

ARRAY DAN ARRAYLIST

- Dapat menyimpan data bertipe objek maupun tipe primitif
- Memiliki ukuran yang tetap
- Dapat multi dimensi
- Tidak mendukung tipe generik
- Sebagai data struktur fundamental

STRUKTUR

ARRAY

Struktur data dengan ukuran tetap yang digunakan untuk menyimpan data dengan tipe yang sama

KESALAHAN UMUM

1. Salah inisiasi

```
int[] array;  
array = {1,2,3};
```

```
int[] array;  
array = new int[]{1,2,3};  
atau  
int[] array = new int[]{1,2,3};
```

2. Lupa import

```
import java.util.Arrays;
```

Beberapa method yang digunakan pada Array berasal dari `java.util.Arrays` class, sehingga perlu untuk di-import.

3. IndexOutOfBounds Exception

```
array[3] = 4;
```

Akan muncul `IndexOutOfBoundsException` karena indeks maksimal bernilai 2 (tidak ada elemen pada indeks ke-3).

- Hanya dapat menyimpan data bertipe objek
- Ukuran dapat diubah
- Tidak dapat multi dimensi
- Mendukung tipe generik
- Milik dari Collections framework

STRUKTUR

ARRAY LIST

Struktur data yang dapat menyimpan berbagai tipe data dengan ukuran yang fleksibel

KESALAHAN UMUM

1. Belum di inialisasi

```
ArrayList<String> arrayList;  
arrayList.add("");
```

```
ArrayList<TipeData> arrayList;  
arrayList = new ArrayList<>();  
arrayList.add(data);
```

2. IndexOutOfBoundsException

```
ArrayList<TipeData> arrayList;  
arrayList = new ArrayList<>();  
arrayList.get(0);
```

3. Lupa import

```
import java.  
util.ArrayList;
```

1 Inisiasi

```
TipeData[] array1;  
array1 = new  
TipeData[]{e11,e12};  
atau  
array2 = new TipeData[n];
```

2 Akses Elemen

```
array1[indeks];  
//mengakses elemen array pada indeks  
array1[0] = e1Baru;  
//mengubah elemen pada indeks ke-0 menjadi e1Baru
```

3 Menyalin elemen pada Array

```
TipeData[] arrayTujuan = Arrays.copyOf(array1, array1.length);  
//menyalin semua elemen pada array1 ke arrayTujuan  
atau
```

```
TipeData[] arrayTujuan = new TipeData[n];  
System.arraycopy(array1, 1, copiedInts, 0, 2);  
//menyalin elemen pada array1 dari indeks ke-1 ke arrayTujuan pada indeks ke-0,  
//dengan elemen yang disalin sebanyak 2 buah
```

5 Mengubah Array menjadi ArrayList (menyalin elemen ke ArrayList)

```
ArrayList<TipeData> lst;  
lst = new ArrayList<TipeData>(Arrays.asList(array1));  
//Menyalin semua elemen pada array1 ke ArrayList lst
```

4 Mencetak elemen pada Array

```
System.out.println(Arrays.toString(array1));
```

7 Mengurutkan elemen pada Array

```
Arrays.sort(array1);
```

6 Mendapatkan panjang Array

```
array1.length;
```

1 Akses Elemen

```
lst.get(indeks);  
//mengakses elemen lst pada indeks  
lst.add(elemen);  
//menambah elemen pada lst  
lst.set(indeks, elemenBaru);  
//mengubah elemen pada indeks menjadi e1Baru  
lst.remove(indeks);  
//menghapus elemen lst pada indeks
```

2 Inisiasi

```
ArrayList<TipeData> lst  
= new ArrayList<TipeData>();  
atau  
ArrayList lst = new ArrayList();
```

3 Menyalin elemen pada ArrayList

```
lstTujuan<TipeData> = new ArrayList<TipeData>(lst);  
//menyalin semua elemen pada lst ke lstTujuan
```

4 Mengubah ArrayList menjadi Array (menyalin elemen ke Array)

```
ArrayList<TipeData> lst = new ArrayList<>();  
lst.add(e11); lst.add(e12); lst.add(e13);  
TipeData[] arrTujuan = new TipeData[lst.size()];  
lst.toArray(arrTujuan);
```

5 Mencetak elemen pada ArrayList

```
System.out.println(lst);
```

7 Method pada java.util.Collections

```
import java.util.  
Collections;
```

```
Collections.sort(lst);  
//mengurutkan semua elemen pada lst  
Collections.shuffle(lst);  
//mengacak urutan elemen pada lst  
Collections.min(lst);  
//mendapatkan elemen terkecil pada lst  
Collections.max(lst);  
//mendapatkan elemen terbesar pada lst
```

6 Mendapatkan panjang ArrayList

```
lst.size();
```