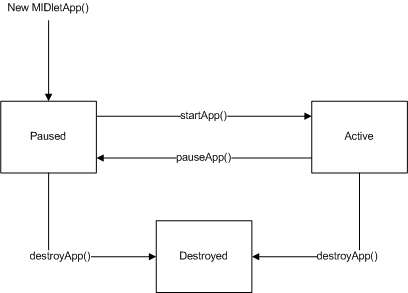
**PRAKTIKUM VII**

**PEMBUATAN CLASS FORM DAN CLASS TEXTFIELD PADA MIDLET DI JAVA ME**

1. **Tujuan Praktikum**
2. Memahami konsep pembuatan form dan textfield yang ada pada class screen.
3. Dapat memahami dan menguasai penggunaan form dan textfield pada Pemograman Bahasa Java me.
4. **Dasar Teori**
5. **Pengertian MIDlet**

Aplikasi yang dibuat dengan menggunakan J2ME dengan standar MIDP disebut MIDlet.Untuk membuat MIDlet diperlukan satu kelas yang menjadi turunan dari kelas java.microedition.midlet.MIDlet.

MIDlet mempunyai siklus hidup/life cycle. Suatu MIDlet setelah diciptakan akan berada di salah satu status (active, paused, destroyed). Begitu obyek suatu MIDlet diciptakan akan memasuki status paused menunggu perintah berikutnya. MIDlet memasuki status active ketika setelah method startApp() dipanggil. MIDlet akan kembali ke status pause jika method pauseApp() dipanggil. Setelah semua proses di dalam MIDlet selesai dilakukan dan MIDlet dimatikan maka MIDlet berada dalam status destroyed. Status destroyed ini dimasuki MIDlet pada pemanggilan method destroyApp(boolean).



Gambar 1. Siklus hidup (Life Cycle) MIDlet

1. **Antarmuka di J2ME**

J2ME, dalam MIDP, seperti halnya kategori java lainnya (J2SE) mendefinisikan serangkaian library untuk membuat tampilan antarmuka penguna/user interface. Pada MIDP terdapat dua jenis library antarmuka pengguna yaitu:

1. Antarmuka tingkat atas/High‐level User Interface
2. Antarmuka tingkat bawah/Low‐level interface

Antarmuka tingkat atas terdiri dari komponen‐komponen yang telah terdefinisi/well‐defined dan siap langsung digunakan dalam MIDlet. Komponen ini mampu merespon input dari pengguna langsung tanpa penambahan kode. Semua komponen‐komponen yang berada pada tingkat atas ini merupakan turunan dari sebuah kelas yang bernama Screen. Pada bab ini khusus dibahas antarmuka tingkat atas ini.

Antarmuka tingkat bawah menyediakan kendali lebih dalam pembentukan komponen baik tampilan maupun interaksinya. Antarmuka jenis ini biasanya digunakan untuk pembuatan antarmuka pengguna grafis/graphical user interface pada aplikasi game maupun aplikasi‐aplikasi lain yang memerlukan komponen antarmuka yang tidak terdapat pada antarmuka pengguna tingkat atas. Antarmuka tingkat ini direpresentasikan oleh Canvas dan turunannya yaitu GameCanvas. Pembahasan mengenai antarmuka tingkat bawah ini ada pada Bab III.

1. **Display, Displayable dan Ticker**

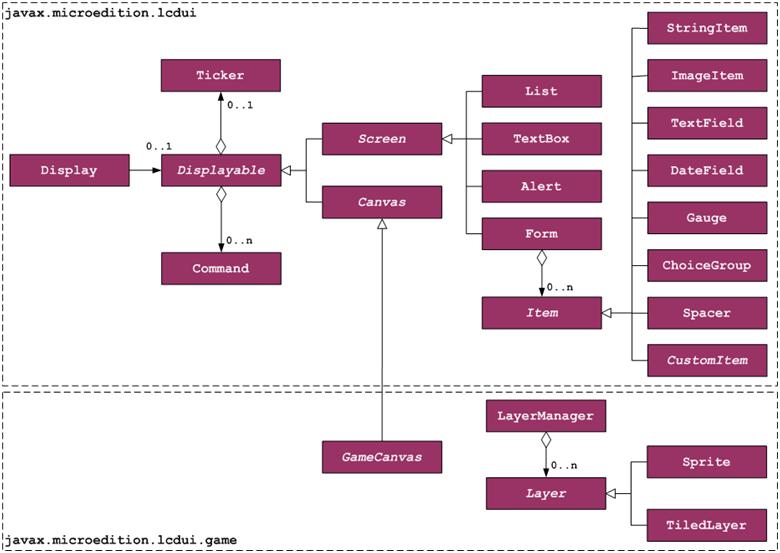
Display adalah representasi dari layar tampilan perangkat bergerak. Display ini tidak dapat diciptakan dan hanya ada satu (singleton). Untuk mendapatkan obyek ini dapat digunakan static method dari Display yaitu sebagai berikut.

static Display getDisplay(Midlet m)

Obyek yang dapat ditampilkan melalui obyek Display ini adalah obyek Displayable dan turunannya serta obyek Alert. Jika ingin menampilkan Alert maka obyek Displayable yang akan ditampilkan selanjutnya juga harus didefinisikan. Untuk menampilkan obyek pada layar perangkat digunakan method sebagai berikut

1. void setCurrent(Displayable nextDisplayable)
2. void setCurrent(Alert alert, Displayable nextDisplayable)

Obyek Displayable adalah sebuah obyek yang dapat tampil pada obyek Display sendirian tanpa kehadiran obyek lagi. Turunan Displayable ini ada dua yaitu Screen dan Canvas. Screen dan turunannya masuk ke dalam antarmuka tingkat atas, dibahas di bab ini, sedangkan Canvas dan turunannya masuk dalam antarmuka tingkat bawah, dibahas pada bab berikutnya.



Gambar 2. Hirarki kelas antarmuka (tingkat atas dan bawah) pada J2ME

1. **Form**

Form adalah komponen turunan Screen yang berfungsi sebagai tempat untuk menampung (container) bagi komponen lain. Fungsi Form pada J2ME sama seperti Form pada HTML. Form selain memiliki obyek Item biasanya juga mempunyai obyek Command. Fungsi Command pada Form ini juga sama dengan fungsi Button pada Form di HTML. Command berfungsi sebagai trigger untuk memulai suatu proses. Form juga dapat memasukkan String ataupun Image. Dalam hal ini obyek‐obyek tersebut dibungkus menjadi Obyek turunan Item terlebih dahulu. Mengenai hal ini akan dibahas pasa sub bab tentang StringItem dan ImageItem. Untuk memasukkan suatu obyek Item ke dalam Form dapat digunakan beberapa method sebagai berikut:

1. void insert(int itemNum, Item item)
2. int append(Image img)
3. int append(Item item)
4. int append(String str)

Method insert() digunakan jika kita ingin memasukkan suatu Item pada posisi tertentu dalam Form. Posisi ini ditentukan dengan biangan indeks. Sedangkan method append() digunakan untuk memasukkan Item pada posisi paling akhir/bawah pada Form. Method append() mengembalikan posisi dari suatu Item dalam bilangan indeks.

Sedangkan untuk menghapus obyek Item dari Form dapat digunakan method delete(int) atau

deleteAll() untuk menghapus semua obyek Item di dalam Form.

Form mempunyai sifat seperti Collection. Obyek‐obyek Item yang tersimpan di dalamnya dapat diakses dengan menggunakan index, method yang digunakan adalah get(int), dan dapat diketahui jumlah total obyek Item didalamnya dengan menggunakan method size(). Index obyek Item dalam Form dimulai dari 0 sehingga index maksimum adalah size()‐1.

TIdak ada batasan berapa jumlah minimum dan maksimum obyek Item yang dapat dimasukkan ke Form. Akan tetapi Form sebaiknya dibuat sependek dan sesederhana mungkin karena pada umumnya pengguna perangkat bergerak tidak menyukai scrolling dan Form yang terlalu kompleks.

1. **TextField**

TextField adalah komponen turunan Item yang menampung teks dan mengedit teks tersebut. TextField mempunyai ukuran maksimum, label dan format input. Beberapa vendor mengimplementasikan TextField sehingga ukuran TextField dapat secara otomatis membesar sesuai dengan input string yang dimasukkan. Berapa jumlah baris dan kolom yang ditampilkan TextField menjadi tanggung jawab dari vendor perangkat bergerak dalam pengimplementasiannya. Ada komponen lain selain TextField di antarmuka tingkat atas yang mempunyai fungsi sama dengan textField yaitu TextBox. TextBox merupakan turunan Screen dan buka turunan Item. Perbedaan antara TextField dan TextBox terletak pada jumlah maksimum karakter, multi‐line dan format input.

Masukan/input pada TextField dapat di format sesuai kebutuhan. Ada enam jenis format yang dikenali oleh TextField yaitu ANY, EMAILADDR, NUMERIC, PHONENUMBER, URL dan DECIMAL. Semua format input diatas merupakan konstanta static dari class TextField. Untuk format EMAILADDR, PHONENUMBER dan URL dapat mendapatkan perlakuan khusus tergantung implementasi vendor. Perlakukan khusus ini antara lain kelebihan untuk melakukan lookup atau melihat ke dalam phonebook perangkat mobile untuk memasukkan nomor telpon, email ataupun alamat URL.

Tabel 1. Format TextField

|  |  |
| --- | --- |
| Format | Keterangan |
| TextField.ANY | Pengguna dapat memasukkan karakter/input apapun kedalam  TextField |
| TextField.EMAILADDR | Hanya mengijinkan input berupa alamat email tetapi implementasi vendor memungkinkan akses untuk lookup pada phonebook perangkat mobile |
| TextField.NUMERIC | Hanya mengijinkan input berupa bilangan bulat |
| TextField.URL | Hanya mengijinkan input berupa alamat URL (Universal Resource Locator) |
| TextField.DECIMAL | Memungkinkan input berupa bilangan desimal/pecahan |
| TextField.PHONENUMBER | Memungkinkan masukan berupa nomor telpon. Implementasi vendor dapat saja membuat format ini sama dengan format NUMERIC dan mungkin juga mendapatkan akses untuk lookup pada phonebook perangkat mobile |

Dalam TextField juga dapat didefinisikan perilaku/sifat TextField setiap kali user memasukkan input/karakter. Jenis‐jenisnya antara lain PASSWORD,UNEDITABLE,SENSITIVE,NON\_PREDICTIVE, INITIAL\_CAPS\_WORD dan INITIAL\_CAPS\_SENTENCE. Suatu obyek t extField dapat memiliki salah satu atau lebih dari perilaku ini. Jika hal ini dilakukan perlu diperhatikan kompatibilitas antar perilaku selain itu juga implementasi antar perangkat bergerak.

|  |  |
| --- | --- |
| **Perilaku** | **Keterangan** |
| TextField.PASSWORD | Masukan karakter disamarkan dengan karakter khusus misalnya asteriks (\*) |
| TextField.UNEDITABLE | Teks yang terdapat pada TextField tidak diperbolehkan untuk diubah |
| TextField.SENSITIVE | Teks yang dimasukkan ke dalam TextField bersifat rahasia dan tidak boleh disimpan di memori |
| TextField.NON\_PREDICTIVE | Teks yang dimasukkan tidak terdapat dalam kamus sehingga tidak cocok bila digunakan prediksi input |
| TextField.INITIAL\_CAPS\_WORD | Huruf pertama dari masing‐masing kata pada teks masukandari user harus menjadi huruf besar |
| TextField.INITIAL\_CAPS\_SENTENCE | Huruf pertama dari keseluruhan teks yang dimasukkan oleh user diganti menjadi huruf besar |

Tabel 2. Perilaku TextField terhadap masukan User

1. **StringItem**

StringItem merupakan komponen untuk menampilkan tulisan pada layar. StringItem ini membungkus obyek String supaya dapat diperlakukan sama dengan turunan obyek Item yang lainnya. Obyek String sendiri dapat langsung dimasukkan ke dalam Form dengan method int append(String) tapi yang terjadi sebenarnya adalah obyek tersebut dibungkus dengan menggunakan StringItem. Jika beberapa obyek StringItem dimasukkan ke dalam Form secara berurutan maka semua obyek tersebut akan di gabungkan/konkatenasi.

Ada tiga jenis tampilan dari StringItem yaitu PLAIN, BUTTON dan HYPERLINK. Secara default suatu obyek StringItem memiliki penampilan PLAIN. Jika menggunakan jenis BUTTON atau HYPERLINK maka perlu didefinisikan juga obyek Command yang akan merespon user jika menekan StringItem tersebut. Perbedaan antara BUTTON dan HYPERLINK terletak pada tampilannya. Dengan menggunakan BUTTON akan tiampilkan StringItem dengan tampilan seperti tombol sedangkan HYPERLINK akan menampilkan StringItem seperti link pada web. Obyek Command yang didefinisikan diatas dapat menjadi aksesibel pada keseluruhan Form ataupun hanya pada StringItem yang memakainya.

1. **Command**

Command adalah obyek yang digunakan sebagai antarmuka respon user perangkat mobile. Untuk mendapatkan respon dari user Command sering diasosiasikan dengan softkey, tombol atau mungkin area pada touch screen. Command pada umumnya digunakan untuk navigasi antar obyek Displayable dan untuk melakukan proses atas masukan dari user.

Command merupakan komponen penting dalam penanganan even atau event handling pada J2ME. Penanganan even menggunakan Command sama seperti pada Button/JButton di J2SE yaitu user mendefinisikan obyek Command lalu menambahkan listener pada obyek tersebut. Kelas untuk listener Command adalah CommandListener.

Command dapat dimasukkan ke dalam obyek Displayable mengunakan method sebagai berikut

public void addCommand(Command cmd)

Sedangkan penghapusan dengan menggunakan method sebagai berikut

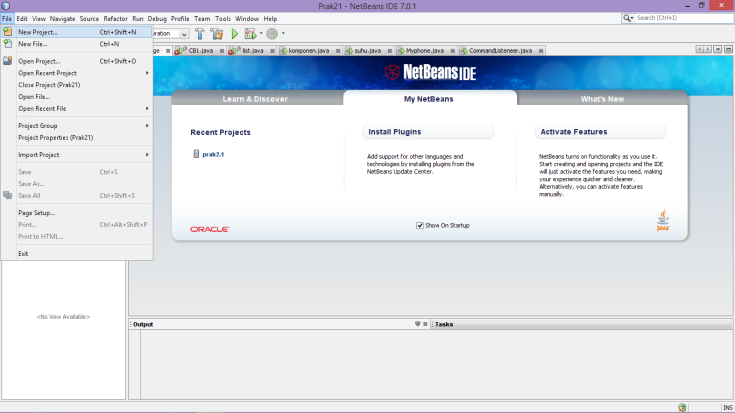
public void removeCommand(Command cmd)

Dalam satu obyek Displayable dapat terdapat lebih dari dua Command.Setiap obyek Command mempunyai tiga atribut yaitu label, type dan priority. label pada Command digunakan untuk menampilkan nama Command pada layar perangkat mobile.

type pada Command digunakan untuk menentukan fungsi dari Command tersebut. type ini menentukan softkey mana yang akan merepresentasikannya. Jenis‐jenis type dari Command antara lain BACK,CANCEL,EXIT,HELP,OK, SCREEN,ITEM dan STOP. Jenis ITEM biasanya diasosiasikan dengan sebuah komponen turunan Item yang terdapat pada Form. Jenis SCREEN diasosiasikan untuk semua komponen yang terdapat pada/dalam turunan obyek Screen.

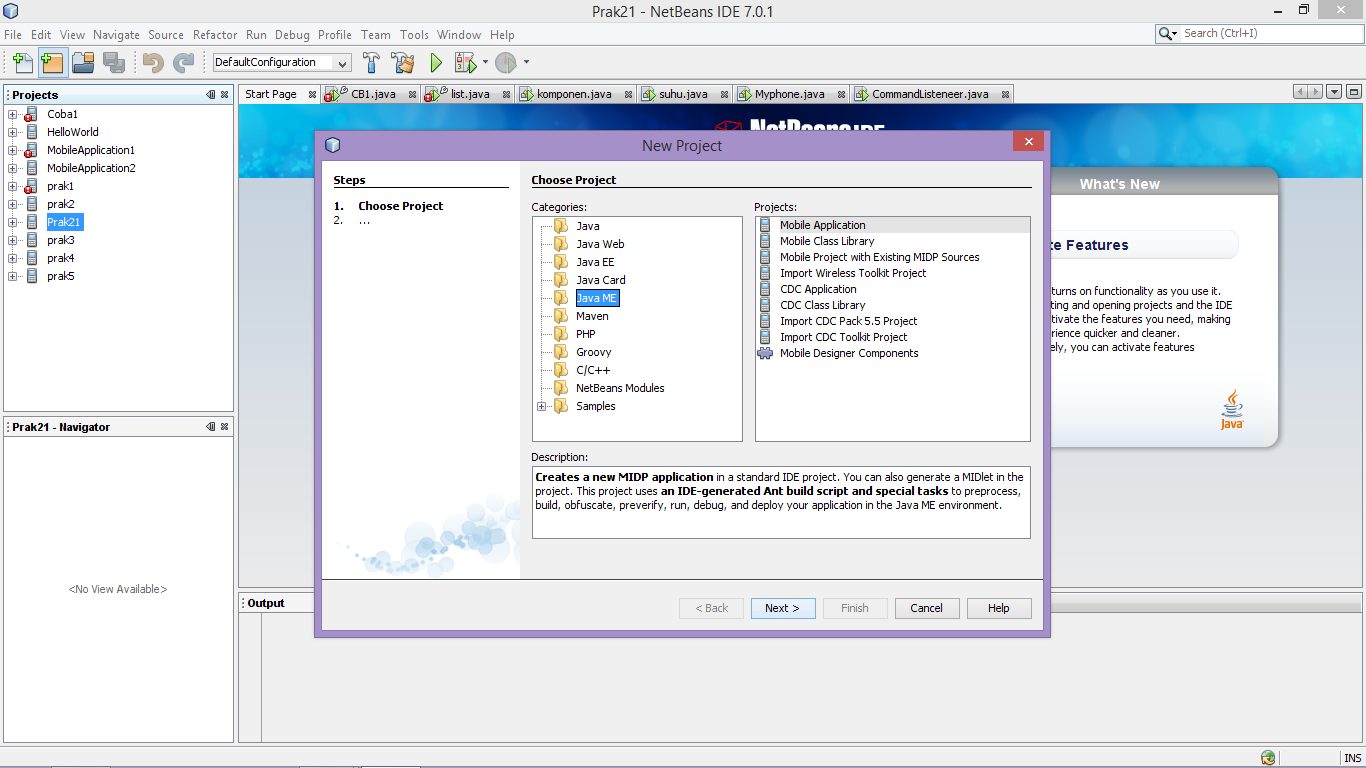
priority menentukan urutan prioritas penampilan suatu Command pada layar. Semakin kecil priority maka Command itu akan lebih mudah diakses dibandingkan dengan Command yang nilai priority ‐nya lebih besar.

1. **Kegiatan Praktikum**
2. Buatlah project baru pada netbeans Klik File – New Project.



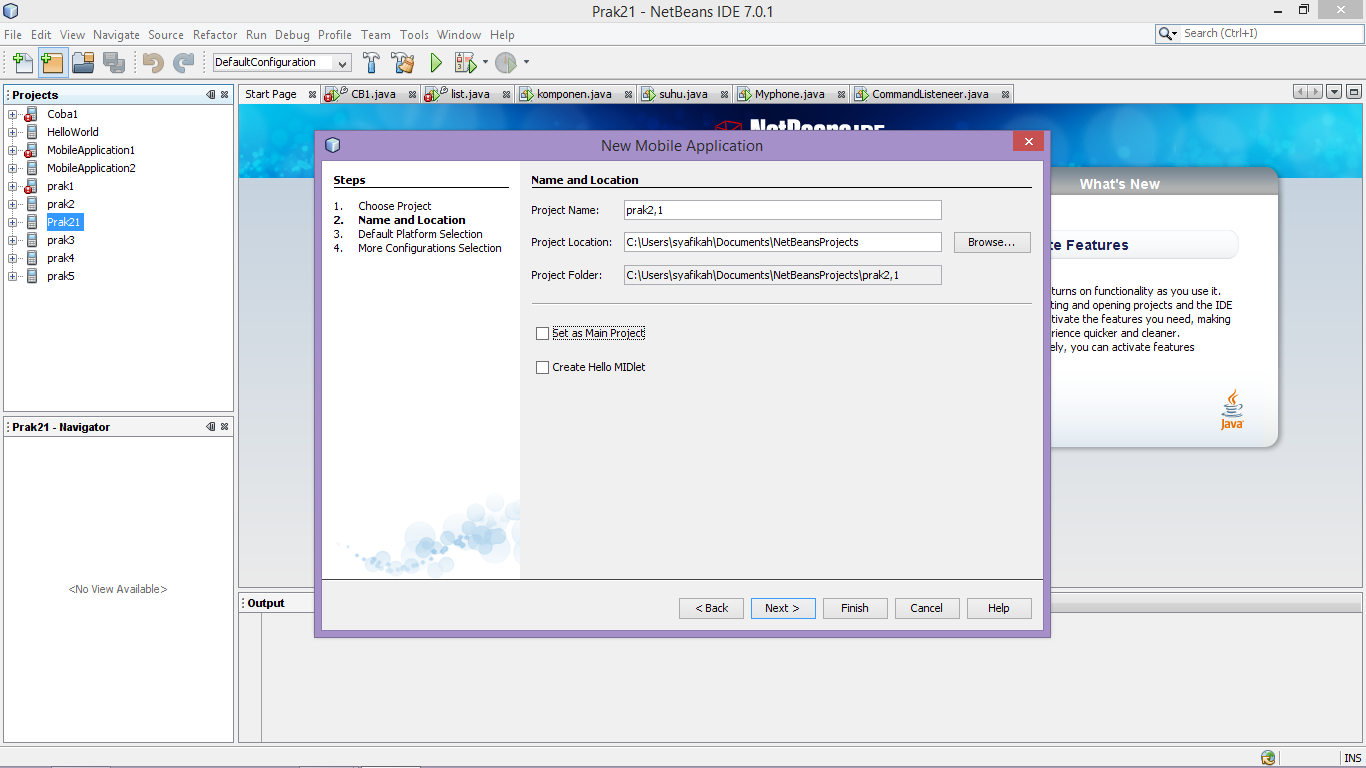
Gambar 3. Membuat Project Baru

1. Kemudian pilih Java Me – Pilih Mobile Application – dan Klik Next.



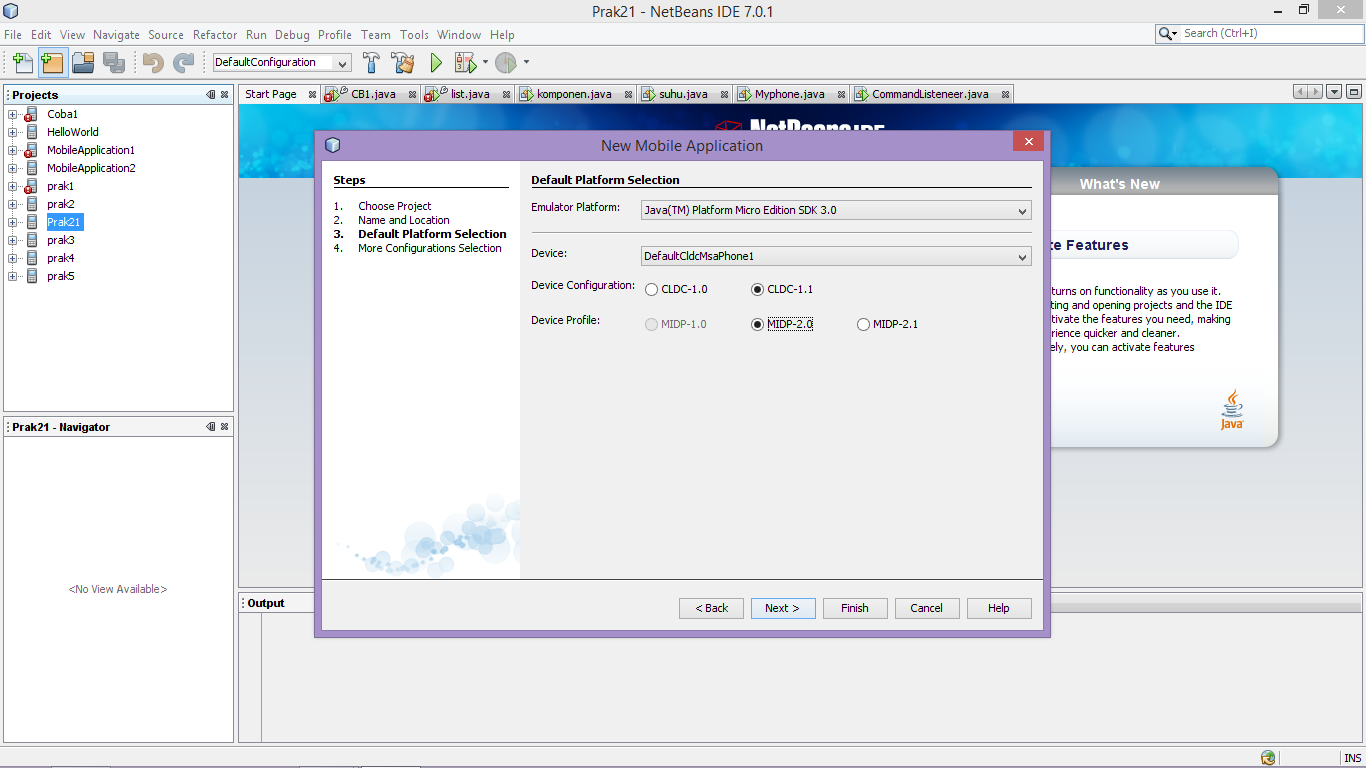
Gambar 4. Memilih Java Me

1. Setelah itu akan muncul Name and Location, kemudian isikan Project Name dengan “Prak21” atau sesuai dengan yang di inginkan dan pilih Project Location untuk memilih dimana project akan di buat di simpan (Hilangkan tanda pada "Create Hello MIDlet", kita akan membuat MIDlet kita sendiri nantinya). Selanjutnya Klik Next.



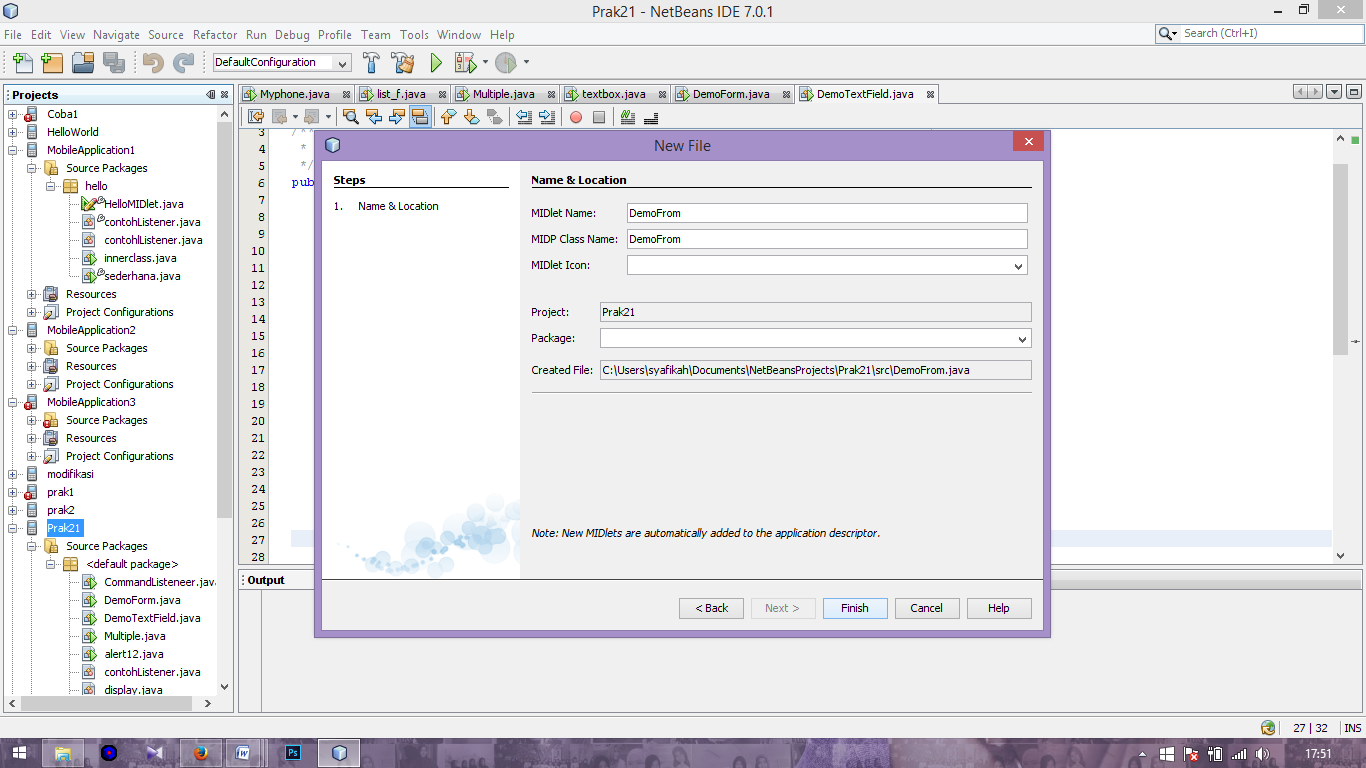
Gambar 5. Membuat Nama Project Baru

1. Dan selanjutnya juga akan muncul Form Default Platfrom Selection pada Divice pilih salah satu objek yang ada pada combobox, Divice Configuration pilih CLDC – 1.1 dan pada Divice Profil pilih MIDP – 2.0 kemudian Klik – Finish.



Gambar 6. Memilih Tampil Aplikasi Java Me

1. Maka selanjutnya akan muncul layar utama pada netbeans pada Source Packages di “prak21” atau pada objek yang di buat tadi Klik kanan – Pilih New – dan Pilih MIDlet dan isikan MIDlet Name dengan nama “Demoform ” atau sesuai yang diinginkan lalu Klik Finish .



Gambar 7. Membuat Nama Midlet Baru

1. Dan kemudian muncul Form dari MIDlet maka masukan listing program sebagai berikut :
2. Listing program menggunakan Objek Form

import javax.microedition.midlet.\*;

import javax.microedition.lcdui.\*;

public class DemoForm extends MIDlet implements CommandListener{

private Display display;

private Form form;

private TextField tf;

private Command cmdKeluar;

private Command cmdOk;

public DemoForm (){

display = Display.getDisplay(this);

cmdKeluar = new Command("Keluar", Command.EXIT, 1);

cmdOk = new Command ("OK", Command.OK,1);

}

public void startApp() {

form = new Form("Demo Form");

tf = new TextField ("Masukan Nomor Telepon ","", 15 ,TextField.PHONENUMBER);

form.append((Item) tf);

form.addCommand(cmdKeluar);

form.addCommand(cmdOk);

form.setCommandListener(this);

display.setCurrent(form);}

public void pauseApp() {

}

public void destroyApp(boolean unconditional) {

}

public void commandAction(Command c, Displayable d) {

if (c == cmdKeluar){

destroyApp(true);

notifyDestroyed();

}

else if (c == cmdOk){

Alert info = new Alert ("Informasi");

info.setType(AlertType.INFO);

String text = tf.getString();

info.setString("Nomor Telepon: \""+

text + "\"");

info.setTimeout(Alert.FOREVER);

display.setCurrent(info,form);

}

}

}

1. Listing Program menggunaka objek textfield

import javax.microedition.midlet.\*;

import javax.microedition.lcdui.\*;

public class DemoTextField extends MIDlet implements CommandListener{

private Display display;

private Form form;

private TextField nama, password, telp, email, URL;

private final Command cmdKeluar = new Command ("Keluar", Command.EXIT,1);

public DemoTextField(){

display = Display.getDisplay(this);

}

public void startApp() {

form = new Form("Demo TextField");

nama = new TextField("Nama :",null,25,TextField.ANY);

password = new TextField("Password :",null,50,TextField.PASSWORD);

email = new TextField("Email :",null,50,TextField.EMAILADDR);

URL = new TextField("URL :",null,50,TextField.URL);

form.append(nama);

form.append(password);

form.append(email);

form.append(URL);

form.addCommand(cmdKeluar);

form.setCommandListener(this);

display.setCurrent(form);

}

public void pauseApp() {

}

public void destroyApp(boolean unconditional) {

}

public void commandAction(Command c, Displayable d) {

if (c == cmdKeluar){

destroyApp(true);

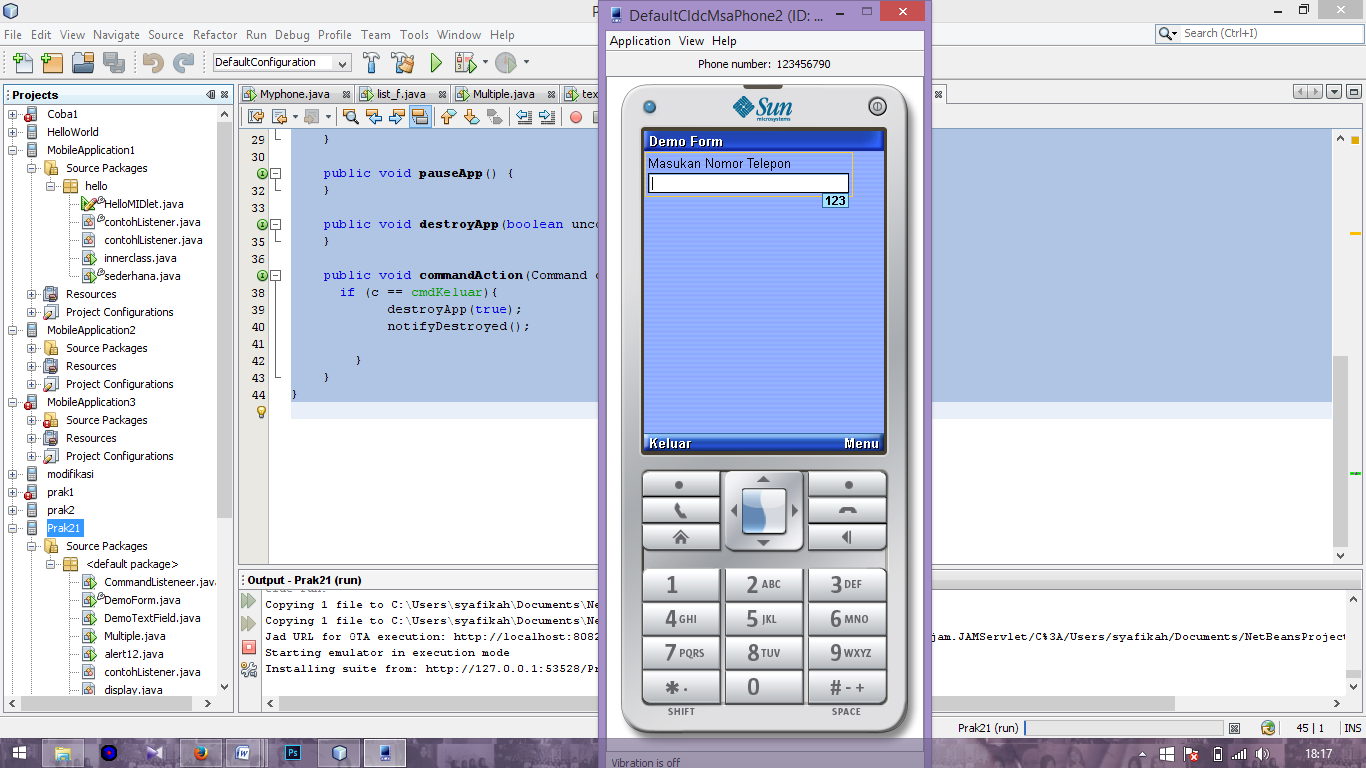
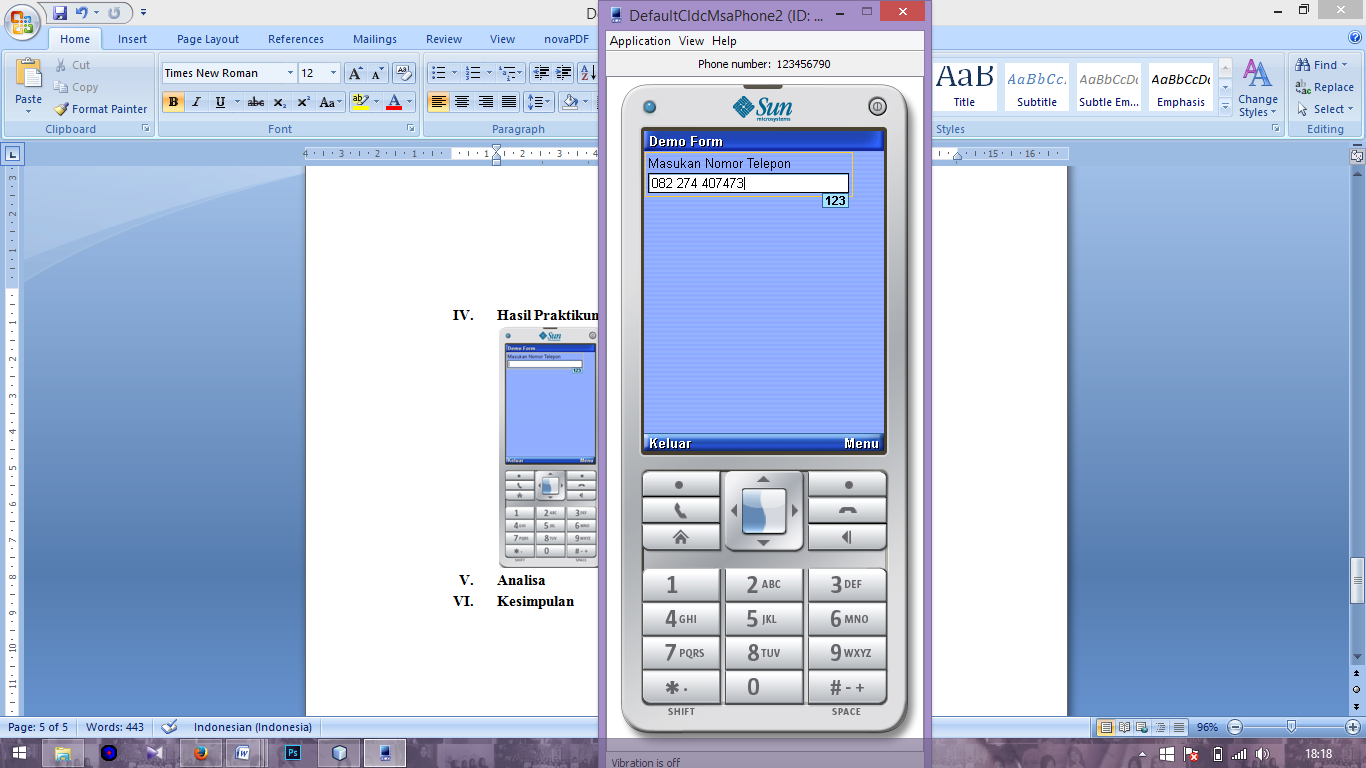
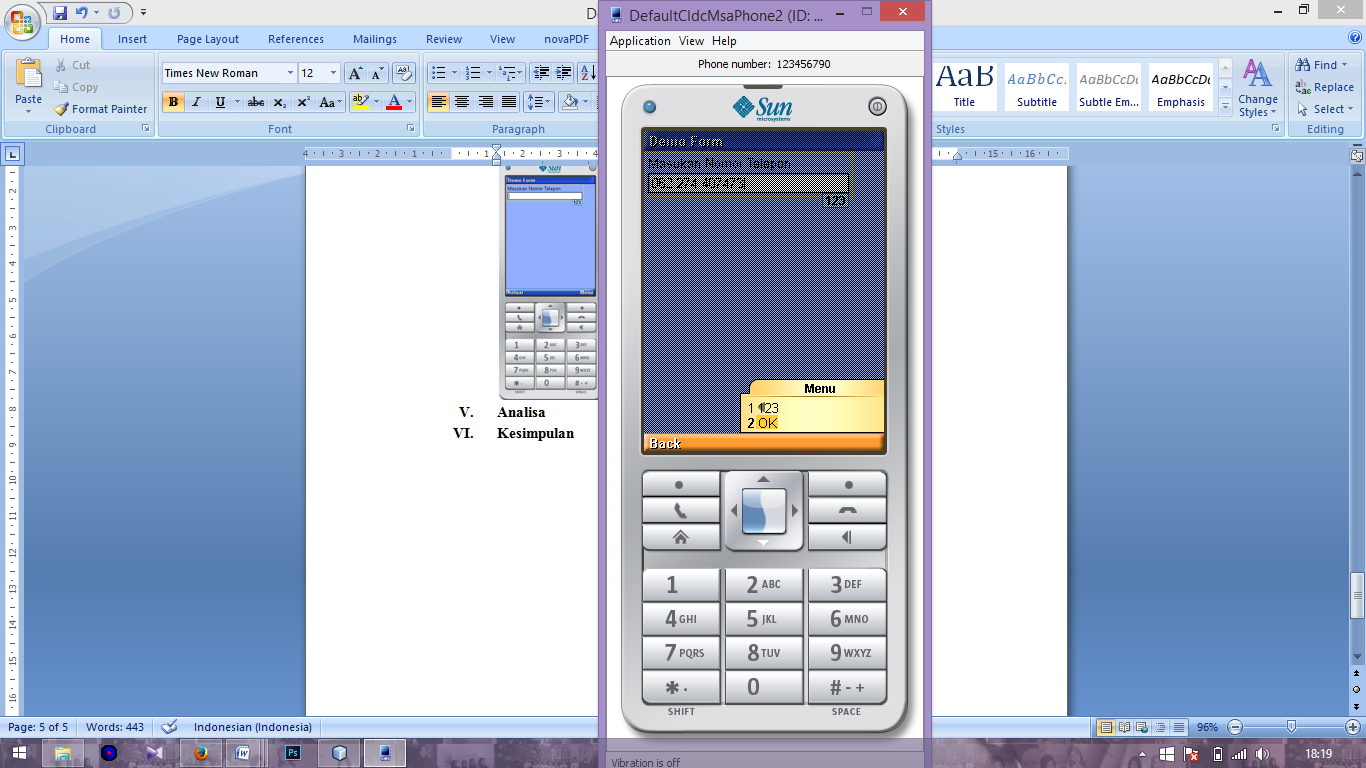
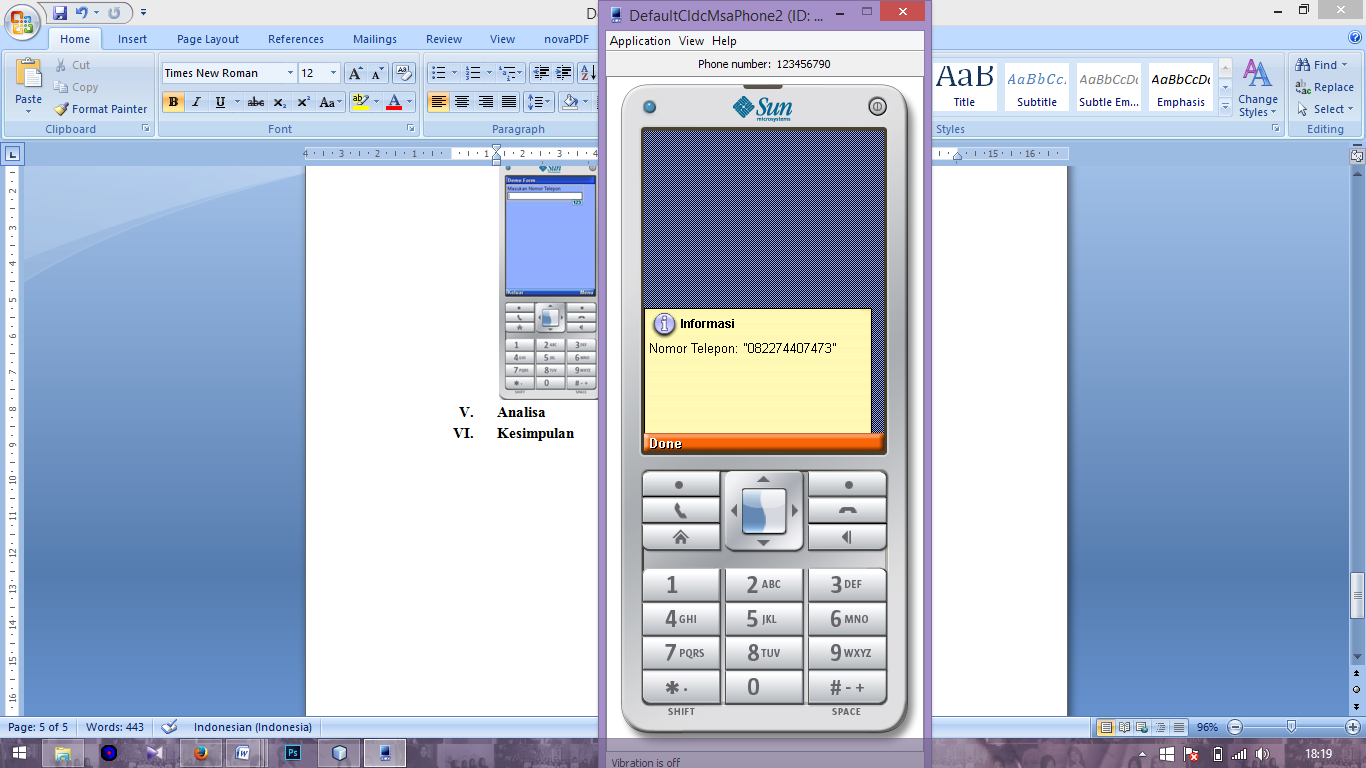
notifyDestroyed();

}

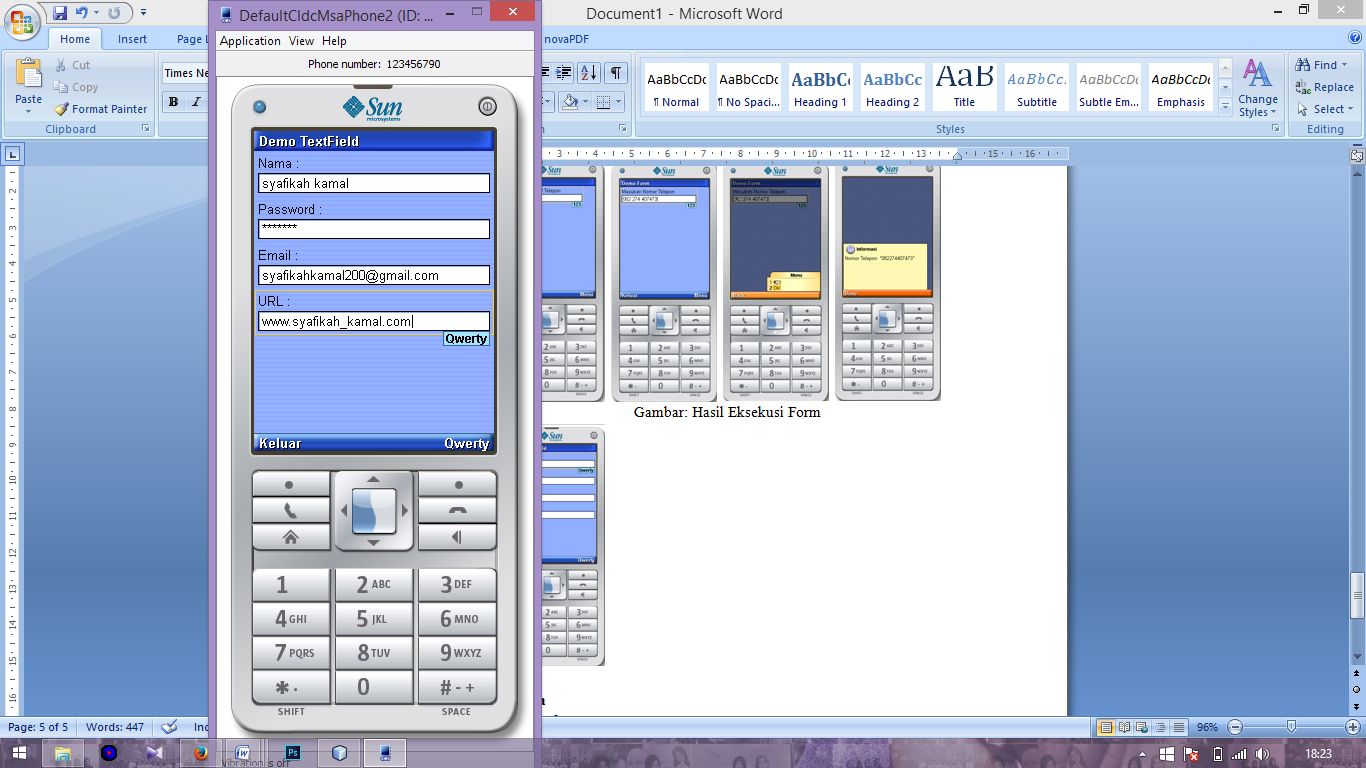
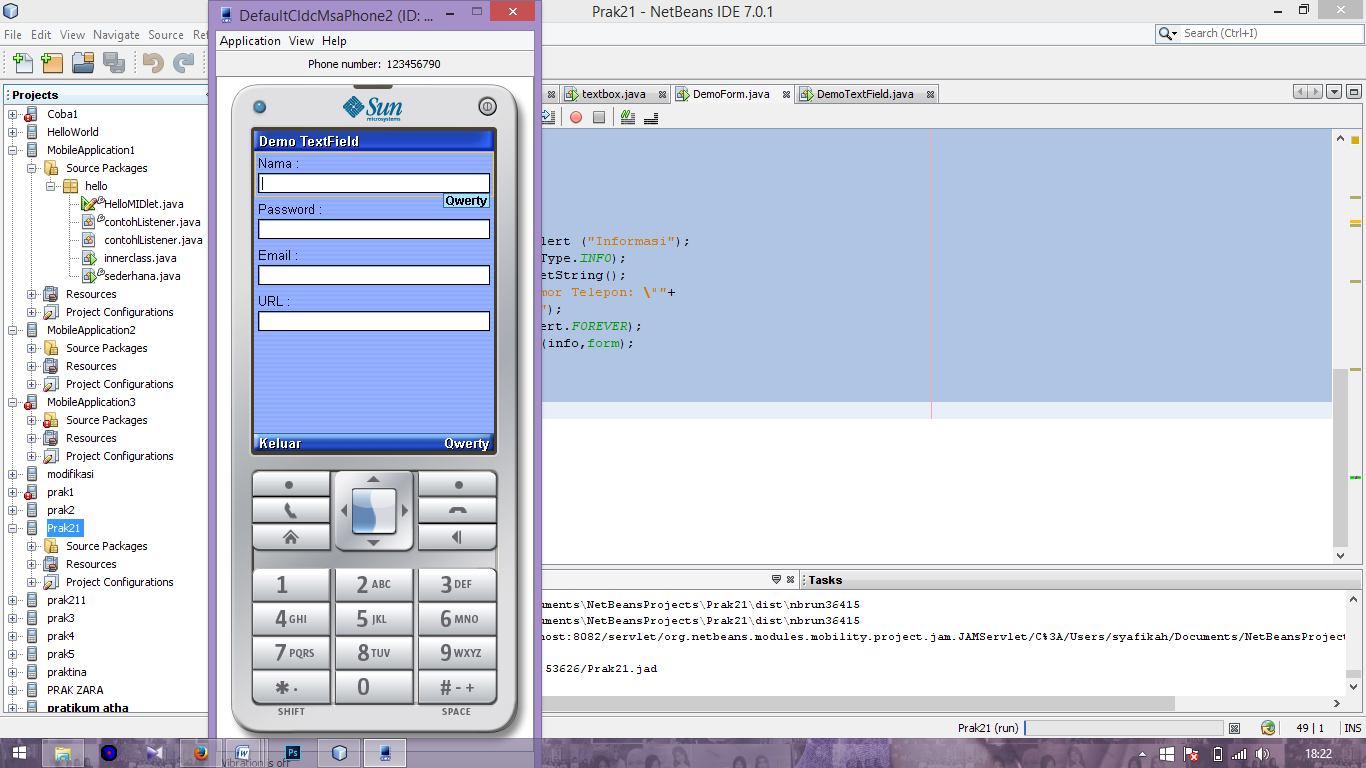
}

}

1. **Hasil Praktikum**

**   **

Gambar 8. Hasil Eksekusi Form



Gambar 9. Hasil Eksekusi TextField

1. **Analisa**
2. Analisa Listing Program Form

import javax.microedition.lcdui.\*;

Untuk menampilkan program pada layar display emulator atau sama juga dengan GUI (Grafical User Interface).

public class DemoForm extends MIDlet implements CommandListener {

private Display display;private Form form; private TextField tf;private Command cmdKeluar; private Command cmdOk;

Merupakan membuat class dengan nama DemoForm dari turunan midlet yang di implementasikan kedalan commandlistener dann juga membuat sebuah class yang bisa diakses oleh kelas itu sendiri, dengan modifier private di mana kelas yang tidak dapat diakses sama sekali oleh kelas lain bahkan juga tidak dapat diturunkan.

public DemoForm (){

display = Display.getDisplay(this);

cmdKeluar = new Command("Keluar", Command.EXIT, 1);

cmdOk = new Command ("OK", Command.OK,1);}

adalah konsuktor dari class midlet dan menampilkan object, membuat beberapa buah aksi (Command) dengan label string dan command type exit dan ok dengan command priority 1.

form = new Form("Demo Form");

tf = new TextField ("Masukan Nomor Telepon ","",15 ,TextField.PHONENUMBER);

form.append((Item) tf); form.addCommand(cmdKeluar); form.addCommand(cmdOk); form.setCommandListener(this); display.setCurrent(form);

Berfungsi untuk membuat form baru denga nama Demo Form dan juga membuat textfield baru dengan nama variabel tf dengan string title “Masukan Nomor Telepon”, dan text maxsize nya 15 dan juga type textfield yaitu phonenumber dan juga menambahkan kan item pada posisi terakhir di dalam form dan menambahkan command kedalam form, dan menambahkan command agar berfungsi didalam form juga mengaktifkan form.

if (c == cmdKeluar){

destroyApp(true);

notifyDestroyed();}

berfungsi sebagai jika di klik commad keluar pada form maka aksi yang di tampilkan adalah keluar dari form.

else if (c == cmdOk){

Alert info = new Alert ("Informasi");

info.setType(AlertType.INFO);

String text = tf.getString();

info.setString("Nomor Telepon: \""+ text + "\"");

info.setTimeout(Alert.FOREVER); display.setCurrent(info,form);}

berfungsi sebagai dan jika di klik commad ok pada form maka aksi yang di tampilkan adalah akan muncul alert dengan nama variabel info dengan title alert Informasi dan type alert informasi dan juga memanggil string text dari textfield yaitu tf kemudian di masukan ke dalam alert info dan di tampilkan dengan waktu selamanya dan juga di aktif kan kedalam form dan.

1. Analisa Listing Program TextField

private Display display;

private Form form;

private TextField nama, password, telp, email, URL;

private final Command cmdKeluar = new Command ("Keluar", Command.EXIT,1);

Merupakan method atau variabel hanya dapat diakses dari dalam kelas tersebut. modifier ini biasanya digunakan untuk kelas dari objek dan mendeklarasikan type objek dan juga variabel, private final bertujuan unutuk modifier private ini menunjukkan bahwa suatu class denga type command tidak bisa diubah lagi nilainya (menjadi konstanta).

form = new Form("Demo TextField");

nama = new TextField("Nama ",null,25,TextField.ANY);

password = new TextField("Password :",null,50,TextField.PASSWORD);

email = new TextField("Email :",null,50,TextField.EMAILADDR);

URL = new TextField("URL :",null,50,TextField.URL);

form.append(nama);form.append(password);

form.append(email);form.append(URL);

form.addCommand(cmdKeluar); form.setCommandListener(this); display.setCurrent(form);

Berfungsi untuk membuat form baru dengan nama form dan title Demo TextField dan juga membuat textfield baru dengan nama variabel masing – masing dengan string title dan text maxsize masing – masing berbeda dan juga type textfield yaitu ANY, EMAILADDR, NUMERIC, PHONENUMBER, URL dan DECIMAL. dan juga menambahkan kan textfield pada posisi terakhir di dalam form dan menambahkan command kedalam form, dan menambahkan command agar berfungsi didalam form juga mengaktifkan form.

1. **Kesimpulan**

Form adalah komponen turunan Screen yang berfungsi sebagai tempat untuk menampung (container) bagi komponen lain. Fungsi Form pada J2ME sama seperti Form pada HTML. Form selain memiliki obyek Item biasanya juga mempunyai obyek Command. Fungsi Command pada Form ini juga sama dengan fungsi Button pada Form di HTML.

TextField adalah komponen turunan Item yang menampung teks dan mengedit teks tersebut. TextField mempunyai ukuran maksimum, label dan format input. Beberapa vendor mengimplementasikan TextField sehingga ukuran TextField dapat secara otomatis membesar sesuai dengan input string yang dimasukkan. Masukan/input pada TextField dapat di format sesuai kebutuhan. Ada enam jenis format yang dikenali oleh TextField yaitu ANY, EMAILADDR, NUMERIC, PHONENUMBER, URL dan DECIMAL.