

BAB II

GUI LANJUT

2.1 Tujuan Instruksional

Bab ini akan membahas mengenai kendali program dengan menggunakan GUI lebih lanjut yaitu TextArea, Panel, CheckBox dan Radio. Komponen GUI akan dipadukan dengan kemampuan program java dalam manipulasi String. Setelah mempelajari bab ini diharapkan mahasiswa akan mampu menulis kode program yang menggunakan komponen GUI untuk memanipulasi string.

2.2 Materi Pembelajaran

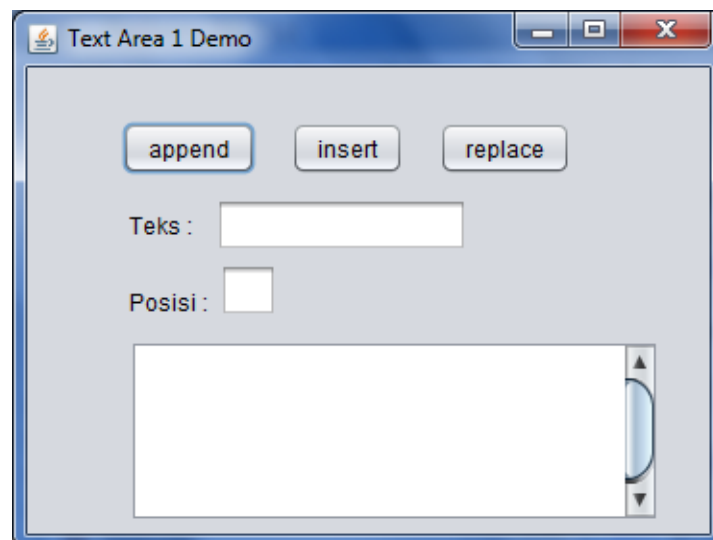
Materi yang akan dibahas dalam bab ini terdiri dari :

- TextArea
- Panel
- CheckBox
- RadioButton
- Table dan ComboBox
- String dan Array
- Hashtable

2.3 TextArea

Komponen TextArea adalah komponen untuk menangkap atau menampilkan beberapa baris teks. Metod penting selain setText() seperti pada TextField adalah :
append(String s) : menambahkan teks s pada komponen yang telah ada teksnya
insert(String s, int pos) : menyisipkan string s pada posisi pos dalam TextArea
replaceRange(String s, int p, int q) : menimpakan string s pada TextArea mulai posisi p sampai q

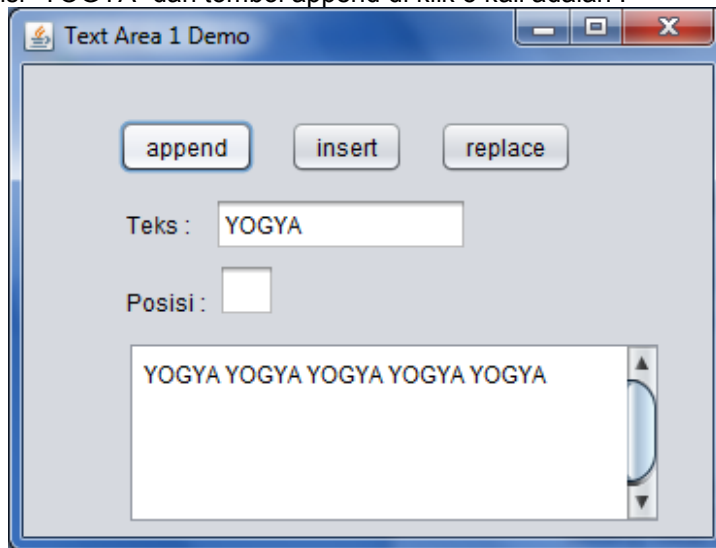
Contoh berikut adalah demo TextArea untuk menambah dan menginsert teks.



Gambar 2.1. Demo TextArea 1

Demo program ini mengambil teks dari TextField pertama (Teks) dan mengambil bilangan posisi dari TextField kedua. Nama variabel TextField pertama adalah tf1 dan TextField kedua adalah tf2 sedang nama TextArea adalah ta.

Hasil jika Teks berisi "YOGYA" dan tombol append di klik 5 kali adalah :

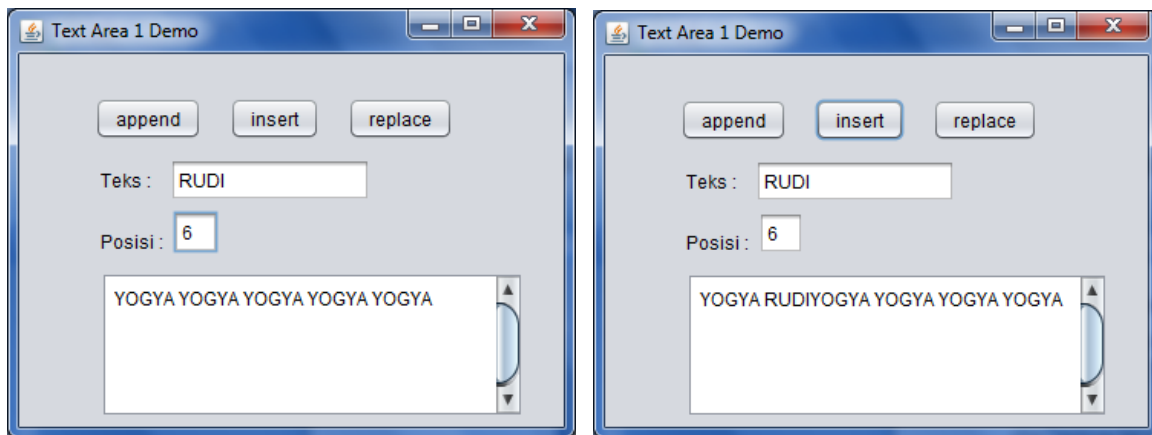


Gambar 2.2. Contoh metod append()

Adapun kode di balik tombol **append** adalah sebagai berikut :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String s=tf1.getText();
ta.append(s+" ");
}
```

Jika TextArea berisi teks seperti pada gambar 2.2. dan Teks diganti "RUDI" sedangkan posisi : 6 maka apabila tombol insert di klik hasilnya seperti Gambar 2.3 berikut :



(a) mula-mula

(b) Setelah tombol insert di klik

Gambar 2.3 Demo metod insert()

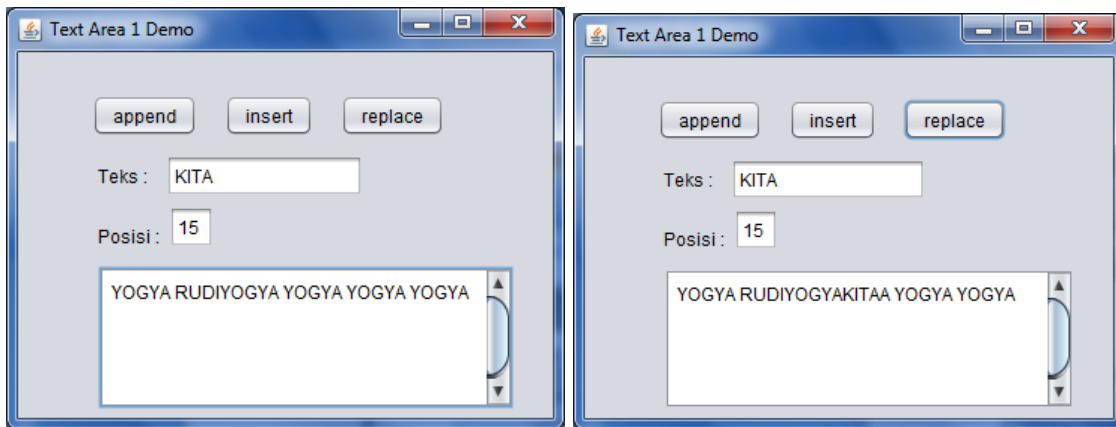
Pada gambar 2.3 (b) terlihat bahwa teks **RUDI** akan diletakkan –menyisip- pada posisi karakter ke enam, yaitu karakter Y pada tulisan YOGYA yang kedua. Berikut ilustrasinya :

YOGYA YOGYA...	YOGYA RUDIYOGYA...
01 23456	01 23456

Adapun kode di balik tombol **insert** adalah sebagai berikut :

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
String s=tf1.getText();
int p=Integer.parseInt(tf2.getText());
ta.insert(s,p);
}
```

Untuk demo metod `replaceRange()`, mula-mula keadaan `TextArea` seperti pada gambar 2.3(b), kemudian pada Teks kita ketikkan kata “KITA” dan posisi kita isi 15 (lihat gambar 2.4 (a)), setelah di klik tombol `replace` akan menjadi seperti gambar 2.4 (b). Terlihat bahwa kata KITA akan menempa



(a) mula-mula

(b) Setelah tombol `replace` di klik

Gambar 2.4 Demo metod `replace()`

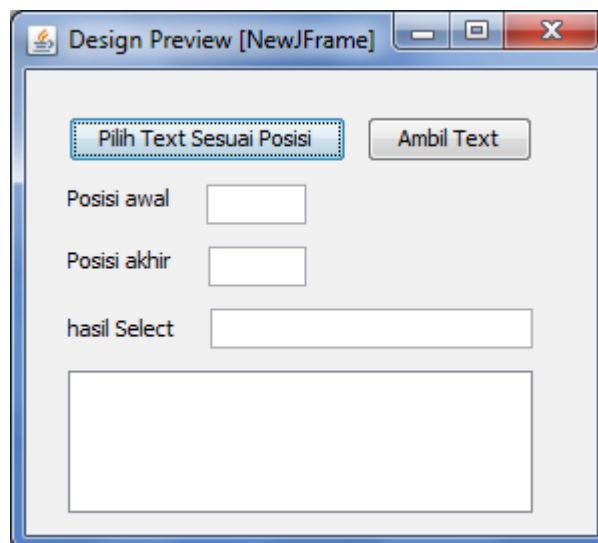
Adapun kode di balik tombol `replace` adalah sebagai berikut :

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    String s=tf1.getText();
    int p=Integer.parseInt(tf2.getText());
    ta.replaceRange(s,p,p+5);
}
```

Contoh kedua berikut akan mendemonstrasikan metod `select()` untuk memilih sub-teks dari suatu teks dan mengambil teks yang sudah terpilih dengan `getSelectedText()` :

`select(int awal , int akhir)` → memilih (blok) seleksi text
`getSelectedText()` → mengambil teks terseleksi oleh `select()`

Mula-mula kita desain Form seperti Gambar 2.5 berikut.



Gambar 2.5 Demo `TextArea 2`

Berikut ini adalah nama-nama variabel pada Form yang sudah diubah, yaitu tombol `Pilih`, tombol `Ambil`, `TextField` `tf1` untuk posisi awal, `tf2` untuk posisi akhir dan `tf3` untuk hasil `select`. Untuk komponen `TextArea` nama variabelnya adalah `ta`.

```
// Variables declaration - do not modify
private javax.swing.JButton Ambil;
private javax.swing.JButton Pilih;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea ta;
private javax.swing.JTextField tf1;
private javax.swing.JTextField tf2;
private javax.swing.JTextField tf3;
// End of variables declaration
```

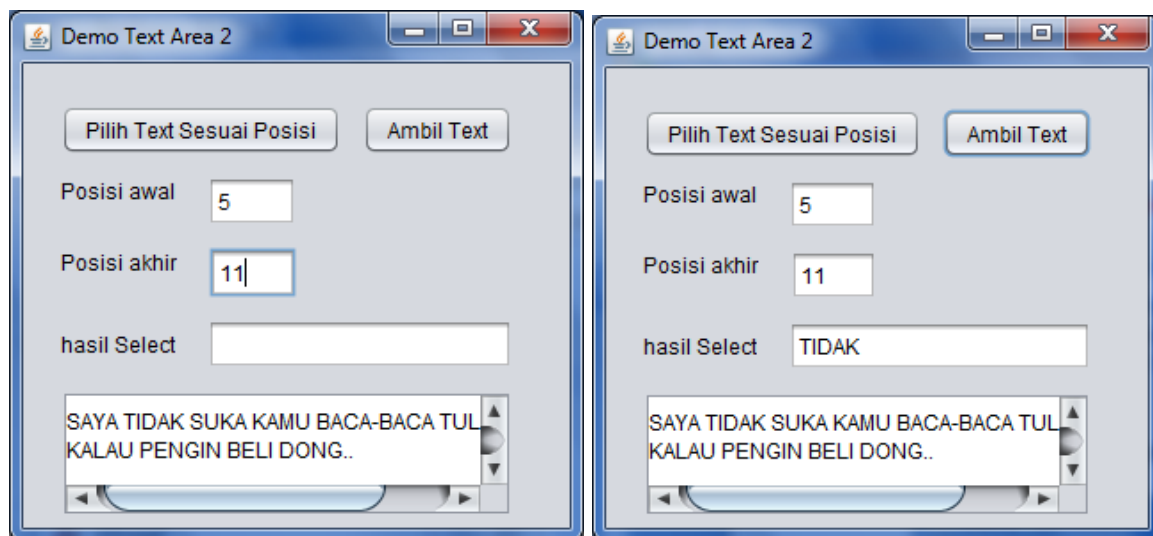
Adapun kode untuk tombol Pilih Text adalah :

```
private void PilihActionPerformed(java.awt.event.ActionEvent evt) {
    int awal=Integer.parseInt(tf1.getText());
    int akhir=Integer.parseInt(tf2.getText());
    ta.select(awal,akhir);
}
```

Dan untuk tombol Ambil adalah :

```
private void AmbilActionPerformed(java.awt.event.ActionEvent evt) {
    String s=ta.getSelectedText();
    tf3.setText(s);
}
```

Jika project dijalankan hasilnya adalah seperti pada Gambar 2.6



(a)memilih posisi awal dan akhir

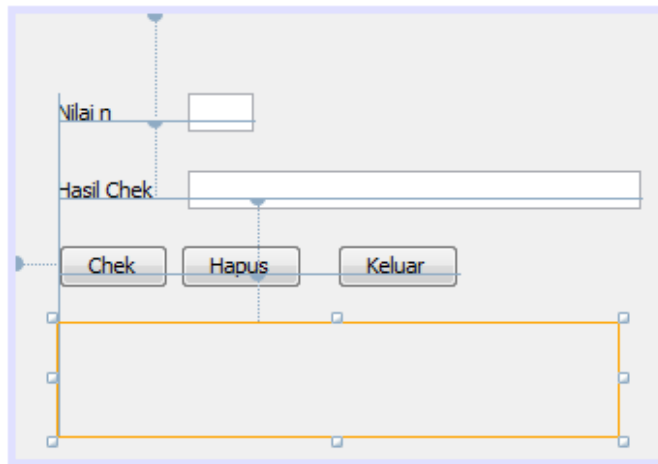
(b) Klik Pilih dan klik Ambil

Gambar 2.6 Hasil Pilih dan Ambil Text

TextAreaPemahaman mengenai perulangan ini memegang peran kunci dalam pembuatan kode program komputer. Dalam kehidupan kita banyak sekali kejadian direncanakan yang selalu dilakukan berulang, seperti mandi makan, tidur, kuliah, ibadah, belajar, beli baju, potong rambut dan seterusnya. Barangkali kalau didaftar kegiatan kita lebih banyak kegiatan berulang daripada kegiatan tidak berulang. Hanya lahir dan mati yang tidak berulang untuk individu. Kegiatan berulang atau “loop” tersebut jika dicermati dapat digolongkan menjadi dua jenis, yaitu ulangan tak hingga dan ulangan terhingga. Dengan demikian karena program komputer adalah simbolisasi kejadian sehari-hari, maka loop pada kode program komputer juga dapat dibedakan menjadi loop tak hingga dan loop terhingga.

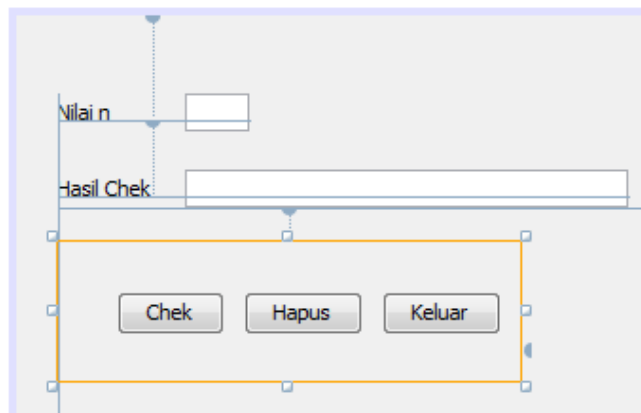
2.4 Panel

Komponen Panel merupakan komponen jenis Container, yaitu komponen yang akan di-“muati” komponen lain. Komponen Panel digunakan untuk menampung beberapa komponen kendali yang sejenis. Komponen panel ini misalnya diletakkan pada Form untuk mengelompokkan beberapa komponen yang memerlukan pengelompokan, misalnya sekelompok tombol atau sekelompok TextField atau CheckBox. Perhatikan gambar 2.7 berikut ini., yang merupakan modifikasi dari Form pada Bab 1 gambar 1.18.



Gambar 2.7 Menyisipkan Komponen Panel pada Form

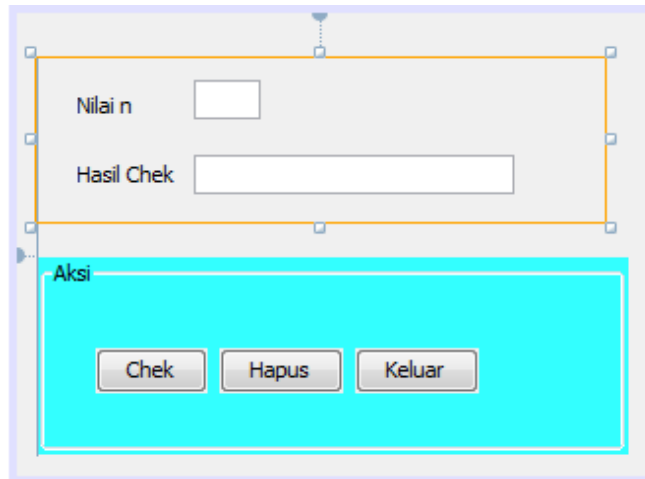
Selanjutnya kelompok Button Chek, Hapus dan Keluar kita drag masuk ke dalam Panel (lihat gambar 2.8) :



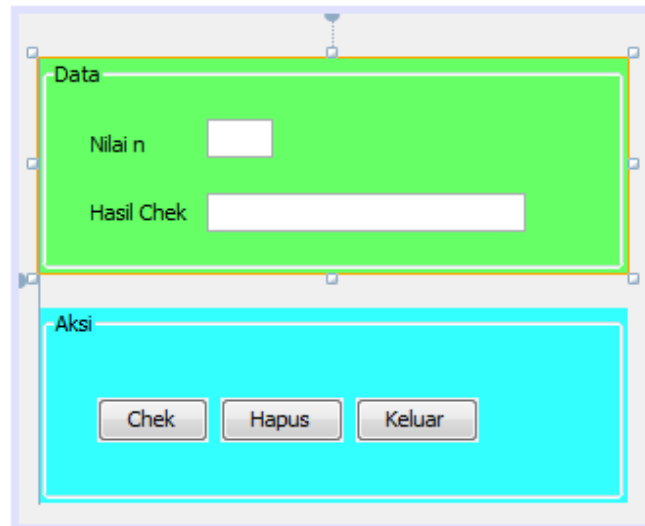
Gambar 2.8 Kelompok Button dimasukkan ke Panel

Selanjutnya Panel pertama tersebut supaya terlihat kontras kita beri warna dan bingkai. Caranya dengan memilih properti Border dan Background. Untuk Border kita pilih Title Border dengan memberikan title : Aksi (lihat gambar 2.9).

Selanjutnya kita pasang Panel yang kedua untuk menaruh TextField dan Label untuk pengelompokan pada komponen input data (lihat gambar 2.9). Pada panel kedua kita juga ubah Background dan Border-nya. Background panel kedua kita ubah berwarna hijau dan bordernya kita beri judul : Data. Hasil akhir Form dapat dilihat pada gambar 2.10.



Gambar 2.9 Border dan Background dari Panel Pertama



Gambar 2.10 Border dan Background Panel kedua

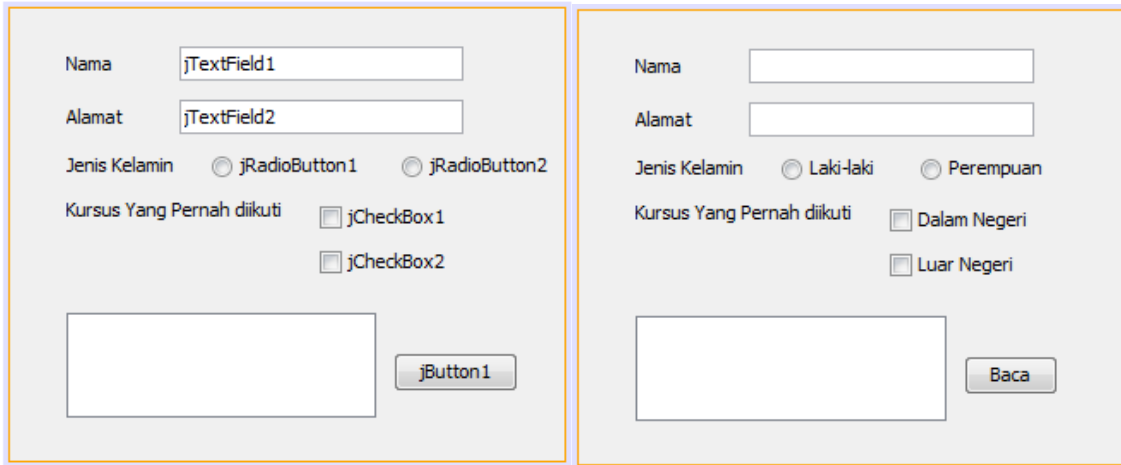
2.5 CheckBox dan RadioButton

CheckBox digunakan untuk pilihan yang dapat memiliki jawaban lebih dari satu, misalnya kursus yang pernah diikuti pegawai (dalam negeri, luar negeri). Sedangkan RadioButton digunakan jika jawaban hanya mungkin satu saja, misalnya jenis kelamin atau agama. Berikut ini contoh penggunaan komponen tersebut untuk input data pegawai.

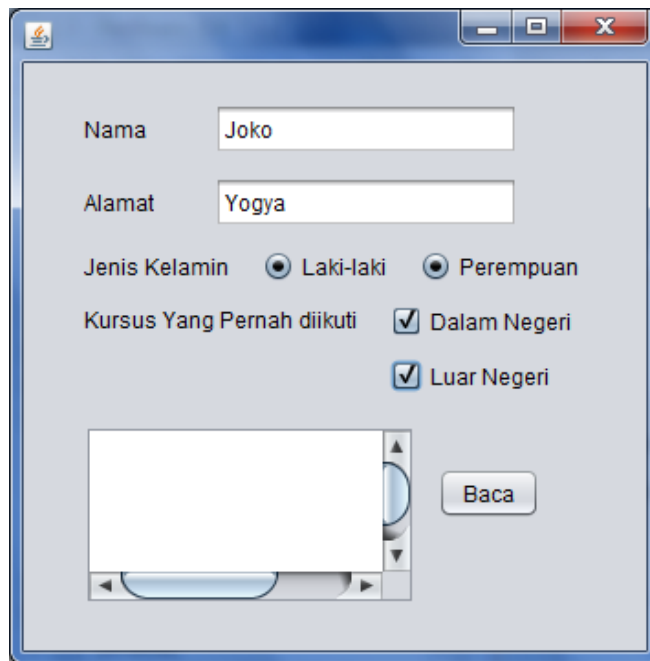
Pada gambar 2.11 (a) adalah komponen-komponen yang akan digunakan untuk membaca data, yaitu dua TextField untuk membaca nama dan alamat, dua RadioButton untuk membaca data jenis kelamin (pilihan hanya boleh satu), dan dua CheckBox untuk membaca kursus yang pernah diikuti (pilihan dapat lebih dari satu). Hasil pemberian label yang sesuai dengan jenis kelamin dan kursus yang diikuti adalah seperti dalam gambar 2.11 (b).

Langkah berikutnya yang lebih penting adalah men-desain aksi dari komponen. Sebelum kita mendesain untuk komponen RadioButton maka komponen RadioButton yang harusnya hanya dapat dipilih satu jawaban ternyata dapat dipilih dua jawaban (lihat gambar 2.12).

Untuk mengatur agar RadioButton hanya dapat dipilih satu pilihan saja, maka RadioButton tersebut harus diikat dalam suatu ButtonGroup. Langkah pertama adalah kita letakkan ButtonGroup dalam Form. Harus diingat bahwa komponen ButtonGroup adalah komponen yang tidak terlihat ketika diletakkan pada Form, tetapi kita dapat melihat dalam daftar variabel akan muncul nama baru yaitu :jButtonGroup1.

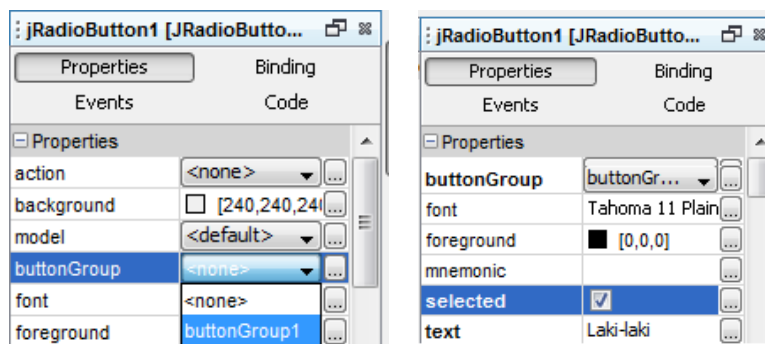


(a) Komponen awal (b) Setelah diatur
Gambar 2.11 Komponen CheckBox dan RadioButton



Gambar 2.12 Pilihan jenis kelamin sebelum diatur

Selanjutnya kita atur properti `jRadioButton1` (untuk pilihan Laki-laki) dan `jRadioButton2` (untuk pilihan Perempuan). Gambar 2.13 menunjukkan set properti `buttonGroup` dan `selected` yang dipilih.

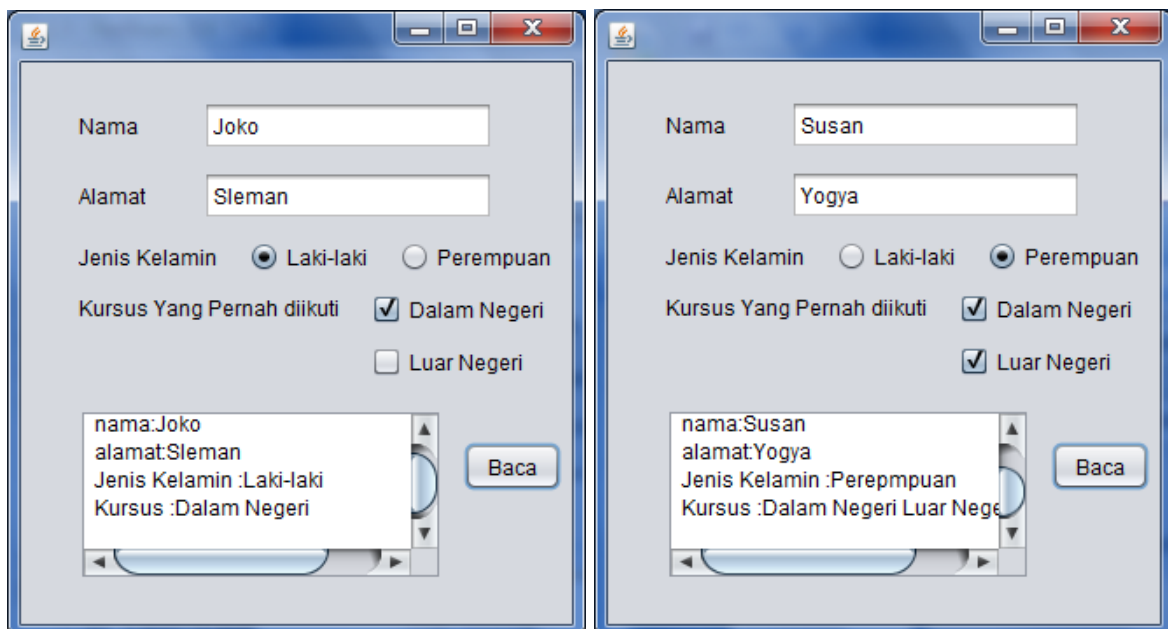


Gambar 2.13 Menset properti dari `jRadioButton1`

Untuk mengetahui RadioButton mana yang dipilih atau CheckBox mana saja yang dipilih pengguna, kita dapat menggunakan properti `isSelected()` yang bernilai `true` jika ia terpilih dan `false` jika tidak terpilih. Sebagai contoh untuk menguji efek pemilihan komponen menggunakan `isSelected()` kita mendefinisikan aksi pada Button Baca seperti berikut ini :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextArea1.append("nama:" + jTextField1.getText()+"\n");
    jTextArea1.append("alamat:" + jTextField2.getText()+"\n");
    String jk;
    if(jRadioButton1.isSelected())jk="Laki-laki"; else jk="Pereempuan";
    jTextArea1.append("Jenis Kelamin :" +jk+"\n");
    String kursus="";
    if(jCheckBox1.isSelected())kursus="Dalam Negeri";
    if(jRadioButton2.isSelected())kursus+=" Luar Negeri";
    jTextArea1.append("Kursus :" +kursus +"\n");
}
```

Jika data-data diisi dan dipilih kemudian tombol baca di-klik maka hasilnya seperti terlihat dalam gambar 2.14.

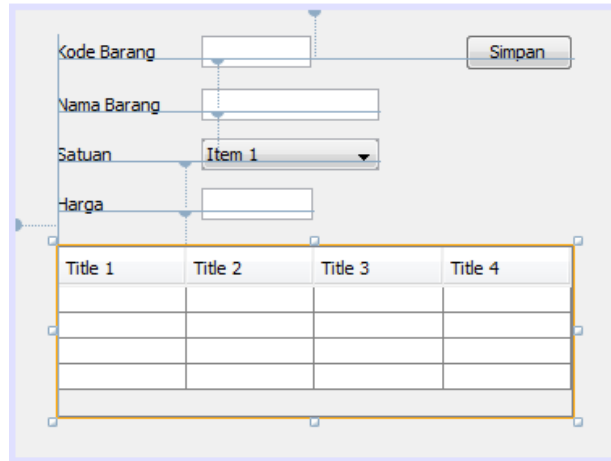


Gambar 2.14 Dua kemungkinan pengisian Jenis Kelamin dan Kursus (isi dan pilih lalu klik tombol Baca)

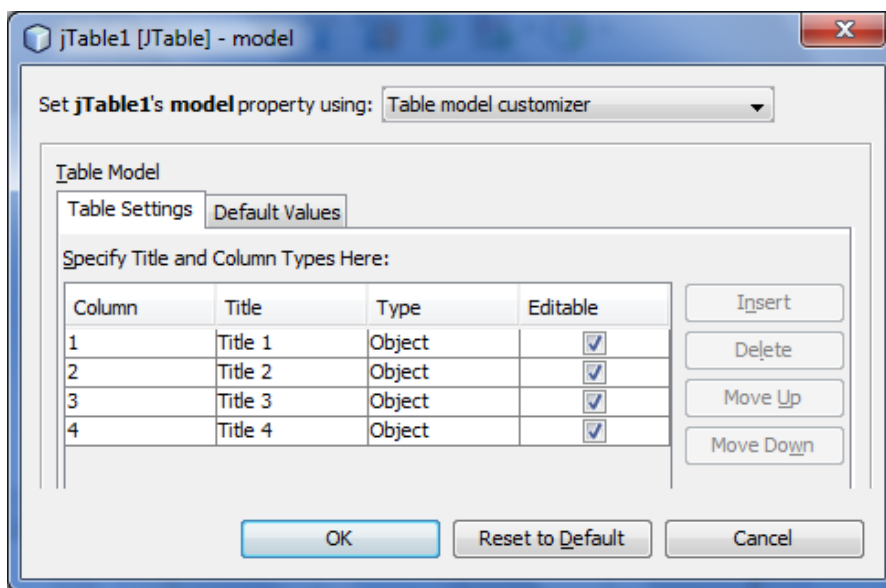
2.6 Table dan ComboBox

Komponen Table digunakan untuk menyimpan data yang umumnya berupa record-record data dari suatu database. Jika suatu komponen Table dipasang dalam Form seperti Gambar 2.15 maka langkah awal adalah menyesuaikan judul-judul kolom disesuaikan dengan data yang ada. Misalnya kita akan menyimpan data barang yang terdiri dari Kode Barang, Nama Barang, Satuan Barang dan Harga Barang. Untuk mengatur jumlah kolom, jumlah baris yang muncul, Judul kolom dan tipe data kolom kita menggunakan properti model dari table (lihat gambar 2.16 dan gambar 2.17).

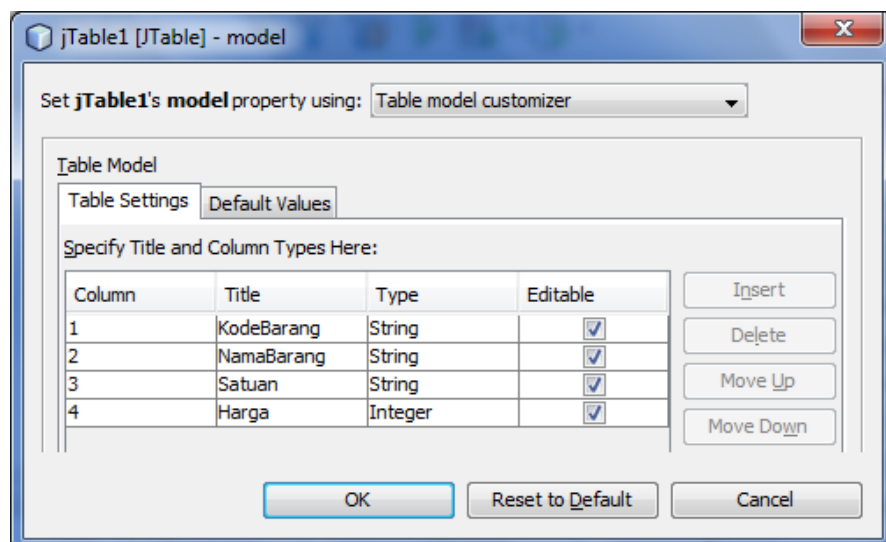
Untuk ComboBox setting dilakukan pada nama-nama item. Untuk satuan barang nama-nama item misalnya Lembar, Kotak, Botol dan Box kita set melalui properti model dari komponen Combo Box (lihat gambar 2.18)



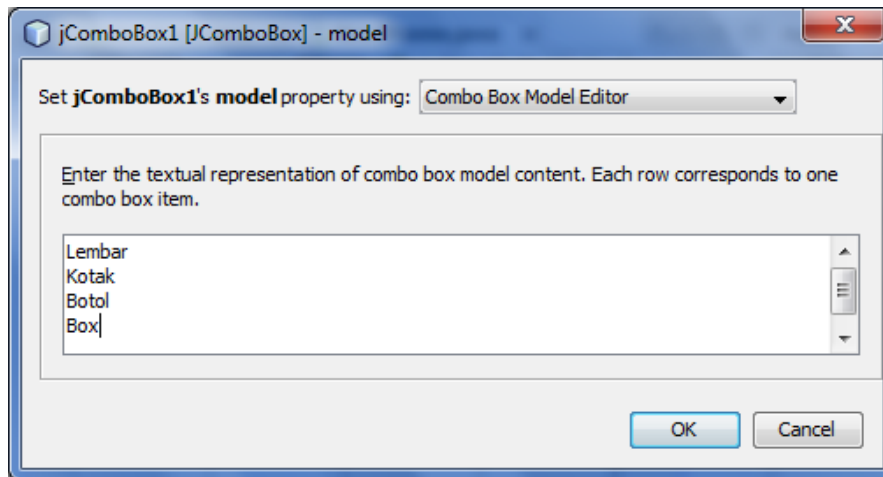
Gambar 2.15 Komponen Table dan ComboBox yang terpasang



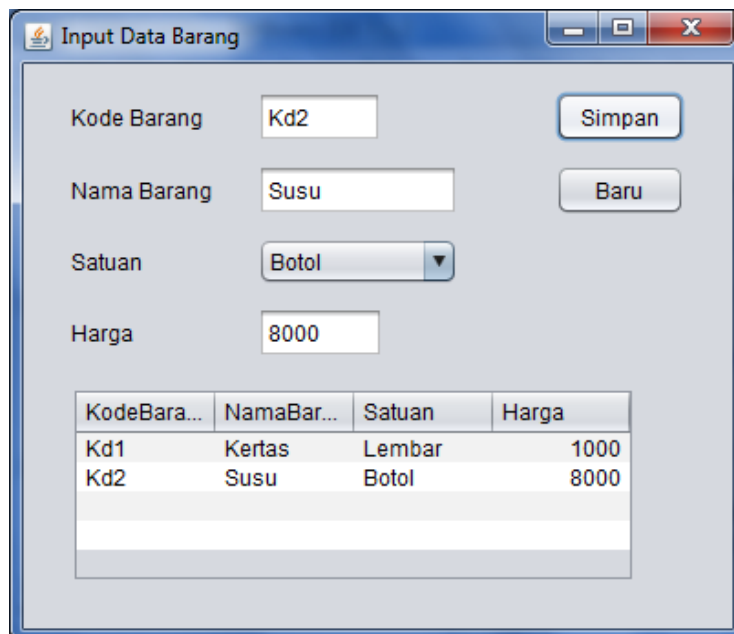
Gambar 2.16 Properti model Table sebelum diset



Gambar 2.17 Judul Kolom dan Tipe diubah



Gambar 2.18 Men-set properti dari ComboBox



Gambar 2.19 Contoh Input data ke dalam Tabel

Adapun kode dalam Tombol Simpan adalah sebagai berikut. Kode ini mengambil data dari TextField dan me-nyisipkan ke dalam Table menggunakan **setValueAt()** :

```
private void SimpanActionPerformed(java.awt.event.ActionEvent evt) {
    jTable1.setValueAt(jTextField1.getText(), brs, 0);
    jTable1.setValueAt(jTextField2.getText(), brs, 1);
    jTable1.setValueAt(jComboBox1.getSelectedItem().toString(), brs, 2);
    jTable1.setValueAt(jTextField3.getText(), brs, 3);
}
```

Adapun kode dalam Tombol Baru adalah sebagai berikut. Kode ini hanya membersihkan TextField dan menaikkan konter brs untuk menyimpan pada baris berikutnya :

```
private void BaruActionPerformed(java.awt.event.ActionEvent evt) {
    brs+=1;
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
}
```

2.7 String dan Array

String merupakan class penting dalam pemrograman umumnya. Pada pemrograman java String ada suatu class yang memiliki banyak sekali method untuk manipulasi string. Sebagai suatu class definisi string pada hakekatnya adalah pembuatan objek string. Sehingga pernyataan berikut memiliki makna yang sama :

```
String s="JOGJA"           sama dengan
String s=new String("JOGJA");
```


Ekspresi pertama dinilai lebih praktis sehingga lebih sering digunakan.

String memiliki banyak sekali method yang bersifat public yang disediakan untuk manipulasi string. Contoh beberapa method tersebut adalah :

1. public int length() : memberikan panjang string
2. public char charAt(int index) : memberikan char pada posisi index
3. public byte[] getBytes(String charsetName) : mengkonversi string ke array byte
4. public boolean equals(Object anObject) : mengecek kesamaan string
5. public int indexOf(String str) : mencari posisi string str pada string
6. public String substring(int beginIndex) : mengambil substring mulai dari beginIndex
7. public String substring(int beginIndex, int endIndex) : mengambil substring mulai dari beginIndex sampai <endIndex
8. public String toUpperCase() : merubah ke huruf kapital
9. public String toLowerCase() : merubah ke huruf kecil
10. public char[] toCharArray() : konversi ke array dari char

Contoh penggunaan charAt() untuk mengurai string per-karakter adalah :

```
class StringManip1
{public static void main(String [] arg)
  {String s="JOGJAKARTA";
   int n=s.length();
   for(int i=0;i<n;i++)
     System.out.println("karakter ke-"+i+" = "+s.charAt(i));
  }
}
```



Gambar 2.20 Hasil program StringManip1

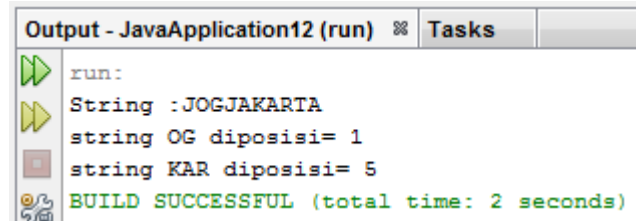
Contoh penggunaan indexOf(), mencari posisi suatu string pada string lain :

```
class StringManip2
{public static void main(String [] arg)
  {String s="JOGJAKARTA";
```

```

String s1="OG";
String s2="KAR";
System.out.println("String :"+s);
System.out.println("string "+s1+" diposisi= "+s.indexOf(s1));
System.out.println("string "+s2+" diposisi= "+s.indexOf(s2));
}
}

```



Gambar 2.21. Hasil program StringManip2

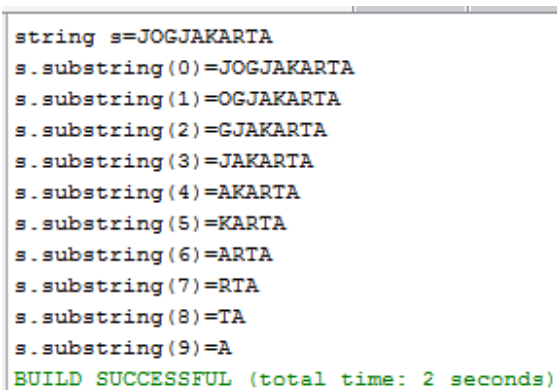
Contoh penggunaan substring() untuk memotong string mulai posisi tertentu sampai akhir

```

class StringManip3
{public static void main(String [] arg)
{String s="JOGJAKARTA";
System.out.println("string s="+s);
for(int i=0;i<s.length();i++)
System.out.println("s.substring("+i+"="+s.substring(i));
}
}

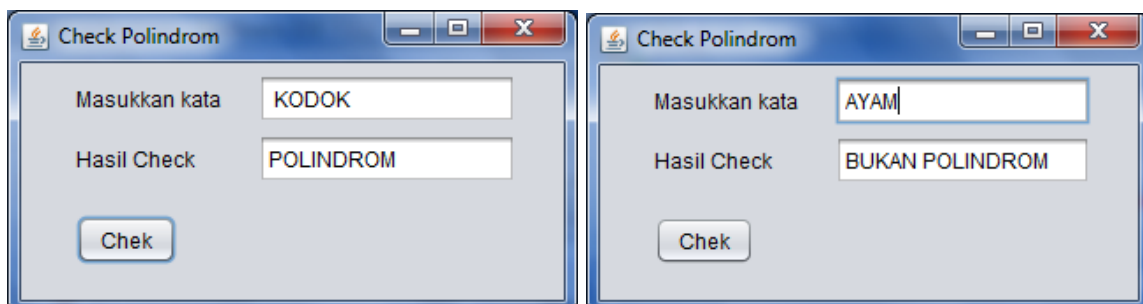
```

Hasil program



Gambar 2.22. Hasil program StringManip3

Contoh program GUI gambar 2.23 berikut adalah penggunaan manipulasi string untuk mengecek suatu string POLINDROM atau BUKAN POLINDROM. String POLINDROM jika ia memiliki kebalikan yang sama dengan string aslinya, misalnya KODOK adalah POLINDROM sedangkan AYAM adalah BUKAN POLINDROM.



Gambar 2.23. Hasil program StringManip3

Adapun kode terpenting dari GUI pada gambar 2.23 adalah kode di dalam button Chek yang berfungsi mengecek apakah string tersebut jika dibalik sama apa tidak. Logikanya sederhana, kita membuat string baru dengan cara menyusun dari string lama tetapi karakternya disusun dari belakang sehingga menjadi "kebalikan". Berikut ini adalah kode tersebut :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String s=jTextField1.getText().trim();  
    String s1="";  
    for(int i=0;i<s.length();i++) s1=s.charAt(i)+s1;  
    if(s1.equals(s))jTextField2.setText("POLINDROM");  
    else jTextField2.setText("BUKAN POLINDROM");  
}
```

Array 1 dimensi

Array satu dimensi didefinisikan dengan bentuk umum :

```
String[] Nama; atau String Nama[];  
int[] Umur; atau int Umur[];
```

Untuk mengalokasikan memory tersebut digunakan kata kunci new:

```
Nama =new String[N];  
Umur =new int[N];
```

Dalam hal ini N adalah kapasitas array tersebut, yang selanjutnya array tersebut diases dengan indek mulai dari 0 sampai N-1.

Kedua langkah tersebut dapat digabungkan menjadi :

```
String[] Nama=new String[N];  
int[] Umur=new int[N];
```

Selanjutnya Nama dan Umur sebagai objek array dapat diakses dengan menyebutkan indeksinya. Indeks array dalam java selalu dimulai dari nilai 0 dan indeks terakhir adalah nilai N-1, jika N adalah cacah elemen array. Misalnya diisikan data kedalam elemen array sebagai berikut :

```
Nama[0]="JON";  
Umur[0]=25;  
Nama[1]="ALI"  
Umur[1]=30;  
.... dst
```

Apabila diinginkan inisialisasi data langsung bersamaan dengan deklarasi array, maka langkah alokasi memory tidak perlu dilakukan, misalnya sebagai berikut :

```
String[] Nama={"JON", "ALI", "SALEH", "PUTRI"};  
Int[] Umur={25, 30, 55, 35};
```

Contoh program :

Contoh berikut menggunakan array Nama dan umur dan mencetaknya dengan loop.

```
class ArrayOne  
{  
    public static void main(String args[])  
    {  
        String[] Nama={"JON", "ALI", "SALEH", "PUTRI", "DEDI"};  
        int[] Umur={25, 30, 55, 35, 40};
```

```

    for (int i=0; i<Nama.length; i++)
        System.out.println(Nama[i]+" " +Umur[i]);
}
}

```

Output program adalah :

```

JON 25
ALI 30
SALEH 55
PUTRI 35
DEDI 40
BUILD SUCCESSFUL (total time: 0 seconds)

```

Gambar 2.24 Hasil Array satu dimensi

Panjang (jumlah elemen) suatu array dapat diketahui dengan menggunakan properti length, seperti terlihat pada program : Nama.length ; artinya jumlah elemen dari array Nama.

Array 2 dimensi

Array dua dimensi didefinisikan dan dialokasikan dengan cara :

```

tipeData[n][m] namaArray
atau
tipeData NamaArray[n][m];

```

Contoh Program : Kamus 2 bahasa

Di bawah ini contoh program dengan array 2 dimensi dari string untuk simulasi kamus inggris indonesia. Kata dalam program adalah suatu array dua dimensi dari string yang digunakan untuk menyimpan data kata inggris dan indonesia seperti gambar berikut :

	Kolom 0	kolom 1
baris 0	ONE	SATU
baris 1	COW	SAPI
baris 2	RUN	LARI
baris 3	RED	MERAH
baris 4	MAN	ORANG

Program Penerapan array 2 dimensi untuk kamus

```

import java.io.*;
public class JavaApplication12 {
public static void main(String [] arg){
    String[][] Kata={ {"ONE", "SATU"},
                      {"COW", "SAPI"},
                      {"RUN", "LARI"},
                      {"RED", "MERAH"},
                      {"MAN", "ORANG"};

    String    KataDicari=bacaString( );

    int    k=-1;
    for (int i=0; i<5; i++)
        if (Kata[i][0].equals(KataDicari) ) k=i;
    //cek ketemu atau tidak , jika k=-1 tidak ketemu
    if (k==(-1))
        System.out.println("KATA TERSEBUT TIDAK ADA");
}
}

```

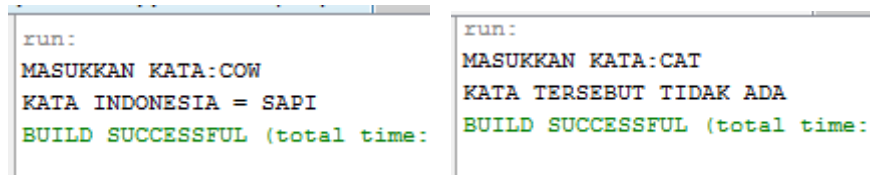
```

        else
            System.out.println("KATA INDONESIA = "+Kata[k][1]);
    }

    private static String bacaString() {
        System.out.print("MASUKKAN KATA:");
        int n=0; String s="";
        try{while(n!=10){
            n=System.in.read();
            if(n!=10)s+=(char)n;
        }
        }catch(IOException ioe){int a=0;};
        return s;
    }
} //end of class

```

Hasil program jika dijalankan adalah seperti tampilan Gambar 2.25 berikut ini.



Gambar 2.25 Hasil Program kamus non GUI

2.8 Hashtable

Hashtable adalah class untuk menyimpan objek dengan menggunakan kunci tertentu yang juga merupakan objek. Beberapa method yang penting antara lain adalah :

put (Objek kunci, Objek elemen) : menaruh elemen dengan kunci tertentu
 get (kunci) : akan memberikan nilai elemen yang bertipe Objek
 containsKey(Objek kunci) : bernilai true jika Hashtable mengandung elemen dengan kunci

Contoh membaca satu elemen berdasar kunci :

```

Hashtable HT=new Hashtable();
HT.put(new Integer(1),"Satu");
HT.put(new Integer(2),"Dua");
HT.put(new Integer(3),"Tiga");
Integer kunci=new Integer(1);
if(HT.containsKey(kunci))
    System.out.println("kunci 1 elemennya = "+(String)HT.get(kunci));

```

Hasilnya adalah :

```
Kunci 1 elemennya = Satu
```

Contoh membaca semua elemen :

```

for (Enumeration e = HT.elements(); e.hasMoreElements();)
    System.out.println(e.nextElement());

```

Hasilnya adalah :

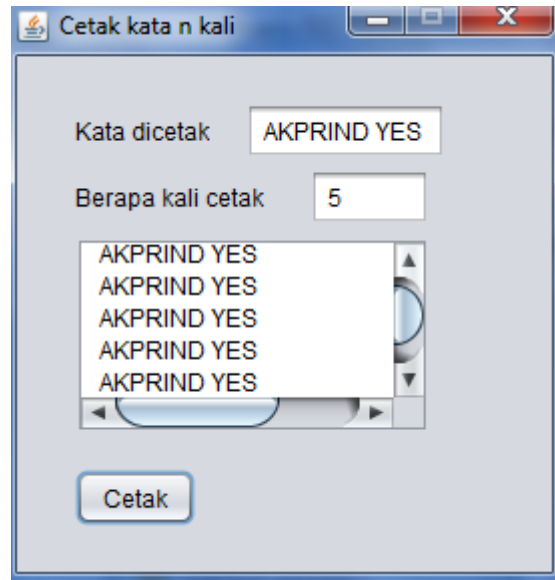
```

Tiga
Dua
Satu

```

2.8 Latihan Soal

1. Buat GUI untuk mencetak kata (yang diinputkan menggunakan TextField) sebanyak n kali (nilai n juga diinputkan dari TextField). Tampilan yang diinginkan seperti pada Gambar 2.26 berikut.



Gambar 2.26 Tampilkan kata n kali

2. Modifikasikan Program Kamus dengan menggunakan Array seperti pada Contoh Gambar 2.20 menggunakan antar muka grafis (GUI) yang sesuai.
3. Modifikasikan program kamus menggunakan Hashtable, jika anda mampu memodifikasikan program kamus dua bahasa dan memprogramnya menjadi kamus English-Indonesia dan Indonesia-English maka anda telah menyelesaikan Tugas Pertama dari Mata Kuliah ini. Ingat Untuk tugas pertama ini anda harus menggunakan Hashtable dengan mendefinisikan method untuk mencari kata dalam Hashtable