

MATCH GAME

Yuliana Setiowati
Politeknik Elektronika Negeri Surabaya

Target Praktikum :

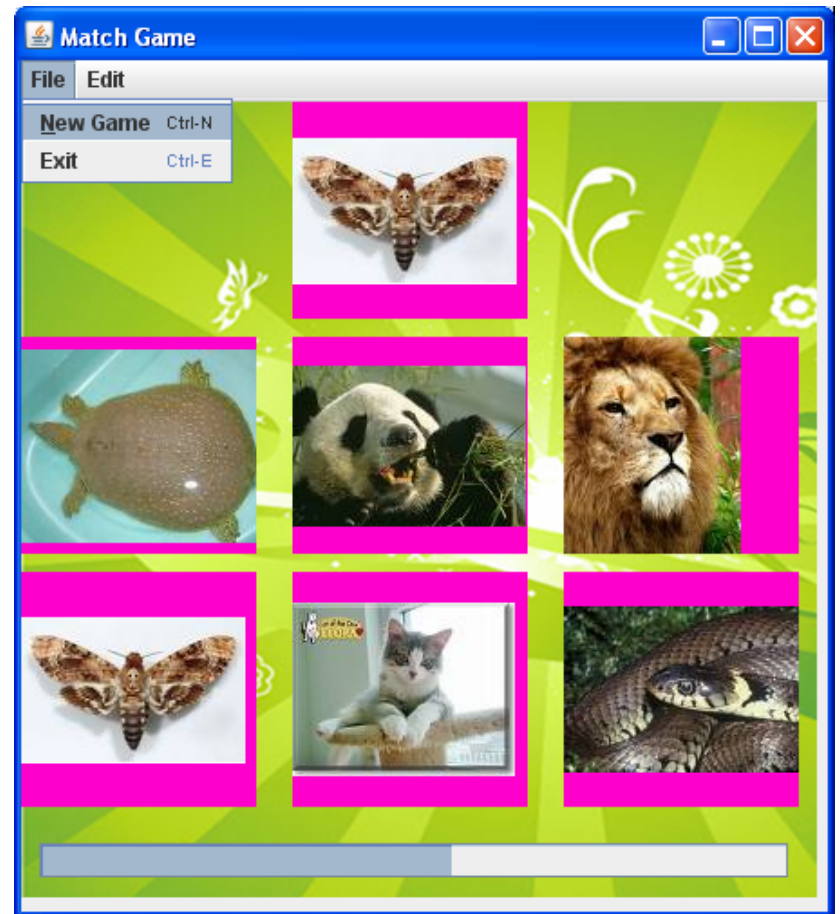
- Mahasiswa dapat membuat aplikasi pembacaan file.
- Mahasiswa dapat menggunakan ImageIcon, Menu Bar, ProgressBar.
- Mahasiswa dapat menggunakan JOptionPane, membuka dan menutup JOptionPane.

Match Game

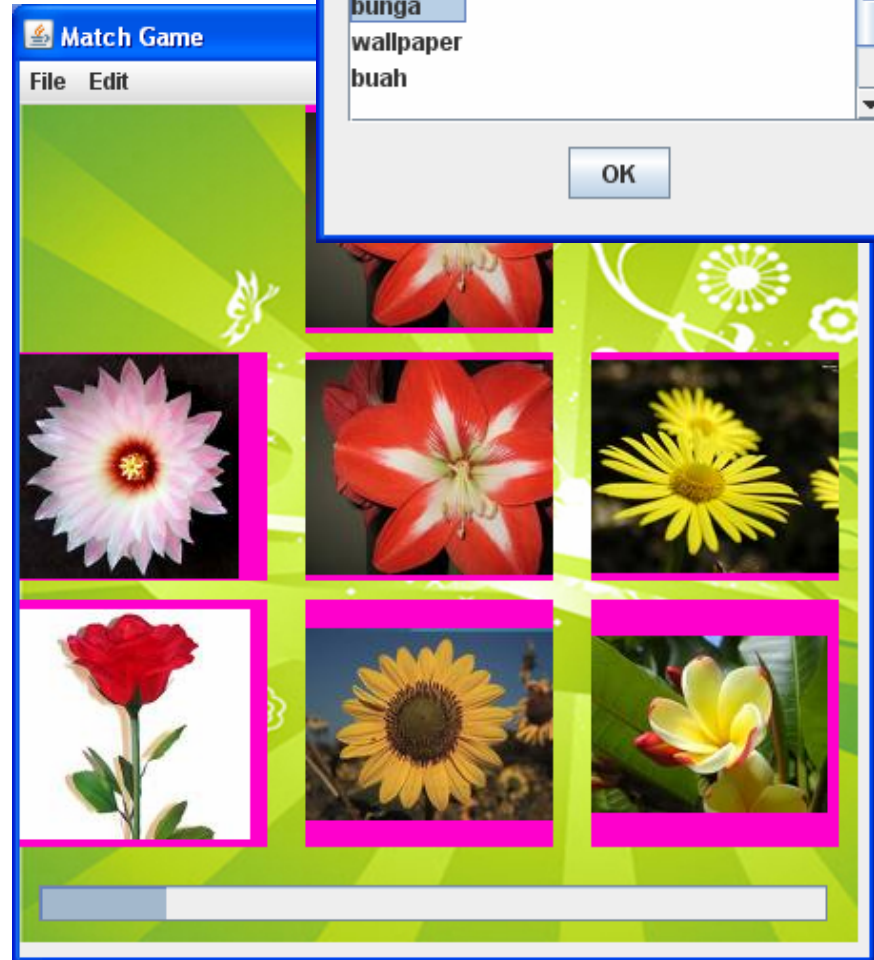
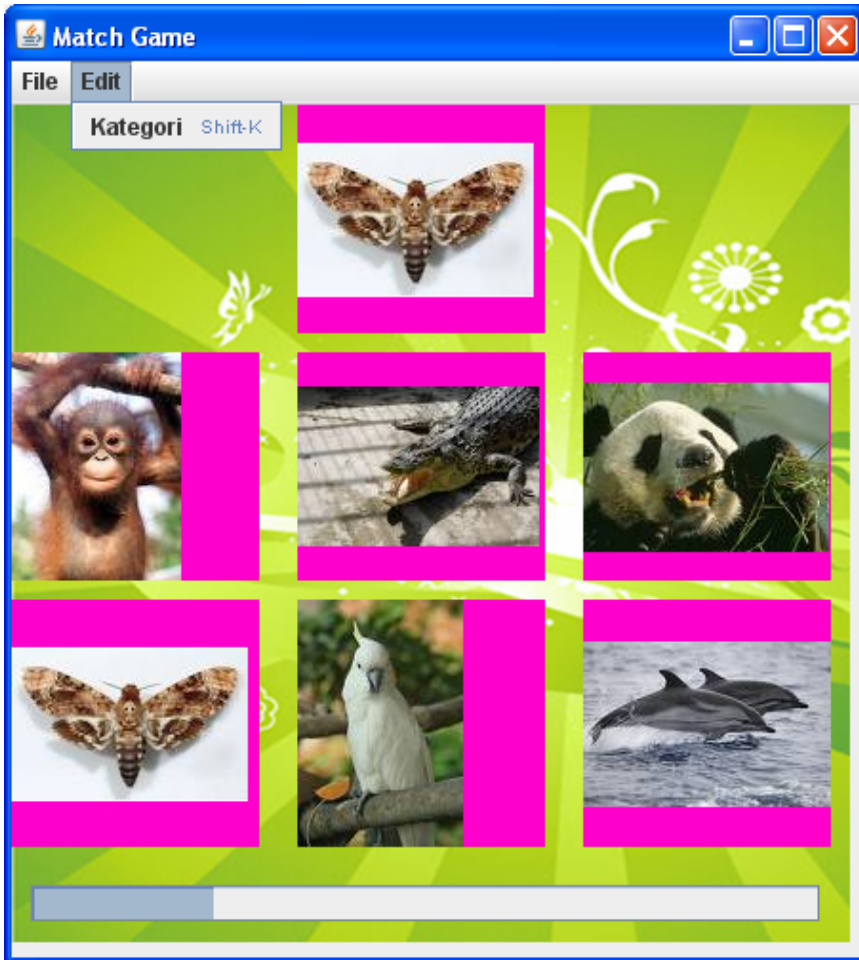
- Pada praktikum ini mahasiswa membuat aplikasi tentang Match Game. User harus mencocokkan gambar acuan dengan 6 gambar yang ada dan harus memilih mana yang sama. Jika user benar dalam melakukan pencocokan maka tampilkan suara "Kamu Hebat" (suara sesuai keinginan), jika user salah maka tampilkan suara "Masih Salah Coba Lagi" (suara sesuai keinginan).
- Kategori gambar dan data gambar tergantung dari keinginan user (bisa ditambahkan maupun dikurangi). Kategori gambar dan nama gambar tersimpan dalam file text.

Match Game

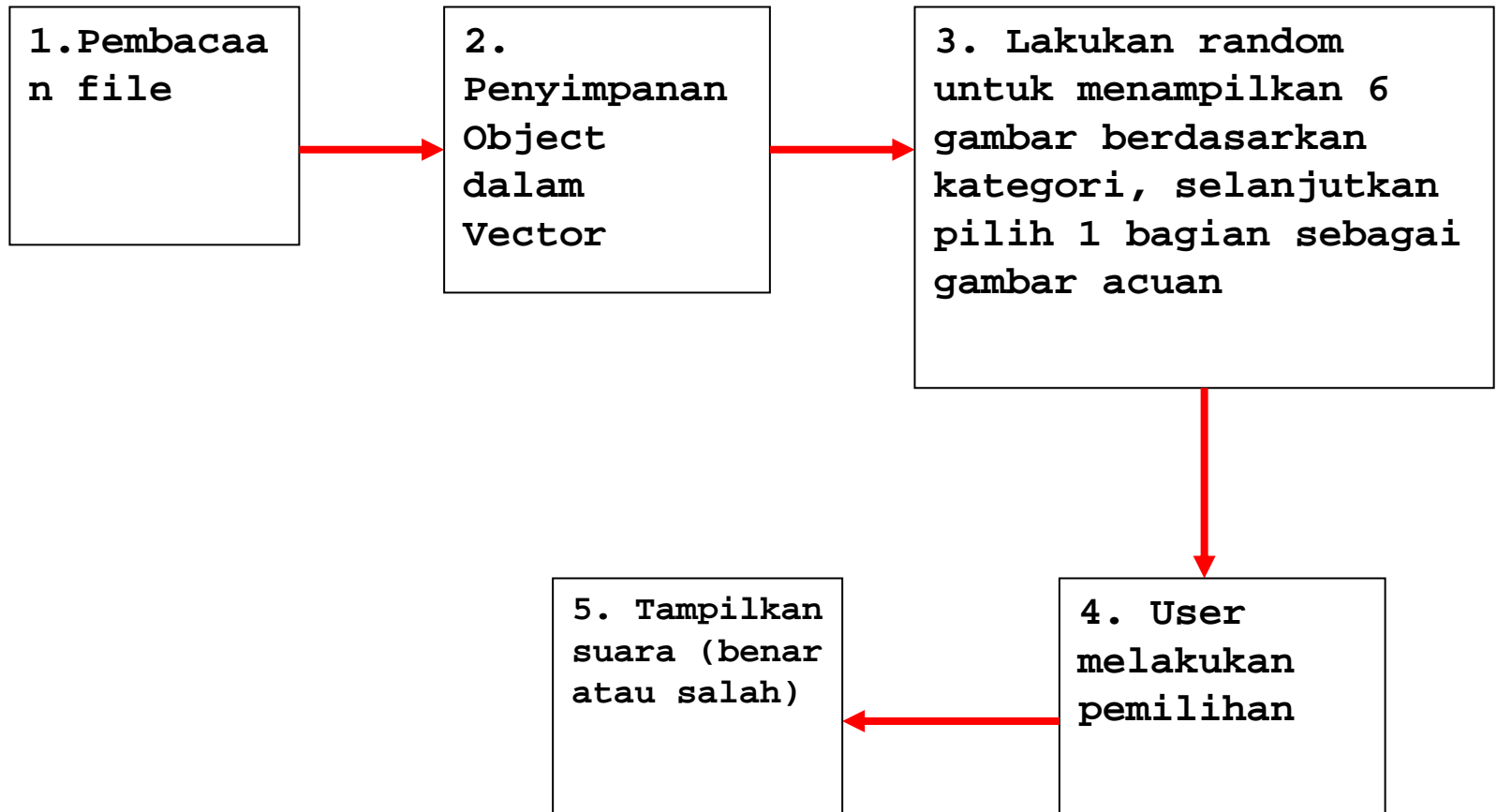
Kategori Binatang



Match Game



Perancangan Sistem



Variabel-variabel penting

- **Vector dataGambar** : objek dataGambar berupa Vector. Objek dataGambar ini digunakan untuk menyimpan objek-objek Gambar, pada saat pembacaan file txt pada baris kedua dan baris selanjutnya.
- **Vector kategori** : objek kategori bertipe Vector, digunakan untuk menyimpan kategori gambar. Isi dari objek kategori berupa String. Penyimpanan ini dilakukan pada saat pembacaan file txt, data pada baris pertama.
- **Gambar kategoriMain** : objek kategoriMain berupa objek Gambar, untuk menyimpan kategori dari gambar yang akan dibangkitkan (contoh kategori misal : bunga, binatang, kendaraan). Yang dipentingkan pada objek kategoriMain ini adalah kategorinya, sehingga namaGambarnya bisa diisi dengan sembarang nilai.
Contoh :
Gambar kategoriMain = new Gambar("binatang","abc");

Variabel-variabel penting

- **int indexGambar[]** : pada aplikasi akan menampilkan gambar sebanyak 6. Misal objek gambar dengan kategori kartun tersimpan pada index 10 – 19. Maka indexGambar ini akan menyimpan 6 data yaitu bilangan antara 10 – 19. Dengan asumsi bilangan tidak boleh sama. Contoh : `int indexGambar[] = {11,18,17,13,15}`
- **int noAcuan** : dari data index yang tersimpan di indexGambar, kita ambil salah satu, kita simpan pada variabel noAcuan. Contoh `int noAcuan = 17`
- **int value =0** : untuk memberikan nilai awal pada progress bar. Contoh `pBar.setValue(value);`

1. Pembacaan File

- Penyimpanan nama gambar menggunakan file teks dengan format seperti di bawah ini. Baris pertama menyatakan kategori yang ada pada aplikasi Match Game, sedangkan baris berikutnya menyatakan kategori dan nama gambar.

```
binatang kartun
binatang binatang1.jpeg
binatang binatang2.jpeg
binatang binatang3.jpeg
binatang binatang4.jpeg
binatang binatang5.jpeg
binatang binatang6.jpeg
binatang binatang7.jpeg
binatang binatang8.jpeg
binatang binatang9.jpeg
binatang binatang10.jpeg
kartun kartu1.jpeg
kartun kartu2.jpeg
kartun kartu3.jpeg
kartun kartu4.jpeg
kartun kartu5.jpeg
kartun kartu6.jpeg
kartun kartu7.jpeg
kartun kartu8.jpeg
kartun kartu9.jpeg
kartun kartu10.jpeg
```

Terdapat 2 kategori yaitu
binatang dan kartun

Binatang binatang1.jpeg →
menyatakan kategori
binatang dengan nama
gambar : binatang1.jpeg

2. Penyimpanan Object dalam Vector

- Baris pertama pada file teks menyatakan kategori, disimpan pada **Vector kategori**.
- Vector kategori berisi String kategori

0	<code>new String(binatang)</code>
1	<code>new String(kartun)</code>

2. Penyimpanan Object dalam Vector

- Baris ke dua dan berikutnya disimpan sebagai objek Gambar, selanjutnya objek-objek tersebut disimpan pada **Vector dataGambar**.

```
public class Gambar {  
    private String kategori ;  
    private String namaGambar ;  
  
    public Gambar(String k, String nm){  
        this.kategori = k ;  
        this.namaGambar = nm ;  
    }  
  
    public void setKategori(String kategori){  
        this.kategori = kategori ;  
    }  
  
    public void setNamaGambar(String nm){  
        this.namaGambar = nm ;  
    }  
  
    public String getKategori(){  
        return kategori ;  
    }  
  
    public String getNamaGambar(){  
        return namaGambar ;  
    }  
  
    public String toString(){  
        return kategori+ " " +namaGambar ;  
    }  
}
```

2. Penyimpanan Object dalam Vector

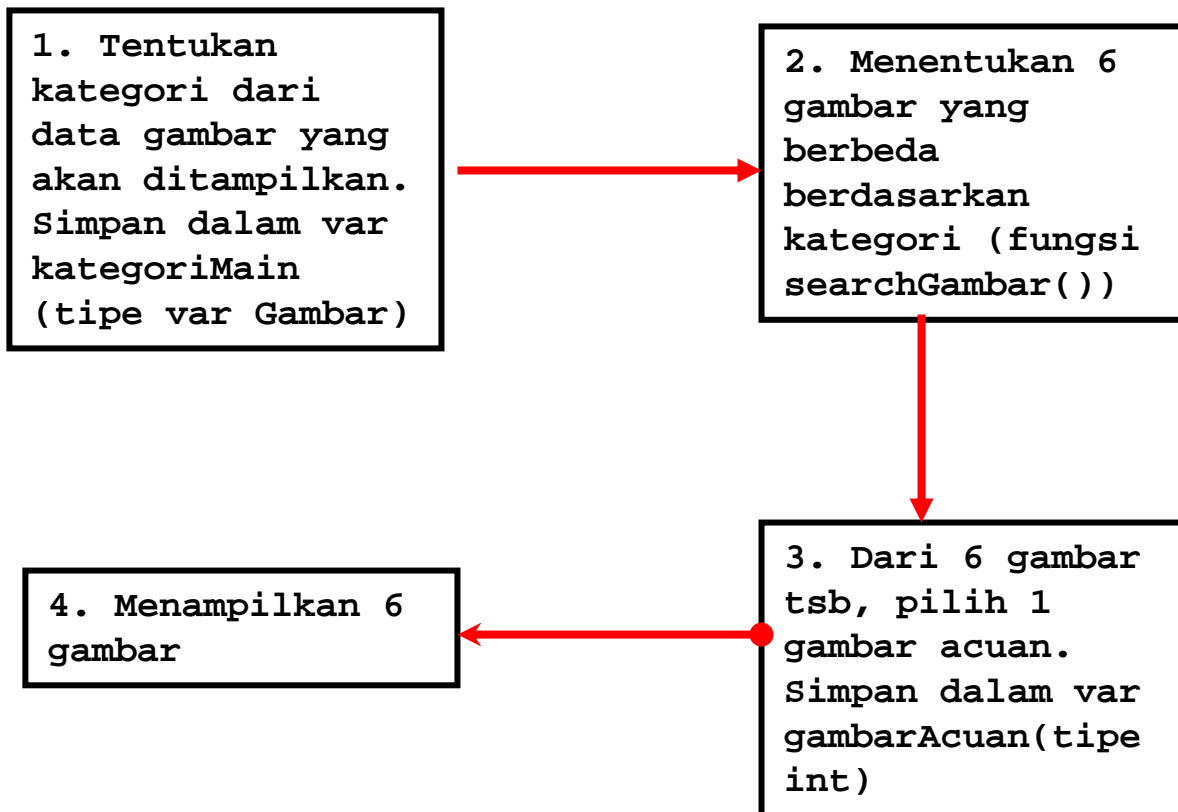
- isi dari **Vector**
dataGambar

0	<code>new Gambar(binatang, binatang1.jpeg);</code>
1	<code>new Gambar(binatang, binatang2.jpeg);</code>
2	<code>new Gambar(binatang, binatang3.jpeg);</code>
3	<code>new Gambar(binatang, binatang4.jpeg);</code>
4	<code>new Gambar(binatang, binatang5.jpeg);</code>
5	<code>new Gambar(binatang, binatang6.jpeg);</code>
6	<code>new Gambar(binatang, binatang7.jpeg);</code>
7	<code>new Gambar(binatang, binatang8.jpeg);</code>
8	<code>new Gambar(binatang, binatang9.jpeg);</code>
9	<code>new Gambar(binatang, binatang10.jpeg);</code>
10	<code>new Gambar(kartun,kartun1.jpeg);</code>
11	<code>new Gambar(kartun,kartun2.jpeg);</code>
12	<code>new Gambar(kartun,kartun3.jpeg);</code>
13	<code>new Gambar(kartun,kartun4.jpeg);</code>
14	<code>new Gambar(kartun,kartun5.jpeg);</code>
15	<code>new Gambar(kartun,kartun6.jpeg);</code>
16	<code>new Gambar(kartun,kartun7.jpeg);</code>
17	<code>new Gambar(kartun,kartun8.jpeg);</code>
18	<code>new Gambar(kartun,kartun9.jpeg);</code>
19	<code>new Gambar(kartun,kartun10.jpeg);</code>

3. Menampilkan 6 Gambar Berdasarkan Kategori

- Sebagai bantuan, terdapat 3 fungsi yaitu :
 - `private int[] searchGambar() {}`
untuk mendapatkan 6 data gambar yang akan ditampilkan untuk dipilih oleh user. Yang disimpan adalah no indeks dari data tsb.
 - `private int gambarAcuan() {}`
dari 6 data gambar yang diperoleh dari fungsi `searchGambar()`, diambil satu data gambar sebagai `gambarAcuan`.
 - `private void TampilGambar() {}`
fungsi ini bertugas untuk menampilkan ke objek `Jlabel`.

3. Menampilkan 6 Gambar Berdasarkan Kategori



3. Menampilkan 6 Gambar Berdasarkan Kategori (1)

- Pertama kali tentukan kategori dari data gambar yang akan ditampilkan. Simpan dalam variabel **kategoriMain (tipe Gambar)**
- Tentukan kategori yang akan ditampilkan adalah binatang.

```
kategoriMain = new  
Gambar( "binatang" , "binatang1.jpeg" );
```

3. Menampilkan 6 Gambar Berdasarkan Kategori (2)

```
private int[] searchGambar(){  
    int awal = dataGambar.indexOf(kategoriMain);  
    int akhir = dataGambar.lastIndexOf(kategoriMain);  
    . . .  
}
```

- Fungsi `indexOf` untuk mendapatkan index yang pertama kali pada objek `Vector dataGambar` yang **sama** (didefinisikan di fungsi `equals()`) dengan objek `kategoriMain`.
- Fungsi `lastIndexOf` untuk mendapatkan index yang terakhir kali yang **sama** (didefinisikan di fungsi `equals()`) dengan objek `kategoriMain`.
- Supaya fungsi `indexOf()` dan `lastIndexOf()` bisa digunakan maka **lakukan overriding pada fungsi equals()** pada class `Gambar`. Fungsi `equals()` ini, menghasilkan nilai `true` jika dua objek dari class `Gambar` mempunyai kategori yang sama, bernilai `false` jika dua objek dari class `Gambar` tidak mempunyai kategori yang sama.

```
public class Gambar {  
    private String kategori ;  
    private String namaGambar ;  
    public boolean equals(Object o){  
        // isi fungsi  
    }  
}
```

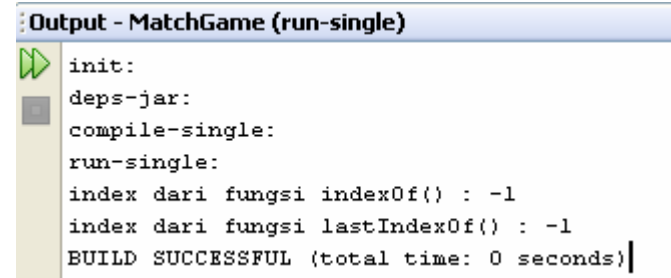

3. Menampilkan 6 Gambar Berdasarkan Kategori (2)

- Dibawah ini contoh penggunaan method `indexOf()` dan `lastIndexOf()` **tanpa** melakukan overriding terhadap method `equals(boolean)`
- hasil dari fungsi `indexOf()` adalah -1 yang artinya tidak menemukan objek yang sesuai dengan objek dengan kriteria yang dimaksud.
- Hasil sama seperti `lastIndexOf()` yaitu -1.

3. Menampilkan 6 Gambar Berdasarkan Kategori (2)

```
import java.util.Vector;
public class Test {
    public static void main(String args[]){
        Vector v = new Vector();
        Gambar gambarAcuan = new Gambar("kartun","k2.jpeg");
        v.add(new Gambar("binatang","b1.jpeg"));
        v.add(new Gambar("binatang","b2.jpeg"));
        v.add(new Gambar("binatang","b3.jpeg"));
        v.add(new Gambar("binatang","b4.jpeg"));
        v.add(new Gambar("kartun","k1.jpeg"));
        v.add(new Gambar("kartun","k2.jpeg"));
        v.add(new Gambar("kartun","k3.jpeg"));
        v.add(new Gambar("bunga","bg1.jpeg"));
        v.add(new Gambar("bunga","bg2.jpeg"));
        v.add(new Gambar("bunga","bg3.jpeg"));
        v.add(new Gambar("bunga","bg4.jpeg"));
        v.add(new Gambar("bunga","bg5.jpeg"));

        int index1 = v.indexOf(gambarAcuan);
        System.out.print("index dari fungsi indexOf() : " + index1 + "\n");
        int index2 = v.lastIndexOf(gambarAcuan);
        System.out.print("index dari fungsi lastIndexOf() : " + index2);
    }
}
```



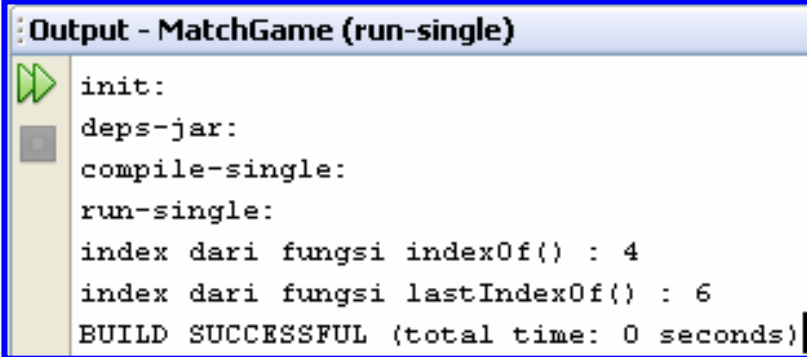
Output - MatchGame (run-single)

```
init:
deps-jar:
compile-single:
run-single:
index dari fungsi indexOf() : -1
index dari fungsi lastIndexOf() : -1
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Menampilkan 6 Gambar Berdasarkan Kategori (2)

- Selanjutnya lakukan overriding terhadap fungsi equals() pada class Gambar. Hasilnya akan tampak sebagai berikut:

```
public class Gambar {  
    private String kategori ;  
    private String namaGambar ;  
    public boolean equals(Object o){  
        // isi fungsi  
    }  
}
```



```
Output - MatchGame (run-single)  
init:  
deps-jar:  
compile-single:  
run-single:  
index dari fungsi indexOf() : 4  
index dari fungsi lastIndexOf() : 6  
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Menampilkan 6 Gambar Berdasarkan Kategori (2)

- Ambil 6 gambar secara acak mulai dari indeks **awal** (diperoleh dengan fungsi `indexOf()`) sampai dengan **akhir** (diperoleh dengan fungsi `lastIndexOf()`)

```
int r = (int)(Math.random() * (akhir-awal+1) + awal);
```

- Sehingga 6 data gambar yang disimpan adalah indeks diantara awal – akhir.
- Output dari fungsi `searchGambar()`, simpan dalam variabel **`int indexGambar[]`**
- Jika objek `kategoriMain` mempunyai kategori binatang maka variabel `awal = 0`, `akhir = 9`.
- Jika objek `kategoriMain` mempunyai kategori kartun maka `awal = 10`, `akhir = 19`.

3. Menampilkan 6 Gambar Berdasarkan Kategori (3)

- Dari 6 gambar tsb, pilih 1 gambar acuan. Simpan dalam var **gambarAcuan**(tipe int), lakukan secara random

3. Menampilkan 6 Gambar Berdasarkan Kategori (4)

- Untuk menampilkan 6 gambar gunakan fungsi `private void TampilGambar(int temp[]) {}` : fungsi ini bertugas untuk menampilkan 6 gambar ke objek JLabel.
- Caranya

```
String s2 = "/coba/buah1.jpeg";
```

```
jLabel1.setIcon(new  
javax.swing.ImageIcon(getClass().getResource(s2)));
```

4. User melakukan pemilihan

- Jika kursor user mengenai sebuah label maka background dari label menjadi berwarna biru (default berwarna pink – warna sesuai selera user). Untuk melakukan hal tersebut kita menggunakan event `MouseListener`.
- Klik kanan pada label pilih **Event** → pilih **Mouse** → pilih **`MouseEntered()`**, isi dengan program di bawah ini, lakukan hal yang sama untuk fungsi `MouseEntered` dengan warna yang berbeda.

```
private void gb5MouseEntered(java.awt.event.MouseEvent evt) {  
    gb5.setBackground(new Color(32,255,255));  
}
```

5. Tampilkan suara (benar atau salah)

- Pada praktikum ini mahasiswa membuat aplikasi tentang Match Game. User harus mencocokkan gambar acuan dengan 6 gambar yang ada dan harus memilih mana yang sama. Jika user benar dalam melakukan pencocokan maka tampilkan suara "Kamu Hebat" (suara sesuai keinginan), jika user salah maka tampilkan suara "Masih Salah Coba Lagi" (suara sesuai keinginan).