

PEMOGRAMAN APLIKASI BERGERAK (Java ME)



YOANNITA



LIST



OBJEK LIST

- ❖ Objek List dapat dibuat dengan tipe Pilihan :
 - Choice.EXCLUSIVE,
 - Choice.MULTIPLE, dan
 - Choice.IMPLICIT.
- ❖ Pilihan jenis Choice.POPUP tidak diperbolehkan pada Objek List.

LIST

- ❖ Bentuk constructor dari kelas list adalah sebagai berikut :

```
List ( String title, int listType)
```

```
List ( String title, int listType, String [] choices,  
Image[] images)
```

- ❖ Pada bentuk constructor pertama diatas, kita membuat sebuah objek List kosong yang hanya memiliki judul dan tipe saja. Artinya, item-item di dalam nya dapat kita manipulasi sendiri melalui kode program, yaitu dengan menggunakan method-method pada tabel berikut →

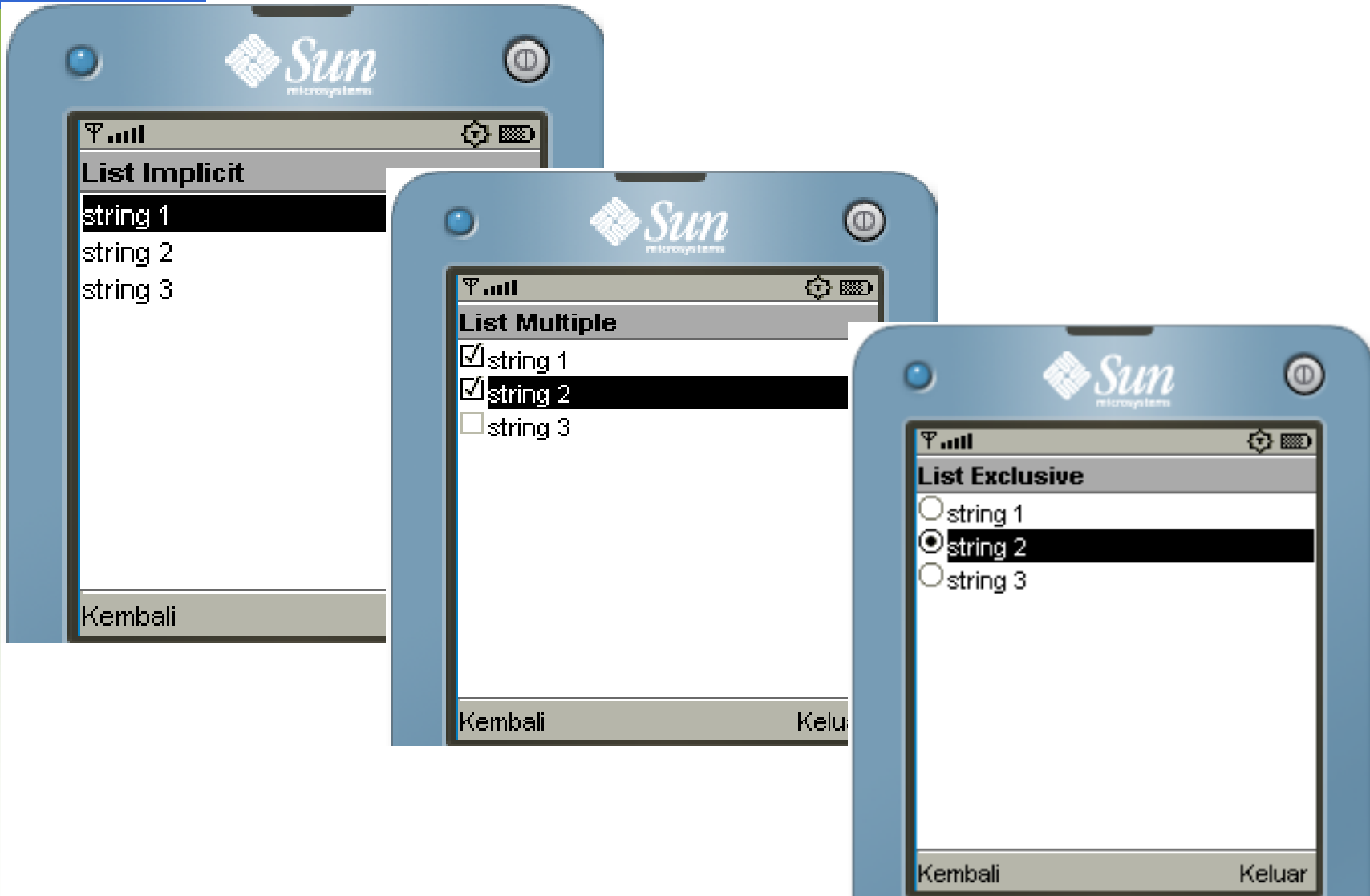


Nama Method	Kegunaan
append()	Menambah item dan menempatkannya di posisi terakhir.
insert()	Menambah item dan menempatkannya di posisi sebelum posisi item aktif (item yang sedang dipilih)
delete()	Menghapus item yang sedang dipilih
deleteAll()	Menghapus semua item yang terdapat di dalam objek List

Tipe-tipe pada List

Tipe	Keterangan
EXCLUSIVE	List ini berupa radio button, yakni user hanya dapat memilih satu pilihan.
MULTIPLE	List ini berupa list yang di dalamnya terdapat check box. Disini, user diizinkan untuk memilih beberapa (lebih dari satu) pilihan.
IMPLICIT	List ini berupa list standar yang umumnya banyak digunakan.

Perbandingan LIST Implicit, Multiple dan Exclusive



Contoh Program (1)

```
public class LatihanListMIDlet extends MIDlet implements CommandListener {  
    private Display display = Display.getDisplay(this);  
    private Command cmdExit, cmdBack;  
    private List myList, listExclusive, listMultiple, listImplicit;  
  
    public LatihanListMIDlet() {  
        cmdExit = new Command ("Keluar", Command.EXIT, 1);  
        cmdBack = new Command ("Back", Command.SCREEN, 3);  
  
        myList = new List("Judul List", List.IMPLICIT);  
        myList.append("EXCLUSIVE", null);  
        myList.append("MULTIPLE", null);  
        myList.append("IMPLICIT", null);  
        myList.addCommand(cmdBack);  
        myList.addCommand(cmdExit);  
        myList.setCommandListener(this);  
    }  
}
```



Contoh Kode Program

```
listExclusive = new List("List EXCLUSIVE", List.EXCLUSIVE);  
listExclusive.append("string 1", null);  
listExclusive.append("string 2", null);  
listExclusive.append("string 3", null);  
listExclusive.addCommand(cmdBack);  
listExclusive.addCommand(cmdExit);  
listExclusive.setCommandListener(this);
```

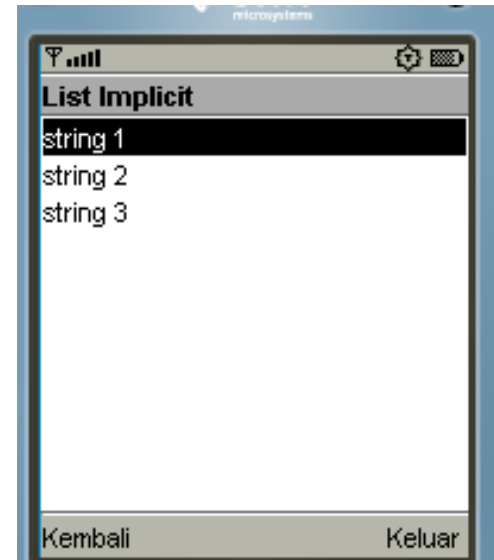


```
listMultiple = new List("List MULTIPLE", List.MULTIPLE);  
listMultiple.append("string 1", null);  
listMultiple.append("string 2", null);  
listMultiple.append("string 3", null);  
listMultiple.addCommand(cmdBack);  
listMultiple.addCommand(cmdExit);  
listMultiple.setCommandListener(this);
```



Contoh Kode Program

```
listImplicit = new List("List IMPLICIT", List.IMPLICIT);  
listImplicit.append("string 1", null);  
listImplicit.append("string 2", null);  
listImplicit.append("string 3", null);  
listImplicit.addCommand(cmdBack);  
listImplicit.addCommand(cmdExit);  
listImplicit.setCommandListener(this);  
  
}
```



Event Handling

```
public void commandAction(Command c, Displayable d) {  
    if (c== List.SELECT_COMMAND){  
        switch ( myList.getSelectedIndex() )  
        {  
            case 0:  
                display.setCurrent(listExclusive);  
                break;  
            case 1:  
                display.setCurrent(listMultiple);  
                break;  
            case 2:  
                display.setCurrent(listImplicit);  
                break;  
        }  
    }  
}
```

Contoh Program (2)

Penanggulangan event untuk List yang bertipe MULTIPLE





```
public class LatihanListMultiple extends MIDlet implements CommandListener{
```

```
    private Display display = Display.getDisplay(this);
```

```
    private Command cmdExit, cmdBack, cmdPilih;
```

```
    private List listMultiple;
```

```
    private Form frmHasil;
```

```
public LatihanListMultiple() {
```

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

```
    cmdBack = new Command ("Back", Command.SCREEN, 3);
```

```
    cmdPilih = new Command("Pilih", Command.SCREEN,2);
```

```
        frmHasil = new Form("Daftar");
```

```
        frmHasil.addCommand(cmdBack);
```

```
listMultiple = new List("Menu", List.MULTIPLE);
listMultiple.append("Apel", null);
listMultiple.append("Belimbing", null);
listMultiple.append("Cempedak", null);
listMultiple.append("Delima", null);
listMultiple.append("Jambu", null);
listMultiple.append("Lemon", null);
listMultiple.append("Mangga", null);
listMultiple.append("Pisang", null);
listMultiple.append("Rambutan", null);

listMultiple.addCommand(cmdExit);
listMultiple.addCommand(cmdPilih);
listMultiple.setCommandListener(this);
```





```
public void startApp() {
    display.setCurrent(listMultiple);
}
public void pauseApp() { }
public void destroyApp(boolean unconditional) { }
public void commandAction(Command c, Displayable d) {
    if (d == listMultiple){
        if (c == cmdPilih)
        {
            frmHasil.append("Berikut ini adalah daftar yang anda pilih : ");
            frmHasil.append("=====\n");

            boolean selected[] = new boolean[listMultiple.size()];
            listMultiple.getSelectedFlags(selected);
        }
    }
}
```



Kode berikut ini dapat disisipkan pada baris program untuk mengetahui status suatu index, apakah terseleksi atau tidak

```
for (int i = 0; i < listMultiple.size(); i++)
```

```
{
```

```
    System.out.println(listMultiple.getString(i) + (selected[i] ?  
    ": selected" : ": not selected"));
```

```
    if (selected[ i ] == true)
```

```
    {
```

```
        frmHasil.append( listMultiple.getString( i ) + "\n");
```

```
    }
```


```
}
```

```
display.setCurrent(frmHasil);
```

```
}
```

```
}
```

Jika terpilih maka
Tambahkan ke dalam
frmHasil



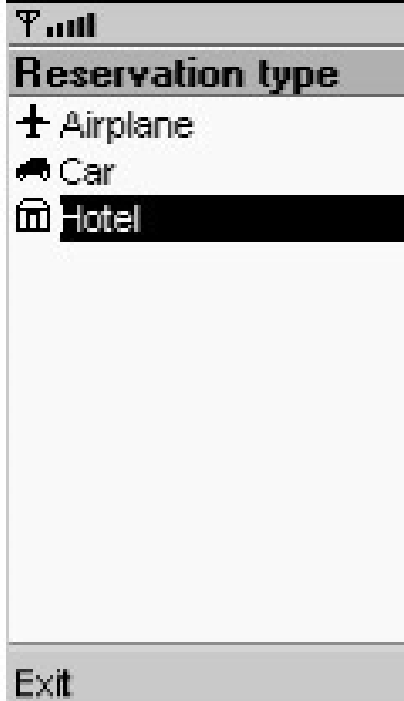
```
if (d == frmHasil){
    if(c == cmdBack ){
        display.setCurrent(listMultiple);
    }
}
}
```

List dengan Image (icon)

```
public class TravelList extends MIDlet
    implements CommandListener {
    List mList;
    Command mExitCommand, mNextCommand;
```

```
public TravelList() {
    String[] stringElements = { "Airplane", "Car", "Hotel" };
    Image[] imageElements = { loadImage("/airplane.png"),
        loadImage("/car.png"), loadImage("/hotel.png") };
```

```
mList = new List("Reservation type", List.IMPLICIT,
    stringElements, imageElements );
```





List dengan Image (icon)

```
mNextCommand = new Command("Next", Command.SCREEN, 0);
mExitCommand = new Command("Exit", Command.EXIT, 0);
mList.addCommand(mNextCommand);
mList.addCommand(mExitCommand);
mList.setCommandListener(this);
}
public void startApp() {
    Display.getDisplay(this).setCurrent(mList);
}
public void pauseApp() {}
public void destroyApp(boolean unconditional) {}
```

List dengan Image (icon)

```
private Image loadImage(String name) {  
    Image image = null;  
    try {  
        image = Image.createImage(name);  
    }  
    catch (IOException ioe) {  
        System.out.println(ioe);  
    }  
  
    return image;  
}
```



List dengan Image (icon)

```
public void commandAction(Command c, Displayable s) {  
    if (c == mNextCommand || c == List.SELECT_COMMAND) {  
        int index = mList.getSelectedIndex();  
        Alert alert = new Alert("Your selection",  
            "You chose " + mList.getString(index) + ".",  
            null, AlertType.INFO);  
        Display.getDisplay(this).setCurrent(alert, mList);  
    }  
    else if (c == mExitCommand)  
        notifyDestroyed();  
}
```

Latihan (1)



❖ Berdasarkan gambar List berikut ini, apa hasil/nilai yang diberikan dari dari `listMultiple.size()` ?

Latihan (2)

```
for (int i = 0; i < listMultiple.size(); i++)  
{  
    if (selected[ i ] == true)  
    {  
        frmHasil.append( listMultiple.getString( i ) + "\n");  
    }  
}
```

Pada potongan kode program di atas, mengapa int i dimulai dari nol?

Latihan(3)

```
for (int i = 0; i < listMultiple.size(); i++)
{
    if ( selected[ i ] )
    {
        frmHasil.append( listMultiple.getString( i ) + "\n");
    }
}
```

- ❖ Jika potongan program sebelumnya yaitu `if (selected[i] == true)` diganti menjadi baris program di atas yaitu `if (selected[i])`, apakah program dapat berjalan sama seperti sebelumnya? Mengapa?



Thanks!

