Image enemyImage = Image.createImage("/alienb.png");

            enemy = new Sprite(enemyImage,30,30);
            enemy.setFrame(1);
            enemy.setPosition(70,15);

        }catch(IOException ioe){
            System.out.println("ERR: Generating Sprite");
        }
    }
    public void stop(){
        running = false;
    }

    public void start(){
        System.out.println("Starting the game thread");
        running = true;
        Thread t = new Thread(this);
        t.start();
    }
}

```

```

public void run(){
    System.out.println("In the game loop");
    Graphics g = getGraphics();
    int ks = 0;
    int currentX = 0;
    lmgr.paint(g,0,0);
    flushGraphics();

    while (running){
        ks = getKeyStates();
        currentX = ship.getX();
        ship.setFrame(1);
        if ((ks & LEFT_PRESSED) != 0){
            if((currentX-5) >20){
                ship.setFrame(0);
                ship.move(-5,0);
            }
        }
        else if((ks & RIGHT_PRESSED) !=0){
            if((currentX+5) <130){
                ship.setFrame(2);
                ship.move(5,0);
            }
        }
        else if((ks & GAME_A_PRESSED) != 0){
            stop();
        }

        if((enemy.getY()+ 5) <140){
            enemy.move(0,5);
            if(enemy.collidesWith(ship,true)){
                destroyShips();
            }
        }
        else{
            enemy.setPosition(70,15);
        }
        lmgr.paint(g,0,0);

        flushGraphics();

        try{
            Thread.sleep(100);
        }catch(InterruptedException ie){ }

    }
}

```

```

System.out.println("Leaving the game loop");

        menu.showMenu();
    }
private void destroyShips(){
    Graphics g = getGraphics();
    enemy.setFrame(3);
    ship.setFrame(4);
    lmgr.paint(g, 0,0);
    flushGraphics();
    try{
        Thread.sleep(1000);
    }catch(InterruptedException ie){ }
    enemy.setFrame(1);
    enemy.setPosition(70,15);
}
}

```

Program 5-8 adalah contoh file MainMenu.java.

```

import javax.microedition.lcdui.*;
import java.io.IOException;

public class MainMenu implements CommandListener{
    private List menu;
    private Display deviceDisplay;

    private Command startGame;
    private Command setOptions;
    private Command exitGame;

    private Gauge loadingGameGauge;
    private Alert loadingAlert;

    private StarShipBattle ssb;
    private UserOptions userOptions;

    private Battle gameEngine;

```

```

public MainMenu(StarShipBattle mid, Display d){
    deviceDisplay = d;

    menu = new List("Game Menu", Choice.IMPLICIT);
    startGame = new Command("Start Game", Command.OK, 1);
    setOptions = new Command("Set Options", Command.OK, 1);
    exitGame = new Command("Exit Application", Command.EXIT, 1);

    menu.addCommand(startGame);
    menu.addCommand(setOptions);
    menu.addCommand(exitGame);

    loadingGameGauge = new Gauge(null,
    false,Gauge.INDEFINITE,Gauge.CONTINUOUS_RUNNING);
    loadingGameGauge.setPreferredSize(-1,-1);

    loadingAlert = new Alert("Loading Game", "Loading StarShip Battle 2004,
    Please standby...", null, AlertType.INFO);

    menu.setCommandListener(this);

    ssb = mid;
    userOptions = new UserOptions(d);
    userOptions.registerMenu(menu);

    gameEngine = new Battle(d, this);
}
public void commandAction(Command c, Displayable d){

    if(c == startGame){

        startGame();

    }else if(c == setOptions){

        setOptions();

    }else if(c == exitGame){

        exitGame();

    }

}
}

```

```

public void startGame(){
    gameEngine.start();
    deviceDisplay.setCurrent(gameEngine);
}

public void setOptions(){
    userOptions.showUserOpt();
}

public void exitGame(){
    ssb.shutdown();
}

public void showMenu(){

    deviceDisplay.setCurrent(menu);
}
}

```

Program 5-9 adalah contoh file StarShipBattle.java.

```

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

public class StarShipBattle extends MIDlet{

    private final String profileVersion = "MIDP-2.0";
    private final String configVersion = "CLDC-1.0";

    public StarShipBattle(){
        String pv = getAppProperty("MicroEdition-Profile");
        String cv = getAppProperty("MicroEdition-Configuration");

        if((pv.equals(profileVersion)) && (cv.equals(configVersion))){
            System.out.println("Versions ok, proceed as needed");
            System.out.println("Loading the application...");
        }else{
            System.out.println("Versions incorrect...unable to proceed");
            System.out.println("Unloading the application...");
            shutdown();
        }
    }
}

```

```

public void startApp(){
    System.out.println("in the startApp()");

    Display d = Display.getDisplay(this);
    LoginForm lf = new LoginForm(d);
    MainMenu mm = new MainMenu(this, d);
    SplashScreen ss = new SplashScreen();
    lf.registerMenu(mm);
    ss.registerLogin(lf);
    d.setCurrent(ss);
}
public void pauseApp(){
    System.out.println("in the pauseApp()... ");
}

public void destroyApp(boolean unc){
    System.out.println("in the destroyApp()... ");
}

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

```

Program 5-10 adalah contoh file UserOptions.java.

```

import javax.microedition.lcdui.*;
import java.io.IOException;

public class UserOptions implements CommandListener{

    private Form inputForm;

    private TextField name;
    private TextField alias;

    private Display deviceDisplay;

    private String nameAlertText = "Your name has changed.";
    private String aliasAlertText = "Your alias has changed.";
    private String saveAlertText = "Data Saved!";
}

```

```

private Image okImage;
private Alert updateAlert;
private Displayable mainMenu;

private Command saveData;
public UserOptions(Display d){

    name = new TextField("UserName", null, 25, TextField.ANY);
    alias = new TextField("alias", null, 25, TextField.ANY);

    inputForm = new Form("User Information");
    inputForm.append(name);
    inputForm.append(alias);

    deviceDisplay = d;

    try{
        okImage = Image.createImage("/dukeok8.png");
    }catch(IOException io){
        System.out.println("ERR:UserOptions:Unable to load image");
    }

    updateAlert = new Alert("Data Changed", null,
        okImage,AlertType.CONFIRMATION);
    updateAlert.setTimeout(Alert.FOREVER);
    //inputForm.setItemStateListener(this);

    saveData = new Command("Save", Command.OK, 1);
    inputForm.addCommand(saveData);
    inputForm.setCommandListener(this);

}
public void registerMenu(Displayable mm){
    mainMenu = mm;
}

public void commandAction(Command c, Displayable d){

    if (c == saveData){
        updateAlert.setString(saveAlertText);
        showAlertUpdate();
    }

}
}

```

```

/*
    public void itemStateChanged(Item i){
        if( i == name){
            updateAlert.setString(nameAlertText);
        }else{
            updateAlert.setString(aliasAlertText);
        }
        showAlertUpdate();
    }
*/
private void showAlertUpdate(){
    deviceDisplay.setCurrent(updateAlert, mainMenu);
}

public void showUserOpt(){
    deviceDisplay.setCurrent(inputForm);
}
}

```

Program 5-11 adalah contoh file SplashScreen.java

```

import javax.microedition.lcdui.*;
import java.io.IOException;

public class SplashScreen extends Canvas{

    private Image splashImage;
    private LoginForm login;

    public SplashScreen(){
        loadSplashScreen();
    }

    public void loadSplashScreen(){
        try{
            splashImage = Image.createImage("/titlepage.png");
        }catch(IOException ioe){
            System.out.println("ERR:SplashScreen: Unable to load image");
        }
    }
}

```

```

public void paint(Graphics g){

    g.fillRect(0,0,getWidth(), getHeight());
    g.drawImage(splashImage, 0,0, Graphics.TOP | Graphics.LEFT);
}

public void registerLogin(LoginForm lf){
    login = lf;
}

public void keyPressed(int keyCode){
    login.showLoginForm();
}

}

```

Program 5-12 adalah contoh file LoginForm.java

```

import javax.microedition.lcdui.*;

public class LoginForm implements CommandListener{

    private SecretWordItem swi;
    private Form loginForm;
    private TextField userEntry;

    private Display deviceDisplay;
    private MainMenu mainMenu;

    private Command enter;

    public LoginForm(Display d ){
        deviceDisplay = d;

        loginForm = new Form("Login");
        swi = new SecretWordItem();
        userEntry = new TextField("Password", null, 25, TextField.ANY);
        enter = new Command("Enter", Command.OK, 1);

        loginForm.append(swi);
        loginForm.append(userEntry);
        loginForm.addCommand(enter);
        loginForm.setCommandListener(this);
    }
}

```

```

public void registerMenu(MainMenu menu){
    mainMenu = menu;
}

public void showLoginForm(){
    deviceDisplay.setCurrent(loginForm);
}

public void showMenu(){
    mainMenu.showMenu();
}

public void commandAction(Command c, Displayable d){
    if (swi.checkWord(userEntry.getString())){
        showMenu();
    }else{
        Alert err = new Alert("Invalid Password!", "Invalid Password\n
Please try again", null, AlertType.ERROR);
        err.setTimeout(Alert.FOREVER);
        deviceDisplay.setCurrent(err, loginForm);
    }
}
}

```