

Mobile Computing

MIDP User Interface

Ramos Somya

MIDP UI API

- `javax.microedition.lcdui`



- Merupakan suatu desain user interface yang dikhususkan untuk perangkat dengan limited screen → perangkat mobile.
- MIDP UI dirancang secara fleksibel sehingga dapat digunakan di berbagai jenis perangkat mobile. Misal mobile phones dan PDA.

Pembagian MIDP UI API

- High Level UI
- Low Level UI

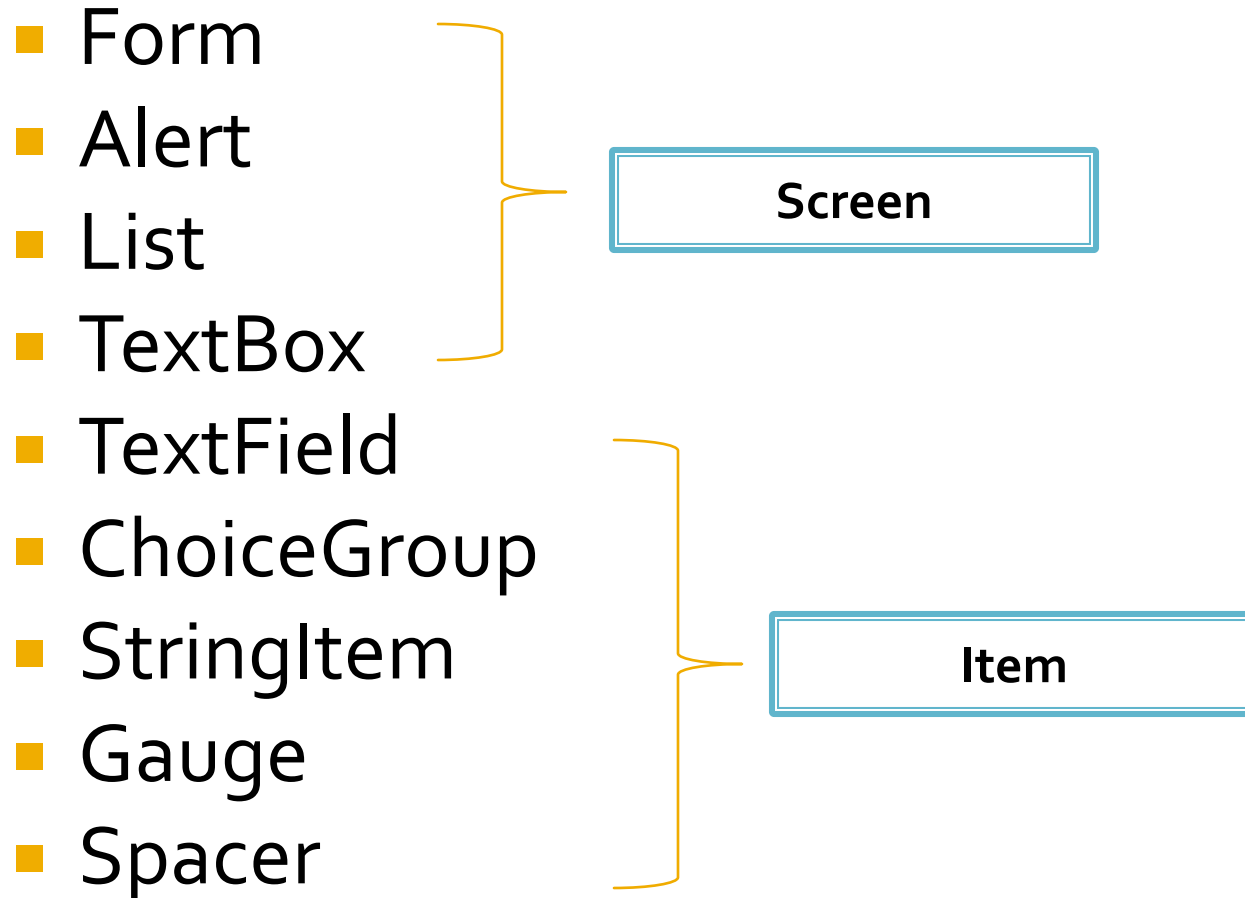
High Level UI

- Mendukung pemrograman aplikasi yang portable ke berbagai macam device.
- Menyediakan berbagai kelas untuk standard User Interface, seperti Form dan List.
- Digunakan untuk membangun aplikasi text based.
- Aplikasi dapat dipertukarkan dengan mudah dengan berbagai macam peralatan.
- Look and feel-nya sama dengan peralatannya.
- Kode program juga lebih sedikit.



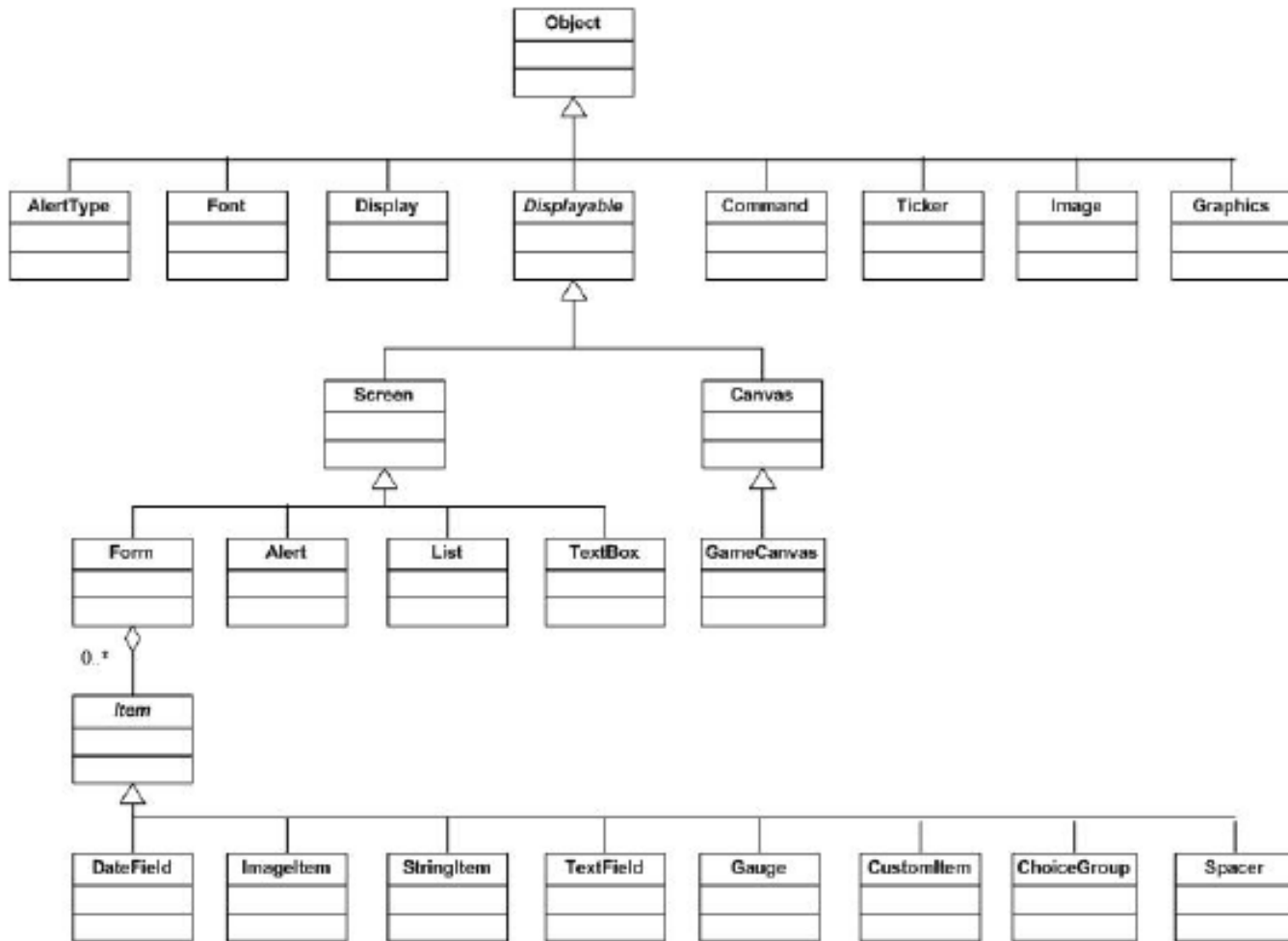
- Kelemahan High Level UI: tidak punya kendali penuh pada layar.
- Misal: pada saat kita ingin menggambar garis dan membuat gambar animasi pada layar yang membutuhkan kendali pixel pada layar.

Contoh Komponen High Level UI



Low Level UI

- Spesifik look and feel
- Digunakan untuk mengontrol tampilan (menggambar pixel, shape, rendering, font)
- Mempunyai kendali yang lebih leluasa pada layar.
- Biasanya digunakan untuk membuat mobile game secara grafik.
- Contoh: canvas.



Memprogram di emulator

- Emulator adalah program yang meniru kerja alat lain
 - Dalam kasus ini meniru HP yang mendukung MIDP
- Memprogram di emulator akan mempermudah proses development:
 - tidak perlu memindah program ke device (melelahkan dan butuh waktu)
 - mudah didebug
- Kelemahan emulator:
 - tidak semua fitur device diemulasikan dengan tepat
 - kadang program berjalan dengan cepat di emulator tapi sangat lambat di device yang sesungguhnya

contoh

- Buatlah sebuah project J2ME dan tambahkan 1 Midlet.

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

public class MyFirstMidlet extends MIDlet {
    public void startApp() {
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```



- **Import package standar untuk MIDlet :**
import javax.microedition.midlet.*;
- **Sebuah Aplikasi Midlet harus diturunkan dari kelas javax.microedition.midlet.MIDlet dan harus mengimplementasikan 3 method abstrak startApp(), pauseApp() dan destroyApp();**



- **startApp()** dipanggil ketika aplikasi dimulai
- **pauseApp()** dipanggil ketika aplikasi dihentikan sementara
 - misalnya ketika ada telepon atau SMS datang
 - Catatan: tidak semua HP mengimplementasikan ini dengan benar (kadang **pauseApp()** tidak dipanggil)
- **destroyApp(boolean unconditional)** dipanggil ketika aplikasi akan dihentikan
 - jika **unconditional** bernilai **true** midlet harus berhenti, jika **false**, midlet boleh meminta agar tidak dihentikan
 - Hal yang dilakukan di **destroyApp()** misalnya adalah menyimpan data

Contoh AlertMidlet

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

public class AlertMidlet extends MIDlet {

    Display display;
    Alert helloAlert;

    public AlertMidlet() {
        helloAlert = new Alert("Salam Damai", "Hello, world\n\nHola....", null, AlertType.INFO);
        helloAlert.setTimeout(Alert.FOREVER);
    }

    public void startApp() {
        if (display == null) {
            display = Display.getDisplay(this);
        }
        display.setCurrent(helloAlert);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```



- **Display** merepresentasikan manager dari tampilan (display) dan menyediakan akses ke layar.
- **Alert** merupakan suatu screen yang menampilkan informasi kepada user dan menunggu dalam waktu tertentu, kemudian akan digantikan dengan tampilan lain.

CommandListener → interface untuk menangani command

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

public class AlertMidlet extends MIDlet implements CommandListener {

    Display display;
    Alert helloAlert;
    Command exitCommand = new Command("Selesai", Command.EXIT, 2);
    Command infoCommand = new Command("Info", Command.OK, 1);

    public AlertMidlet() {
        helloAlert = new Alert("Salam Damai", "Hello, world\n\nHola....", null,
AlertType.INFO);
        helloAlert.setTimeout(Alert.FOREVER);
        helloAlert.addCommand(exitCommand);
        helloAlert.addCommand(infoCommand);
        helloAlert.setCommandListener(this); // mendeteksi penekanan tombol
    }
}
```



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

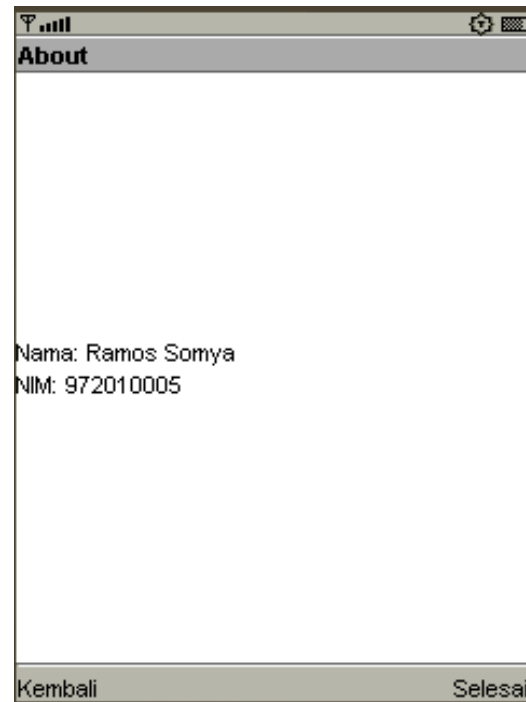
public void pauseApp() {
}

public void destroyApp(boolean unconditional) {
}

public void commandAction(Command c, Displayable d) {
    if (c == exitCommand) {
        destroyApp(true);
        notifyDestroyed();
    }
}
}
```

Latihan

- Tambahkan 1 buah alert untuk menginformasikan nama dan NIM Anda. Kemudian tambahkan tombol back dan exit pada alert tersebut.



Form

- Form merupakan screen yang dapat memuat item-item, misalnya image, textfield, gauge, dan sebagainya.
- Secara umum, semua kelas yang merupakan kelas turunan dari kelas Item dapat ditempatkan dalam form
- StringItem merupakan item yang memuat string.
- String item digunakan untuk menampilkan string, user tidak dapat mengedit isi dari string item
- TextField merupakan komponen teks yang dapat diedit dan ditempatkan pada form



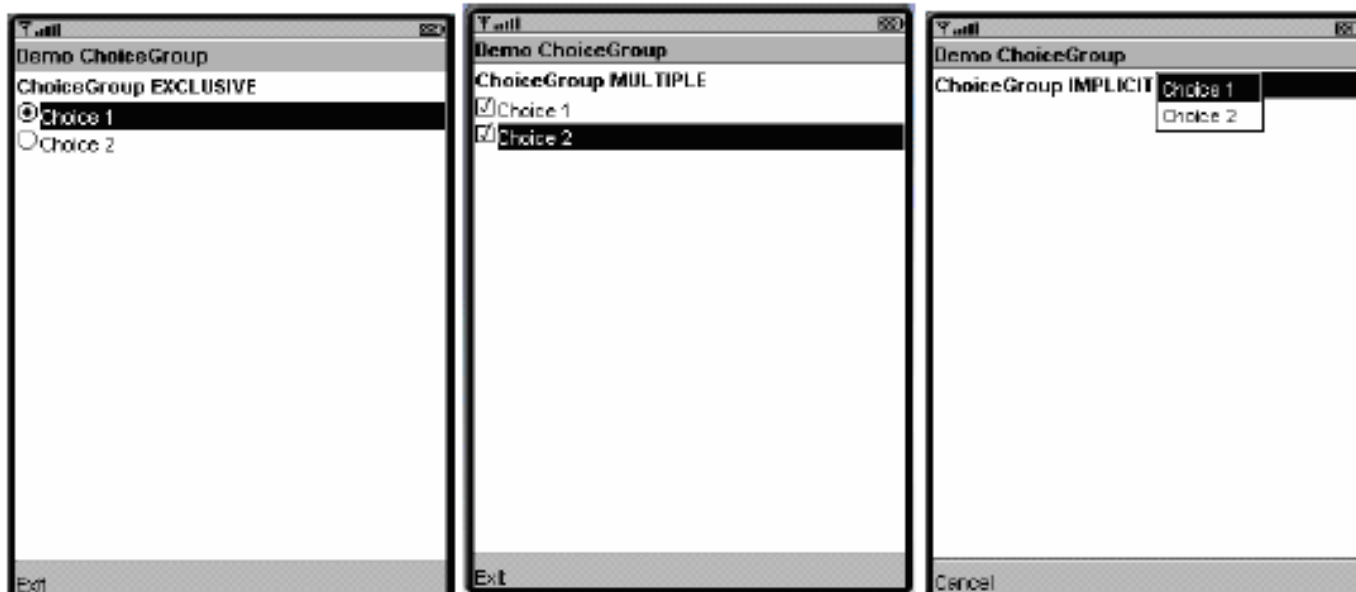
Constraint	Deskripsi
<code>TextField.ANY</code>	Text field dapat diisi teks apa saja
<code>TextField.EMAILADDR</code>	Text field hanya menerima teks berformat alamat email
<code>TextField.NUMERIC</code>	Text field hanya menerima input angka
<code>TextField.PHONENUMBER</code>	Text field hanya menerima input nomor telepon
<code>TextField.URL</code>	Text field hanya menerima input alamat URL
<code>TextField.DECIMAL</code>	Text field menerima input angka yang disertai dengan tanda minus (-) dan titik desimal (.)



Modifier	Deskripsi
<code>TextField.PASSWORD</code>	Biasanya digunakan bersama <code>TextField.ANY</code> atau <code>TextField.NUMERIC</code> untuk text field yang berguna menerima pengisian password (karakter masukan akan diubah menjadi asterik)
<code>TextField.UNEDITABLE</code>	Teks pada text field tidak dapat diedit
<code>TextField.SENSITIVE</code>	Menandakan bahwa teks yang dimasukkan adalah data yang sensitif sehingga tidak boleh disimpan ke tabel atau dictionary
<code>TextField.NON_PREDICTIVE</code>	Teks yang dimasukkan pada text field ini bukan merupakan teks yang dapat diprediksi (atau kata yang biasa terdapat dalam kamus)
<code>TextField.INITIAL_CAPS_WORD</code>	Huruf awal dari kata adalah huruf besar
<code>TextField.INITIAL_CAPS_SENTENCE</code>	Huruf awal dari kalimat adalah huruf besar



- **ChoiceGroup** merupakan sekumpulan elemen untuk memberikan pilihan pada user.
- **ChoiceGroup** digunakan sebagai item pada Form.



a. EXCLUSIVE b. MULTIPLE c. POPUP

Contoh Form

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

public class MyFirstMidlet extends MIDlet {

    Form f = new Form("Hello World");

    public void startApp() {
        Display.getDisplay(this).setCurrent(f);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```

Contoh Form Lanjut

.....

Tugas Kelas

- Buatlah program J2ME untuk mengkonversi Angka ke Biner dan Biner ke Angka.

Form Utama (Ramos Somya)

Pilihan Anda:

Angka ke Biner

Biner ke Angka

Keluar Pilih

Angka ke Biner

Masukkan Angka : 87

Hasil Konversi: 1010111

Kembali Konversi

Biner ke Angka

Masukkan Biner : 1010111

Hasil Konversi: 87

Kembali Konversi

Ketentuan

- Tugas dikirim ke email: ramos.somya@gmail.com
- Paling lambat Sabtu, 1 Oktober 2011 pukul 12.00 WIB.
- Subject: TGS_MOBCOM1
- Nama Project: TGS_UI1_NIMLENGKAP.rar

See You Next Week