

Seri User Interface

Tipe Masukan pada TextField

Chocolove Mic

chocolove_mic@yahoo.co.uk

<http://www.mycgiserver.com/~chocolove2003>

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarluaskan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

Pada bagian sebelumnya telah banyak digunakan class TextField untuk menangani masukan dari user. Pada bagian ini akan dibahas lebih lanjut tentang class TextField terutama tentang pengaturan tipe masukan. Dengan pengaturan ini aplikasi dapat membatasi user untuk memasukkan karakter ke dalam kolom input.

Nilai Properti Constrains pada TextField

Constrains adalah properti yang dimiliki oleh class TextField. Nilai yang dimiliki oleh properti ini adalah :

Nilai Constrain	Fungsi dan Keterangan	Nilai
TextField.ANY	Mengizinkan seluruh karakter untuk diketikkan ke dalam kolom masukan.	0
TextField.EMAILADDR	Kotak input yang hanya bisa diisi dengan aturan penulisan alamat email yang valid.	1
TextField.NUMERIC	Kotak input hanya dapat diisi dengan nilai bilangan bulat.	2
TextField.PHONENUMBER	Kotak input yang dapat diisi dengan nomor telephone.	3
TextField.URL	Kotak input akan hanya bisa diisi dengan URL yang valid.	4
TextField.PASSWORD	Kotak input untuk keperluan pengisian password.	65536

Untuk melihat nilai dari properti di atas dapat digunakan dengan program di bawah ini :

TipeTextfieldMidlet01.java
<pre>import javax.microedition.midlet.*; import javax.microedition.lcdui.*; public class TipeTextfieldMidlet01 extends MIDlet implements CommandListener { private Form form;</pre>

```
private Command cmdKeluar;
private Display display;
private TextField nilaiANY;
private TextField nilaiEMAIL;
private TextField nilaiNUMERIC;
private TextField nilaiPHONE;
private TextField nilaiURL;
private TextField nilaiPASSWD;
private String nilaiTipe;

public TipeTextfieldMidlet01() {
    cmdKeluar = new Command("Keluar", Command.EXIT, 1);

    nilaiANY = new TextField("ANY", "", 50, TextField.ANY);
    nilaiTipe = String.valueOf(TextField.ANY);
    nilaiANY.setString(nilaiTipe);

    nilaiEMAIL = new TextField("EMAIL", "", 50, TextField.ANY);
    nilaiTipe = String.valueOf(TextField.EMAILADDR);
    nilaiEMAIL.setString(nilaiTipe);

    nilaiNUMERIC = new TextField("NUMERIC", "", 50, TextField.ANY);
    nilaiTipe = String.valueOf(TextField.NUMERIC);
    nilaiNUMERIC.setString(nilaiTipe);

    nilaiPHONE = new TextField("PHONE", "", 50, TextField.ANY);
    nilaiTipe = String.valueOf(TextField.PHONENUMBER);
    nilaiPHONE.setString(nilaiTipe);

    nilaiURL = new TextField("URL", "", 50, TextField.ANY);
    nilaiTipe = String.valueOf(TextField.URL);
    nilaiURL.setString(nilaiTipe);

    nilaiPASSWD = new TextField("PASSWORD", "", 50, TextField.ANY);
    nilaiTipe = String.valueOf(TextField.PASSWORD);
    nilaiPASSWD.setString(nilaiTipe);

    form = new Form("Tipe TextField");

    form.append(nilaiANY);
    form.append(nilaiEMAIL);
    form.append(nilaiNUMERIC);
    form.append(nilaiPHONE);
    form.append(nilaiURL);
    form.append(nilaiPASSWD);

    form.addCommand(cmdKeluar);
    form.setCommandListener(this);
}

public void startApp() {
    display = Display.getDisplay(this);
    display.setCurrent(form);
}

public void pauseApp() {
}

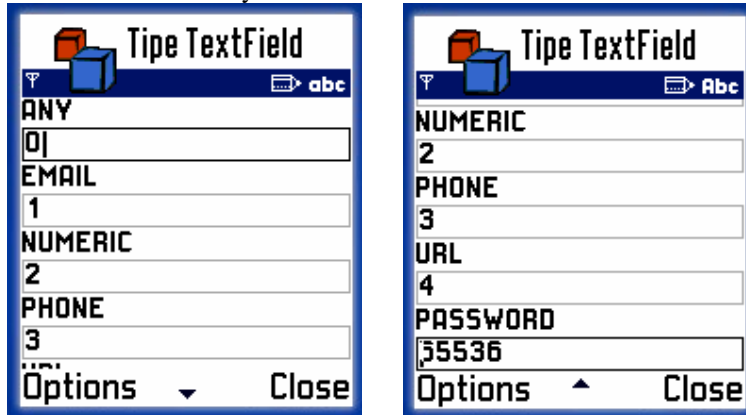
public void destroyApp(boolean unconditional) {
}

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

```
}  
}  
}  
}
```

Sumber :-

Berikut adalah hasilnya :



Bisa dilihat nilai-nilai dari properti-properti yang telah dituliskan pada tabel di atas adalah benar. Dengan begitu untuk menentukan tipe masukan dari textfield dapat digantikan dengan menuliskan nilai dari properti tersebut. Yaitu seperti di bawah ini :

```
nilaiANY = new TextField("ANY", "", 50, TextField.ANY);
```

Menjadi :

```
nilaiANY = new TextField("ANY", "", 50, 0);
```

Contoh lain :

```
nilaiEMAIL = new TextField("EMAIL", "", 150, TextField.EMAIL);
```

Menjadi :

```
nilaiEMAIL = new TextField("EMAIL", "", 150, 1);
```

Penggunaan Properti Constrain

Pada bahasan di atas, contoh program yang diberikan hanya untuk menampilkan nilai dari properti constrain pada TextField. Pada bagian ini akan diberikan contoh penggunaanya. :

```
TipeTextfieldMidlet02.java  
import javax.microedition.midlet.*;  
import javax.microedition.lcdui.*;  
  
public class TipeTextfieldMidlet02 extends MIDlet implements CommandListener {  
  
    private Form form;  
    private Command cmdKeluar;  
    private Display display;  
    private TextField nilaiANY;  
    private TextField nilaiEMAIL;
```

```
private TextField nilaiNUMERIC;
private TextField nilaiPHONE;
private TextField nilaiURL;
private TextField nilaiPASSWD;
private String nilaiTipe;

public TipeTextfieldMidlet02() {
    cmdKeluar = new Command("Keluar", Command.EXIT, 1);

    nilaiANY = new TextField("ANY", "", 50, TextField.ANY);
    nilaiEMAIL = new TextField("EMAIL", "", 50, TextField.EMAILADDR);
    nilaiNUMERIC = new TextField("NUMERIC", "", 50, TextField.NUMERIC);
    nilaiPHONE = new TextField("PHONE", "", 50, TextField.PHONENUMBER);
    nilaiURL = new TextField("URL", "", 50, TextField.URL);
    nilaiPASSWD = new TextField("PASSWORD", "", 50, TextField.PASSWORD|0);

    form = new Form("Tipe TextField");

    form.append(nilaiANY);
    form.append(nilaiEMAIL);
    form.append(nilaiNUMERIC);
    form.append(nilaiPHONE);
    form.append(nilaiURL);
    form.append(nilaiPASSWD);

    form.addCommand(cmdKeluar);
    form.setCommandListener(this);
}

public void startApp() {
    display = Display.getDisplay(this);
    display.setCurrent(form);
}

public void pauseApp() {
}

public void destroyApp(boolean unconditional) {
}

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

Sumber :-

Efek dari pendeklarasian tipe input ini, salah satunya adalah untuk tipe input yang menggunakan properti constrain `TextField.NUMERIC` dan `TextField.PHONENUMBER` hanya bisa diisi dengan angka. Sedangkan untuk yang menggunakan `TextField.ANY` dapat diisi apa saja.

Untuk input yang digunakan untuk mengisi password ada perbedaan dibandingkan tipe input yang lain yaitu sebagai berikut :

```
nilaiPASSWD = new TextField("PASSWORD", "", 50, TextField.PASSWORD|0);
```

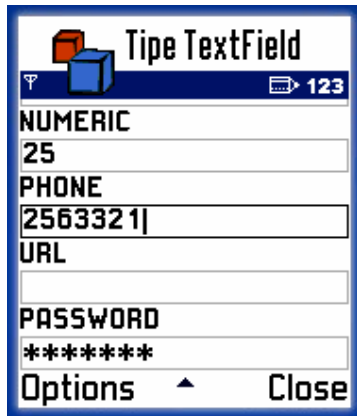
Tipe input ini digunakan properti constrain yang merupakan gabungan `TextField.PASSWORD` dan `TextField.ANY`. Artinya input ini digunakan untuk password dan password tersebut

terdiri abjad, angka atau karakter lain. Penulisannya untuk menggunakan aturan ini adalah seperti contoh di atas atau sebagai berikut :

```
TextField.PASSWORD|TextField.ANY
```

Apabila password yang diinginkan hanya berisi atas angka saja maka deklarasi yang digunakan adalah sebagai berikut :

```
TextField.PASSWORD|TextField.NUMBER
```



Catatan :

Properti `TextField.EMAIL` dan `TextField.URL` tidak dapat memeriksa ke-valid-an suatu alamat email atau URL yang dimasukkan pada kolom input secara otomatis.

Catatan Penulis :

Penulis adalah pemula dalam mempelajari Java, jadi mohon masukannya bagi pembaca yang menemukan kesalahan konsep atau asumsi yang digunakan penulis.