

## MODUL 9

### J2ME (Java 2 Micro Edition)

#### A. Tujuan :

1. Memahami konsep GUI Pilihan di J2ME.
2. Memahami konsep pembuatan command
3. Memahami passing parameter (manajemen event)

#### B. Dasar Teori

##### Ticker

Ticker merupakan teks yang berjalan pada bagian atas layer aplikasi MIDlet. Sebuah ticker dapat ditempatkan pada objek-objek turunan class Screen yang telah disebutkan yaitu TextBox, List, Form dan Alert. Untuk menambahkan ticker ke screen lakukan seperti berikut:

```
Ticker t = new Ticker("J2ME") ;  
t.setTicker(tick);  
Display.getDisplay(this).setCurrent(t);
```

keterangan :

t adalah Objek dari Textbox, sedangkan display adalah Objek Display.

##### Bekerja dengan Form

Bekerja dengan Form memungkinkan Anda untuk menampilkan beberapa komponen GUI semacam daftar pilihan, masukan teks (textbox) dalam satu layar. Form diimplementasikan oleh class javax.microedition.lcdui.Form. Form dapat menampung komponen-komponen yang disebut item dalam satu layar. Item tersebut adalah textfield, image, date field, gauge dan choice group. Item-item tersebut merupakan implementasi dari class turunan Item(javax.microedition.lcdui.Item).

```
public Form(String title)  
public Form(String title, Item[] items)
```

Konstruktor pertama, menyediakan sebuah form dengan judul form seperti parameter title yang diberikan, sedangkan konstruktor kedua mendefinisikan item-item apa saja yang akan ada di form yang akan dibuat.

##### Komponen Form :

- ChoiceGroup  
Class ini menyediakan komponen yang mirip dengan list, yakni menyediakan daftar pilihan.
- DateField  
Class ini menyediakan komponen untuk memasukkan informasi tanggal dan waktu.
- Gauge  
Class ini menyediakan komponen grafik horisontal yang biasanya digunakan untuk memberikan gambaran berapa persen proses berjalan.
- Image dan ImageItem  
Class ini menyediakan komponen grafik untuk manipulasi gambar.
- StringItem

Class ini menyediakan komponen teks string yang tidak bisa diedit oleh pengguna.

- **TextField**

Class ini menyediakan komponen untuk masukan teks string oleh pengguna.

### **TextField**

Objek TextField digunakan untuk meletakkan objek teks string yang bisa diubah oleh pengguna secara langsung pada form. Jadi objek ini mirip dengan objek TextBox. Objek TextField diimplementasikan oleh class TextField (*javax.microedition.lcdui.TextField*) yang merupakan class turunan dari class abstract *javax.microedition.lcdui.Item*.

Konstruktor dari class TextField adalah:

```
public TextField(String title, String text, int maxSize, int constraints)
```

Parameter-parameter pada konstruktor adalah:

- String title : untuk memberikan title pada TextField
- String text : untuk memberikan nilai awal pada TextField
- int maxsize : untuk memberikan batasan maksimal jumlah karakter
- int constraints

### **Ada beberapa nilai yang bisa digunakan untuk parameter ini yaitu :**

TextField.ANY

Artinya Anda bisa memasukkan sembarang karakter teks

TextField.EMAILADDR

Artinya Anda bisa memasukkan teks yang merupakan alamat email. Misalnya :  
intan@eepis-its.edu

TextField.NUMERIC

Artinya Anda hanya bisa memasukkan teks berupa angka saja

TextField.PHONENUMBER

Artinya Anda hanya bisa memasukkan teks berupa nomor telepon saja misalnya  
+6281234567

TextField.URL

Artinya Anda hanya bisa memasukkan teks berupa alamat web atau URI lain,  
misalnya

http://java.sun.com

TextField.PASSWORD

Seperti memasukkan password pada saat login ke Windows, masukan Anda akan ditampilkan dengan karakter tertentu, misalnya \*.

Tipe batasan ini bisa digunakan dengan tipe batasan yang lain dengan menggunakan operator | sehingga menambah kemampuan pembatasan yang ada.

Misalnya :

```
TextField t = new TextField ("", "", 25, TextField.PASSWORD |  
TextField.PHONENUMBER) ;  
TextField t = new TextField ("", "", 25, TextField.PASSWORD |  
TextField.NUMERIC) ;
```

### **C. Tugas Pendahuluan**

Buatlah desain flowchart untuk setiap soal dalam percobaan

### **D. Percobaan**

#### **D.1. Latihan**

##### **Latihan 1:**

Membuat aplikasi teks berjalan dengan Ticker.

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

public class tickerDemo extends MIDlet {
    public tickerDemo() {
    }
    public void startApp() {
        Ticker t = new Ticker ("Belajar Ticker di J2ME");
        Form form = new Form( "First Program" );
        form.append( "Hello World" );
        form.setTicker(t);
        Display.getDisplay(this).setCurrent( form );
    }
    public void pauseApp() {
    }
    public void destroyApp( boolean unconditional ) {
    }
}
```

##### **Latihan 2 :**

Membuat tombol dengan perintah Command untuk keluar dari suatu aplikasi.

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class cobaCmd extends MIDlet implements CommandListener {
    public void startApp() {
        Form d = new Form( "Test Command" );
        Command cb = new Command("Exit", Command.EXIT, 0);
        d.addCommand(cb);
        d.setCommandListener(this);
        Display.getDisplay(this).setCurrent(d);
    }
    public void pauseApp() { }
    public void destroyApp(boolean unconditional) { }
    public void commandAction(Command c, Displayable s) {
        notifyDestroyed();
    }
}
```

**Latihan 3:** Pembuatan navigasi antar Form dengan Command.

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class cobaCmd extends MIDlet implements CommandListener {
    Form satu, dua;
    Command keluarCmd = new Command ("Keluar",Command.EXIT,1);
    Command prosesCmd = new Command ("Proses",Command.OK,2);
    Command kembaliCmd = new Command ("Kembali",Command.BACK,2);
    public void startApp() {
        satu = new Form( "Form Satu" );
        satu.append ("Ini di Form Satu");
        satu.addCommand(keluarCmd);
        satu.addCommand(prosesCmd);
        satu.setCommandListener(this);
        Display.getDisplay(this).setCurrent(satu);
    }
    public void pauseApp() { }
    public void destroyApp(boolean unconditional) { }
    public void awal() {
        Display.getDisplay(this).setCurrent( satu );
    }
    public void keluar() {
        notifyDestroyed();
    }
    public void proses() {
        dua = new Form ( "Form Dua" );
        dua.append ("Ini di Form Dua");
        dua.addCommand(kembaliCmd);
        dua.addCommand(keluarCmd);
        dua.setCommandListener(this);
        Display.getDisplay(this).setCurrent(dua);
    }
    public void commandAction (Command c, Displayable d) {
        String data = c.getLabel();
        if (data == "Keluar") {
            keluar();
        } else if (data == "Proses") {
            proses();
        } else if (data == "Kembali") {
            awal();
        }
    }
}
```

**Latihan 4:** Membuat tampilan GUI berupa nama dan password.

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class passDemo extends MIDlet {
    public passDemo() {
    }
    public void startApp() {
        TextField nama = new TextField ("Nama : ","", 15,
        TextField.ANY);
        TextField pass = new TextField ("Password : ","", 15,
        TextField.PASSWORD);
        Form f = new Form( "Form Login" );
        f.append(nama);
        f.append(pass);
        Display.getDisplay(this).setCurrent( f );
    }
    public void pauseApp() {
    }
    public void destroyApp( boolean unconditional ) {
    }
}

```

**Latihan 5:**

Membuat tampilan GUI berupa nama dan password.

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class Text2 extends MIDlet implements CommandListener {
    Command keluar = new Command ("Keluar",Command.EXIT,1);
    Command proses = new Command ("Proses",Command.OK,2);
    Command kembali = new Command ("Kembali",Command.BACK,2);
    Form f; //agar bisa diakses di semua fungsi
    TextField tf; //agar bisa diakses di semua fungsi
    public void startApp() {
        tf = new TextField ("Nama Anda :","", 15, TextField.ANY);
        f = new Form( "Form Login" );
        f.append(tf);
        //Menambahkan obyek command
        f.addCommand(keluar);
        f.addCommand(proses);
        //Menghubungkan dengan perintah CommandListener
        f.setCommandListener(this);
        Display.getDisplay(this).setCurrent( f );
    }
    public void pauseApp() {
    }
    public void destroyApp( boolean unconditional ) {
    }
    public void awal() {
        Display.getDisplay(this).setCurrent( f );
    }
    public void keluar() {
        notifyDestroyed();
    }
}

```

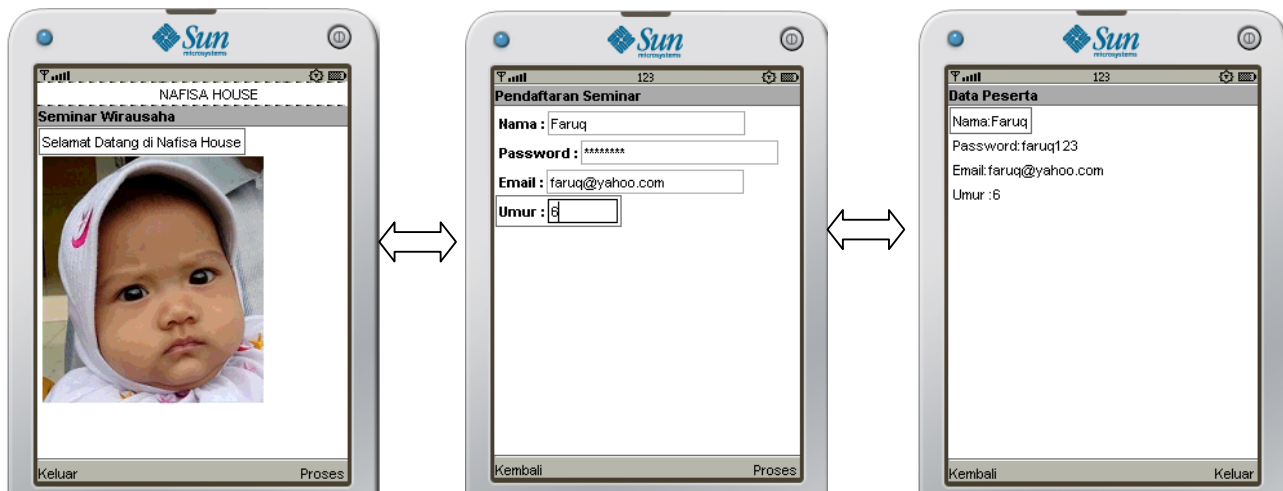
```

public void proses() {
    String dtproses = tf.getString();
    Form pf = new Form ("Proses Data");
    pf.append ("Nama anda : ");
    pf.append (dtproses);
    pf.addCommand (kembali);
    pf.setCommandListener(this);
    Display.getDisplay(this).setCurrent( pf );
}
public void commandAction (Command c, Displayable d) {
    if (c == keluar) {
        keluar();
    } else if (c == proses) {
        proses();
    } else if (c == kembali) {
        awal();
    }
}
}
}

```

**D.2. Permasalahan**

Untuk permasalahan dalam praktikum, lakukan langkah-langkah berikut :  
 Buat program aplikasi berbasis J2ME untuk melakukan proses berikut :



**E. Laporan Resmi**

Buatlah analisa dari program yang telah anda buat di praktikum.