

BAHASA PEMROGRAMAN JAVA

Pertemuan XII

Kelas Utilitas

(Kelas Vector)

Oleh
Achmad Arrosyidi



TUJUAN PEMBELAJARAN

Mahasiswa dapat membedakan penggunaan Kelas Utilitas pada Pemrograman (Vector).



Kelas Vector

1. Konsep Vector
2. Metode Pada Kelas Vector
3. Deklarasi Vector
4. Penambahan Object Ke Vector
5. Pencetakan Object Dari Vector
6. Menghapus Object Di Dalam Vector
7. Operasi Matematika Pada Vector



1. KELAS VECTOR - KONSEP

- Kelas Vector merupakan kelas yang berada dalam paket Java.util.
- Kelas ini memungkinkan pengimplementasian array yang ukurannya dapat diubah sewaktu-waktu.
- Indeks vector juga dimulai dari 0. (sama seperti indeks pada array).
- Tipe Data pada elemen vector boleh tidak sama.



2. KELAS VECTOR - METODE

- Terdapat beberapa Methode pada kelas Vector, yaitu:

SYNTAX METHOD	KETERANGAN
<code>addElement(Object obj)</code>	menambahkan elemen pada vector
<code>capacity()</code>	kapasitas vector
<code>elementAt(int indeks</code>	mencetak elemen vector indeks ke-
<code>firstElement()</code>	mencetak elemen vector yang pertama
<code>lastElement()</code>	mencetak elemen vector yang terakhir
<code>indexOf(Object obj)</code>	mencetak indeks elemen vector berdasarkan objek (string/int)
<code>indexOf(Object element, int indeks)</code>	mencetak indeks elemen vector berdasarkan objek (string/int) dan indeks ke
<code>insertElementAt(Object obj, int indeks)</code>	menyisipkan elemen baru pada indeks
<code>remove(int index)</code>	menghapus elemen indeks ke
<code>removeElement(Object element)</code>	menghapus elemen object (string/int)
<code>size()</code>	jumlah elemen array

3. KELAS VECTOR – DEKLARASI

```
1 //file: DeklarasiVector.java
2 import java.util.*;
3 public class DeklarasiVector {
4     public static void main (String[] args) {
5         ..... Vector vektorku = new Vector();
6     }
7 }
```



4. KELAS VECTOR – PENAMBAHAN OBJECT

```
1 //file: PenambahanDataVectorDenganWarning.java
2 import java.util.*;
3 // Warning is On @SuppressWarnings("unchecked")
4 public class PenambahanDataVectorDenganWarning extends Thread{
5     public static void main (String[] args) {
6         Vector vektorku = new Vector ();
7         Vector vektorku2 = new Vector ();
8         vektorku.add("satu");
9         vektorku.add("3");
10        vektorku2.add("Sembilan");
11        vektorku.add(vektorku2);
12    }
13 }
```

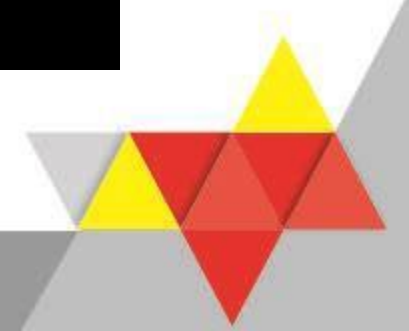
Console

```
NPP_SAVE: C:\java 19.2\M10\PenambahanDataVectorDenganWarning.java
CD: C:\java 19.2\M10
Current directory: C:\java 19.2\M10
javac PenambahanDataVectorDenganWarning.java"
Process started (PID=9888) >>>
Note: PenambahanDataVectorDenganWarning.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
<<< Process finished (PID=9888). (Exit code 0)
===== READY =====
```



4. KELAS VECTOR – PENAMBAHAN OBJECT

```
1 //file: PenambahanDataVector.java
2 import java.util.*;
3 @SuppressWarnings("unchecked")
4 public class PenambahanDataVector extends Thread{
5     public static void main (String[] args) {
6         Vector vektorku = new Vector ();
7         Vector vektorku2 = new Vector ();
8         vektorku.add("satu");
9         vektorku.add("3");
10        vektorku2.add("Sembilan");
11        vektorku.add(vektorku2);
12    }
13 }
```



5. KELAS VECTOR – PENCETAKAN OBJECT

```
1 //file: PencetakanDataVector.java
2 import java.util.*;
3 @SuppressWarnings("unchecked")
4 public class PencetakanDataVector {
5     public static void main (String[] args) {
6         Vector vektorku = new Vector ();
7         Vector vektorku2 = new Vector ();
8         vektorku.add("satu");
9         vektorku.add(3);
10        vektorku2.add("Sembilan");
11        vektorku.add(vektorku2);
12
13        for (int i=0; i<vektorku.size(); i++) {
14            System.out.println(vektorku.elementAt(i));
15            //mengambail nilai dari tiap elemen vektorku berdasarkan indeks ke-1
16        }
17    }
18 }
```

Console

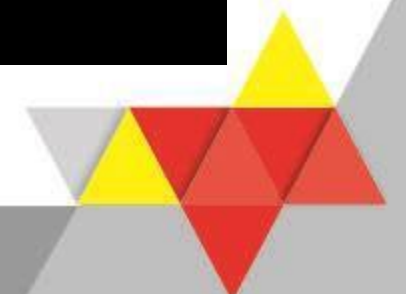
satu

3

[Sembilan]

<<< Process finished (PID=17312). (Exit code 0)

===== READY =====



5. KELAS VECTOR – PENAMBAHAN DAN PENCETAKAN

```
1 //file: Vector1.java
2 import java.util.Vector;
3 import java.util.Enumeration;
4 /*Kelas Enumeration adalah interface yang
5 menyediakan 2 buah metode yaitu.hasMoreElements()
6 dan.nextElement() */
7
8 public class Vector1 {
9     public static void main(String[] args) {
10         Vector kota = new Vector();
11         kota.addElement("Bandung");
12         kota.addElement("Surabaya");
13         kota.addElement("Jakarta");
14         kota.addElement("Medan");
15
16         Enumeration e = kota.elements();
17         while (e.hasMoreElements())
18             System.out.println(e.nextElement());
19     }
20 }
```

Console

Bandung

Surabaya

Jakarta

Medan

<<< Process finished (PID=13800). (Exit code 0)

===== READY =====



6. KELAS VECTOR – PENGHAPUSAN

```
1 //file: PenghapusanDataVector.java
2 import java.util.*;
3 @SuppressWarnings("unchecked")
4 public class PenghapusanDataVector {
5     public static void main (String[] args) {
6         Vector vektorku = new Vector ();
7         Vector vektorku2 = new Vector ();
8         vektorku.add("satu");
9         vektorku.add(3);
10        vektorku2.add("Sembilan");
11        vektorku.add(vektorku2);
12        for (int i=0; i<vektorku.size(); i++) {
13            System.out.println(vektorku.elementAt(i));
14            //mengambail nilai dari tiap elemen vektorku berdasarkan indeks ke-1
15        }
16        vektorku.remove("satu"); // Menghapus elemen yang berisi String "satu"
17        vektorku.removeElementAt(1); // Menghapus elemen urutan ke-2
18        System.out.println("\nVektorku setelah ada penghapusan elemen");
19        for (int i=0; i<vektorku.size(); i++){
20            System.out.println(vektorku.elementAt(i)); // Mengambil nilai dari tiap elemen vector vektorku
21        }
22    }
23 }
```

Console

satu
3
[Sembilan]

Vektorku setelah ada penghapusan elemen
3
<<< Process finished (PID=16680). (Exit code 0)
===== READY =====

6. KELAS VECTOR

PENGHAPUSAN DAN PENCETAKAN

```
1  import java.util.Vector;
2
3  public class Vector2 {
4      public static void main(String[] a) {
5          Vector <String> kota = new Vector<String>();
6          // Penambahan Kota secara berurutan
7          kota.addElement("Bandung");
8          kota.addElement("Surabaya");
9          kota.addElement("Jakarta");
10         kota.addElement("Medan");
11         tampilVektor("addElement", kota);
12         //Penambahan pada sembarang tempat
13         kota.insertElementAt("Kudus", 1);
14         tampilVektor("insertElement", kota);
15         //Penghapusan sebuah elemen
16         kota.removeElement("Bandung");
17         tampilVektor("removeElement : Bandung", kota);
18         //Hapus semua elemen
19         kota.removeAllElements();
20         tampilVektor("removeAllElement", kota);
21     }
```

```
22
23     static void tampilVektor(String info, Vector v) {
24         System.out.println(info);
25         System.out.println("=====");
26         if(v.isEmpty()) {
27             System.out.println("Vektor Kosong");
28         }
29         else {
30             System.out.println("Isi Vektor");
31             for(int i=0;i<v.size();i++) {
32                 System.out.println(v.elementAt(i));
33             }
34             System.out.println("=====");
35         }
36     }
37 }
```



6. KELAS VECTOR

PENGHAPUSAN DAN PENCETAKAN

Console

addElement

=====

Isi Vektor

Bandung

Surabaya

Jakarta

Medan

=====

insertElement

=====

Isi Vektor

Bandung

Kudus

Surabaya

Jakarta

Medan

=====

removeElement : Bandung

=====

Isi Vektor

Kudus

Surabaya

Jakarta

Medan

=====

removeAllElement

=====

Vektor Kosong

<<< Process finished (PID=18164). (Exit code 0)

===== READY =====



7. KELAS VECTOR

OPERASI MATEMATIKA PADA VECTOR

- Pada vector operasi matematika tidak bisa dilakukan secara langsung seperti pada variabel biasa.
- Operasi matematika baru dapat dilakukan jika sebelumnya dilakukan konversi tipe data terhadap elemen yang hendak dipakai untuk operasi matematika tersebut.
- Caranya adalah dengan mengkonversi objek pada elemen vector tersebut ke tipe data String menggunakan method `.toString()` kemudian baru dikonversi ke tipe data bilangan (int, double dan sebagainya).



7. KELAS VECTOR

OPERASI MATEMATIKA PADA VECTOR

```
1 //file: MatematikaVektor.java
2 import java.util.*;
3 @SuppressWarnings("unchecked")
4 public class MatematikaVektor {
5     public static void main (String [] args) {
6         Vector x = new Vector();
7         Vector y = new Vector();
8         //Menambah dan mencetak bilangan dari vector x
9         x.add(1);
10        System.out.println("Vector Bilangan Pertama : " + x.elementAt(0));
11        y.add(9);
12        System.out.println("Vector Bilangan Kedua : " + y.elementAt(0));
13        int tambah = Integer.parseInt(x.elementAt(0).toString()) + Integer.parseInt(y.elementAt(0).toString());
14        int kurang = Integer.parseInt(x.elementAt(0).toString()) - Integer.parseInt(y.elementAt(0).toString());
15        int kali = Integer.parseInt(x.elementAt(0).toString()) * Integer.parseInt(y.elementAt(0).toString());
16        double bagi = Integer.parseInt(x.elementAt(0).toString()) / Integer.parseInt(y.elementAt(0).toString());
17
18        System.out.println("Hasil Penjumlahan : " + tambah);
19        System.out.println("Hasil Pengurangan : " + kurang);
20        System.out.println("Hasil Perkalian : " + kali);
21        System.out.println("Hasil Pembagian : " + bagi);
22    }
23 }
```

Console

Vector Bilangan Pertama : 1
Vector Bilangan Kedua : 9
Hasil Penjumlahan : 10
Hasil Pengurangan : -8
Hasil Perkalian : 9
Hasil Pembagian : 0.0
<<< Process finished (PID=14340). (Exit code 0)
===== READY =====

**SELESAI
TERIMA KASIH**

