1.Tujuan

- Mengetahui keuntungan dan kerugian dengan menggunakan high-level dan low-level UI classes
- Mengetahui desain MIDlets menggunakan komponen high-level UI
- Mengidentifikasi perbedaan sub-classes pada screen
- Mengetahui perbedaan item-item yang dapat dimasukkan kedalam sebuah object Form

2. Latar Belakang

MDIP user interface didesain untuk peralatan mobile. Aplikasi MDIP ditunjukan pada area limited screen. Peralatan memory juga menjadi faktor penting jika perlengkapan mobile hanya memiliki kapasitas memory yang kecil.

Dengan berbagai macam peralatan mobile, dari berbagai model mobile phones sampai PDAs, MIDP user interface telah didesain untuk lebih fleksibel dan mudah digunakan dalam berbagai macam peralatan ini.

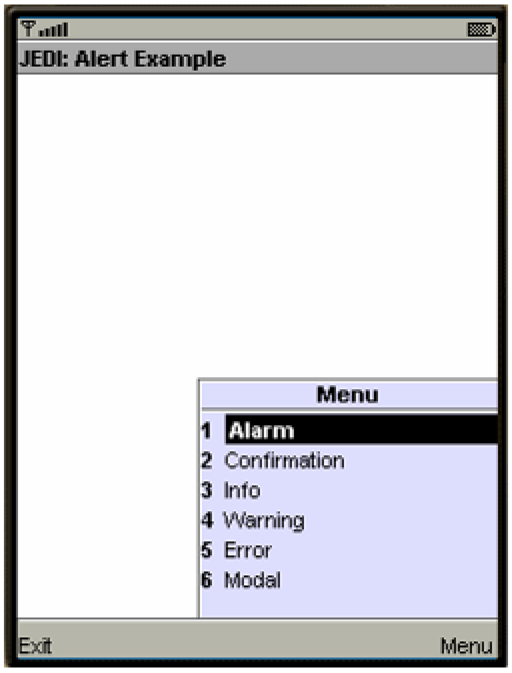
MIDP mempunyai class yang dapat menangani fungsi high-level dan low-level user interface. High-level UI interfaces didesain secara fleksibel. Penampilan dari komponen ini tidak didefinisikan secara spesifik. Penampilan screen yang sebenarnya dari berbagai macam komponen ini digunakan dari satu peralatan ke peralatan yang lain. Tetapi para programmer telah teryakinkan oleh kegunaan dari high-level komponen UI interfaces memiliki persamaan dalam berbagai spesifikasi-pengimplementasi secara keseluruhan.

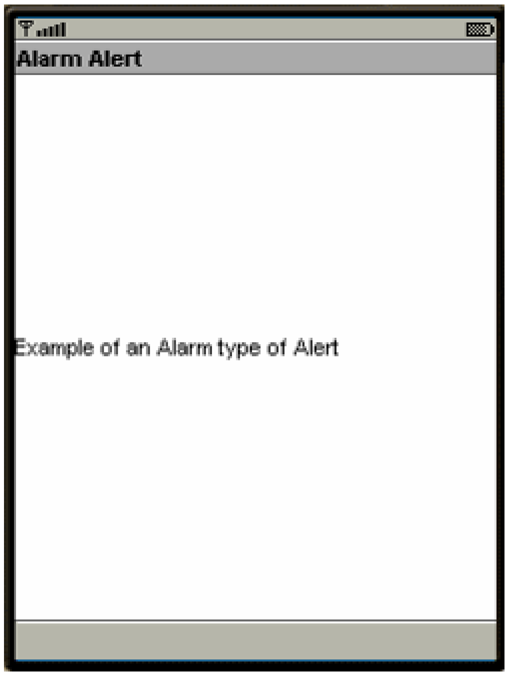
3. Percobaan

Percobaan 1:Penggunaan Alert

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class AlertExample extends MIDlet implements CommandListener {
      Display display;
      Form mainForm;
      Command exitCommand = new Command("Exit", Command.EXIT, 0);
      Command okCommand = new Command("Ok", Command.OK, 0);
      Gauge gauge = new Gauge(null, false, 5, 0);
      Command[] commands = {
            new Command("Alarm", Command.OK, 0),
            new Command("Confirmation", Command.OK, 0),
            new Command("Info", Command.OK, 0),
            new Command("Warning", Command.OK, 0),
            new Command("Error", Command.OK, 0),
            new Command("Modal", Command.OK, 0)
      };
      Alert[] alerts = {
            new Alert("Alarm Alert",
                  "Example of an Alarm type of Alert",
                  null, AlertType.ALARM),
            new Alert("Confirmation Alert",
                  "Example of an CONFIRMATION type of Alert",
                  null, AlertType.CONFIRMATION),
            new Alert("Info Alert",
                  "Example of an INFO type of Alert",
                  null, AlertType.INFO),
            new Alert("Warning Alert",
                  "Example of an WARNING type of Alert, w/ gauge indicator",
                  null, AlertType.WARNING),
            new Alert("Error Alert",
                  "Example of an ERROR type of Alert, w/ an 'OK' Command",
                  null, AlertType.ERROR),
            new Alert("Modal Alert",
                  "Example of an modal Alert: timeout = FOREVER",
                  null, AlertType.ERROR),
      };
      public AlertExample(){
            mainForm = new Form("JEDI: Alert Example");
            mainForm.addCommand(exitCommand);
            for (int i=0; i< commands.length; i++){
                  mainForm.addCommand(commands[i]);
            }
            mainForm.setCommandListener(this);
            // Menambah sebuah gauge dan menge-set timeout (milliseconds)
            alerts[3].setIndicator(gauge);
            alerts[3].setTimeout(5000);
            // Menambah sebuah command untuk Alert
            alerts[4].addCommand(okCommand);
            // Menge-Set alert
            alerts[5].setTimeout(Alert.FOREVER);
      }
      public void startApp() {
            if (display == null){
                  display = Display.getDisplay(this);
                  display.setCurrent(mainForm);
            }
      }
      public void pauseApp() {}
      public void destroyApp(boolean unconditional) {}
      public void commandAction(Command c, Displayable d){
            if (c == exitCommand){
                  destroyApp(true);
                  notifyDestroyed(); // Exit
            }
            for (int i=0; i<commands.length; i++){
                  if (c == commands[i]){
                        display.setCurrent(alerts[i]);
                  }
            }
      }
}
```

Hasil :





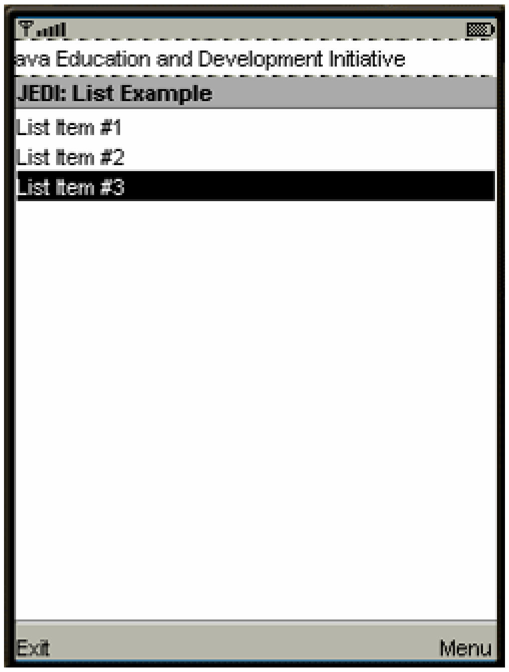
Example of an Alarm type of Alert

Percobaan 2: Penggunaan List

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class ListExample extends MIDlet implements CommandListener {
	Display display;
	List list;
	Command exitCommand = new Command("Exit", Command.EXIT, 1);
	Command newCommand = new Command("New Item", Command.OK, 1);
	Command renameCommand = new Command("Rename Item", Command.OK, 1);
	Command deleteCommand = new Command("Delete Item", Command.OK, 1);
	Ticker ticker = new Ticker(
		"JEDI - Java Education and Development Initiative");

	public ListExample(){
		list = new List("JEDI: List Example", List.IMPLICIT);
		list.append("List Item #1", null);
		list.append("List Item #2", null);
		list.append("List Item #3", null);
		list.setTicker(ticker);
		list.addCommand(exitCommand);
		list.addCommand(newCommand);
		list.addCommand(renameCommand);
		list.addCommand(deleteCommand);
		list.setCommandListener(this);
	}

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

	public void pauseApp() {
	}

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

```
      public void commandAction(Command c, Displayable d){
            if (c == exitCommand){
                  destroyApp(true);
                  notifyDestroyed(); // Exit
            }
            if (c == List.SELECT_COMMAND){
                  int index = list.getSelectedIndex();
                  String currentItem = list.getString(index);
                  // menjalankan suatu hal
            }
      }
}
```

Hasil :



ava Education and Development Initiative

JEDI: List Example

List Item #1

List Item #2

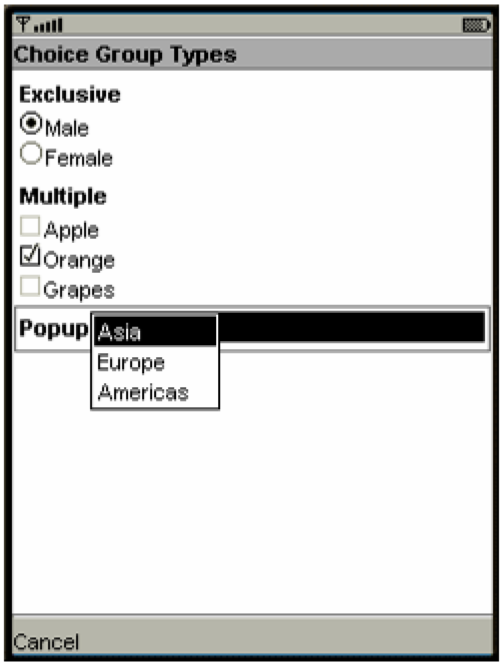
List Item #3

Percobaan 3: Choice Group

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class ChoiceExample extends MIDlet implements CommandListener {
	Display display;
	Form choiceForm;
	Command exitCommand = new Command("Exit", Command.EXIT, 1);
	ChoiceGroup choiceExclusive,choiceMultiple,choicePopup;

	public ChoiceExample(){
		choiceForm = new Form("Choice Group Types");
		choiceForm.addCommand(exitCommand);
		choiceForm.setCommandListener(this);
		choiceExclusive = new ChoiceGroup("Exclusive", Choice.EXCLUSIVE);
		choiceExclusive.append("Male", null);
		choiceExclusive.append("Female", null);
		choiceForm.append(choiceExclusive);

		choiceMultiple = new ChoiceGroup("Multiple", Choice.MULTIPLE);
		choiceMultiple.append("Apple", null);
		choiceMultiple.append("Orange", null);
		choiceMultiple.append("Grapes", null);
		choiceForm.append(choiceMultiple);

		choicePopup = new ChoiceGroup("Popup", Choice.POPUP);
		choicePopup.append("Asia", null);
		choicePopup.append("Europe", null);
		choicePopup.append("Americas", null);
		choiceForm.append(choicePopup);
	}

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

	public void pauseApp() {
	}
```

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

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

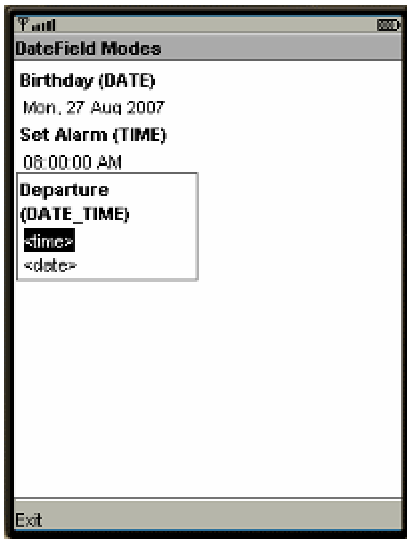
Hasil :



Percobaan 4: Date Field

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class DateFieldExample extends MIDlet implements CommandListener {
	Display display;
	Form dateForm;
	Command exitCommand = new Command("Exit", Command.EXIT, 1);
	DateField dateonly,timeonly,datetime;

public DateFieldExample(){
		dateForm = new Form("DateField Modes");
		dateForm.addCommand(exitCommand);
		dateForm.setCommandListener(this);
		DateField dateonly =
			new DateField("Birthday (DATE)", DateField.DATE);
		DateField timeonly =
			new DateField("Set Alarm (TIME)", DateField.TIME);
		DateField datetime =
			new DateField("Departure (DATE_TIME)", DateField.DATE_TIME);
		dateForm.append(dateonly);
		dateForm.append(timeonly);
		dateForm.append(datetime);
	}

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

	public void pauseApp() {
	}

	public void destroyApp(boolean unconditional) {
	}

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

Hasil :


DateField Modes

Birthday (DATE)

Mon, 27 Aug 2007

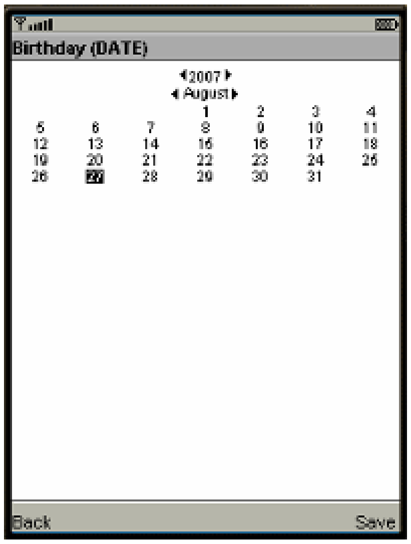
Set Alarm (TIME)

08:00:00 AM

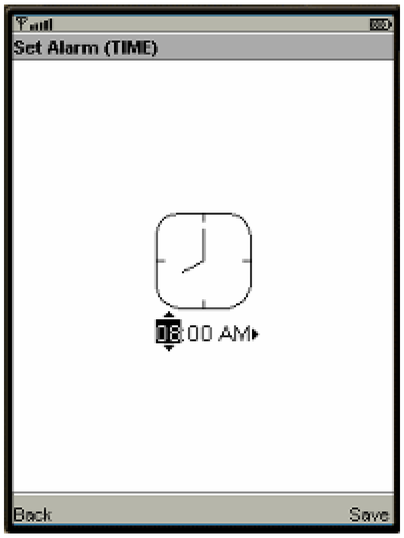
Departure (DATE_TIME)

<time>

<date>


Birthday (DATE)

```
              2007
             August
                 1    2    3    4
5    6    7    8    9   10   11
12   13   14   15   16   17   18
19   20   21   22   23   24   25
26   27   28   29   30   31
```


Set Alarm (TIME)

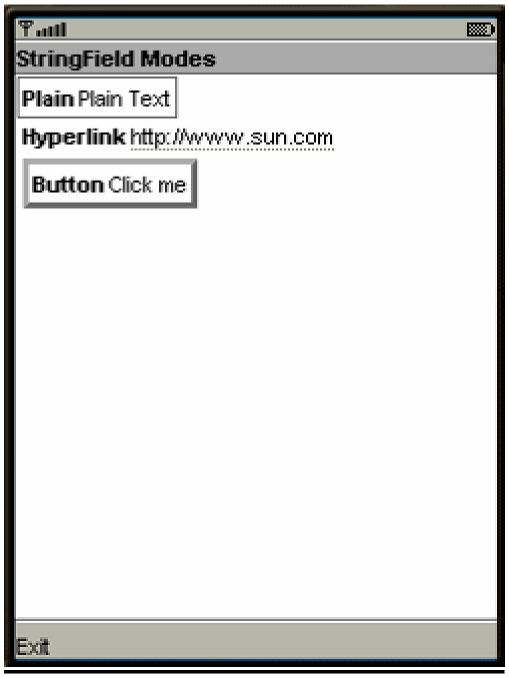
08:00 AM

Percobaan 5: Penggunaan StringItem

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class StringExample extends MIDlet
                          implements CommandListener,ItemCommandListener {
      Display display;
      Form stringForm;
      Command exitCommand = new Command("Exit", Command.EXIT, 1);
      DateField dateonly,timeonly,datetime;

      public StringExample(){
            stringForm = new Form("StringField Modes");
            stringForm.addCommand(exitCommand);
            stringForm.setCommandListener(this);
            StringItem plain = new StringItem("Plain", "Plain Text", Item.PLAIN);
            StringItem hyperlink = new StringItem("Hyperlink", "http://www.sun.com",
                                          Item.HYPERLINK);
            hyperlink.setDefaultCommand(new Command("Set", Command.ITEM, 0));
            hyperlink.setItemCommandListener(this);
            StringItem button = new StringItem("Button", "Click me", Item.BUTTON);
            button.setDefaultCommand(new Command("Set", Command.ITEM, 0));
            button.setItemCommandListener(this);
            stringForm.append(plain);
```

```
                stringForm.append(hyperlink);
                stringForm.append(button);
        }

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

        public void pauseApp() {
        }

        public void destroyApp(boolean unconditional) {
        }

        public void commandAction(Command c, Displayable d){
                if (c == exitCommand){
                        destroyApp(true);
                        notifyDestroyed(); // Exit
                }
        }
        public void commandAction(Command c, Item item){
                if(item.getLabel().equals("Button")){
                        //kerjakan Sesuatu
                }
                if(item.getLabel().equals("Hyperlink")){
                        //kerjakan Sesuatu
                }
        }
}
```

Hasil :

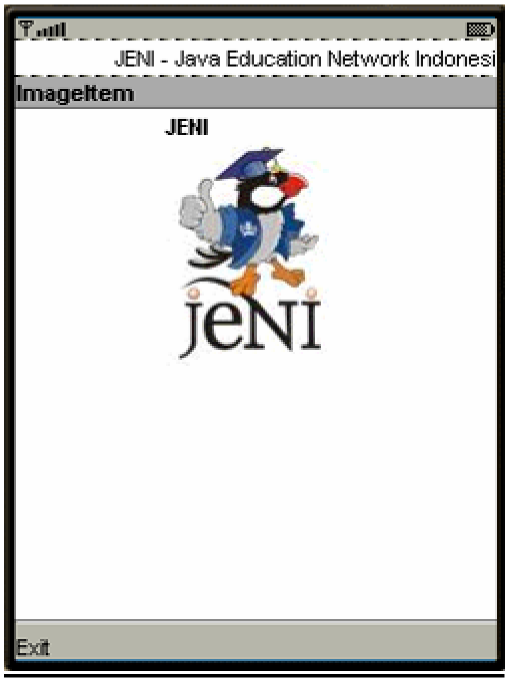


Percobaan 6: Penggunaan Image

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class ImageExample extends MIDlet implements CommandListener {
       Display display;
       Form imageForm;
       Command exitCommand = new Command("Exit", Command.EXIT, 1);
       DateField dateonly,timeonly,datetime;
       Ticker ticker = new Ticker(
              "JENI - Java Education Network Indonesia");

       public ImageExample(){
              imageForm = new Form("ImageItem");
              imageForm.addCommand(exitCommand);
              imageForm.setCommandListener(this);
              imageForm.setTicker(ticker);
              try {
                     Image img = Image.createImage("/JENI.png");
```

```
                    ImageItem image =
                          new ImageItem("JENI", img, Item.LAYOUT_CENTER, "jeni logo");
                          imageForm.append(image);
              } catch (Exception e){e.printStackTrace();}
       }

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

       public void pauseApp() {
       }

       public void destroyApp(boolean unconditional) {
       }

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

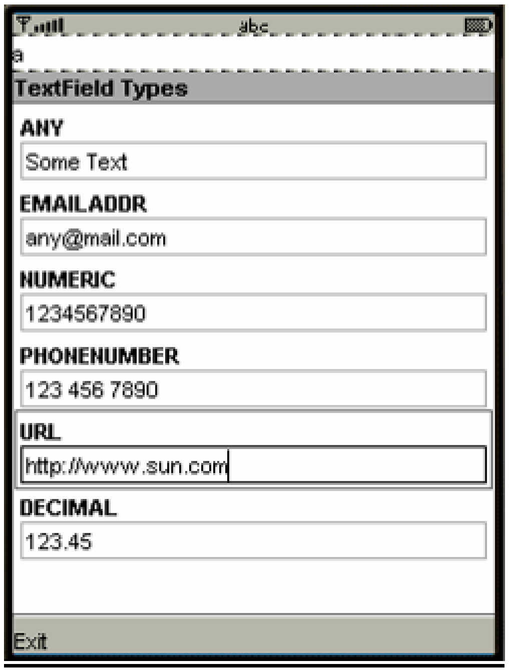
<u>Hasil :</u>


JENI - Java Education Network Indonesi
ImageItem
JENI
jeNi
Exit

Percobaan 7: Penggunaan TextField

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class TextFieldExample extends MIDlet implements CommandListener {
	Display display;
	Form textForm;
	Command exitCommand = new Command("Exit", Command.EXIT, 1);
	DateField dateonly,timeonly,datetime;
	Ticker ticker = new Ticker(
		"JENI - Java Education Network Indonesia");
	public TextFieldExample(){
		textForm = new Form("TextField Types");
		textForm.addCommand(exitCommand);
		textForm.setCommandListener(this);
		TextField ANY = new TextField("ANY", "", 64, TextField.ANY);
		TextField EMAILADDR =
			new TextField("EMAILADDR", "", 64, TextField.EMAILADDR);
		TextField NUMERIC =
			new TextField("NUMERIC", "", 64, TextField.NUMERIC);
		TextField PHONENUMBER =
			new TextField("PHONENUMBER", "", 64, TextField.PHONENUMBER);
		TextField URL =
			new TextField("URL", "", 64, TextField.URL);
		TextField DECIMAL =
			new TextField("DECIMAL", "", 64, TextField.DECIMAL);
		textForm.append(ANY);
		textForm.append(EMAILADDR);
		textForm.append(NUMERIC);
		textForm.append(PHONENUMBER);
		textForm.append(URL);
		textForm.append(DECIMAL);
		textForm.setTicker(ticker);
	}
	public void startApp() {
		if (display == null){
			display = Display.getDisplay(this);
			display.setCurrent(textForm);
		}
	}
	public void pauseApp() {
	}
	public void destroyApp(boolean unconditional) {
	}
	public void commandAction(Command c, Displayable d){
		if (c == exitCommand){
			destroyApp(true);
			notifyDestroyed(); // Exit
		}
	}
}
```

Hasil :



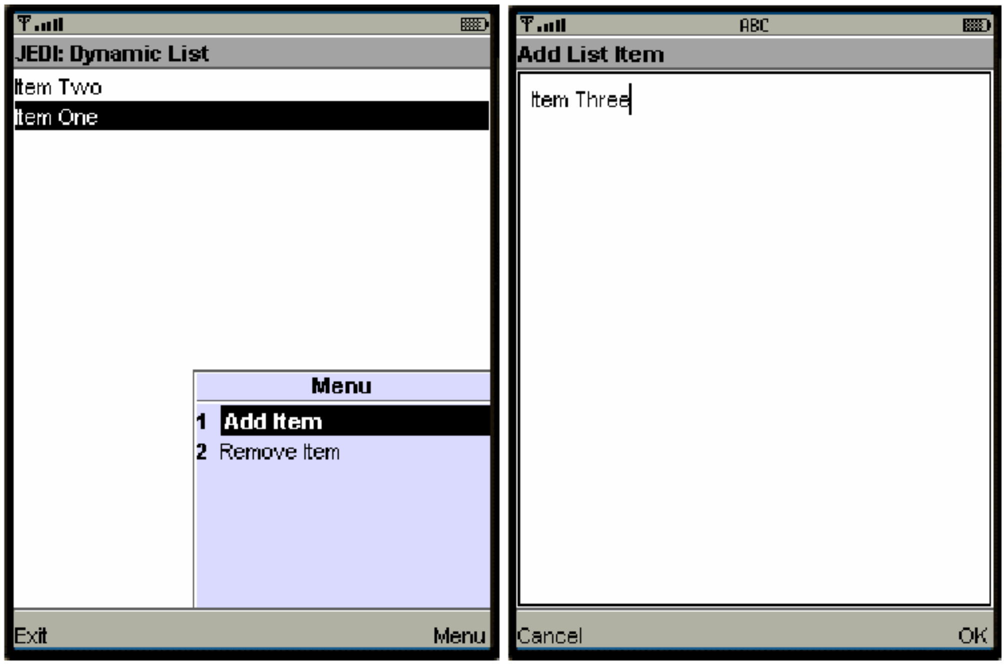
ANY
Some Text
EMAILADDR
any@mail.com
NUMERIC
1234567890
PHONENUMBER
123 456 7890
URL
http://www.sun.com
DECIMAL
123.45

4.Latihan

3.12.1 List Dinamis

Buatlah sebuah MIDlet yang memiliki List IMPLICIT sebagai Screen main. Masukan tiga Command kedalam List ini - “Add Item”, “Remove Item” dan “Exit”. Comman “Add Item” akan memberikan layanan pada user untuk memasukan list menggunakan TextBox, kemudian insert item tersebut sebelum current item yang dipilih dari list. “Remove Item” akan menghapus currently selected list item (getSelectedIndex). Command “Exit” akankeluar dari program.

Hasil Output:



Jawaban:

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

public class DynamicList extends MIDlet implements CommandListener {
```

```
Display display;
List list;
TextBox textbox;
Command exitCommand = new Command("Exit", Command.EXIT, 1);
Command addCommand = new Command("Add Item", Command.OK, 1);
Command removeCommand = new Command("Remove Item", Command.OK, 1);

Command okCommand = new Command("OK", Command.OK, 1);
Command cancelCommand = new Command("Cancel", Command.EXIT, 1);

public DynamicList(){
    list = new List("JEDI: Dynamic List", List.IMPLICIT);
    list.addCommand(exitCommand);
    list.addCommand(addCommand);
    list.addCommand(removeCommand);
    list.setCommandListener(this);

    textbox = new TextBox("Add List Item", "", 64, TextField.ANY);
    textbox.addCommand(okCommand);
    textbox.addCommand(cancelCommand);
    textbox.setCommandListener(this);
}
public void startApp() {
    if (display == null){
        display = Display.getDisplay(this);
        display.setCurrent(list);
    }
}
public void pauseApp() {
}
public void destroyApp(boolean unconditional) {
}
public void commandAction(Command c, Displayable d){
    if (c == exitCommand){
        notifyDestroyed(); // Exit
    }
    if (c == addCommand){
        textbox.setString("");
        display.setCurrent(textbox);
    }
    if (c == removeCommand){
        int index = list.getSelectedIndex();
        if (index != -1){
            list.delete(index);
        } else {
            // no item selected
        }
    }
    if (c == okCommand){
        int index = list.getSelectedIndex();
```

```
            if (index != -1){
                list.insert(index, textbox.getString(), null);
            } else {
                // list might be empty
                list.append(textbox.getString(), null);
            }
            display.setCurrent(list);
        }
        if (c == cancelCommand){
            display.setCurrent(list);
        }
    }
}
```