PEMROGRAMAN BERORIENTASI OBYEK

Array

Pertemuan 3

Faisal Fajri Rahani, S.Si., M.Cs.
Teknik Informatika
2022

Guna array?

Menggunakan array untuk mengatur data.

Refrensi

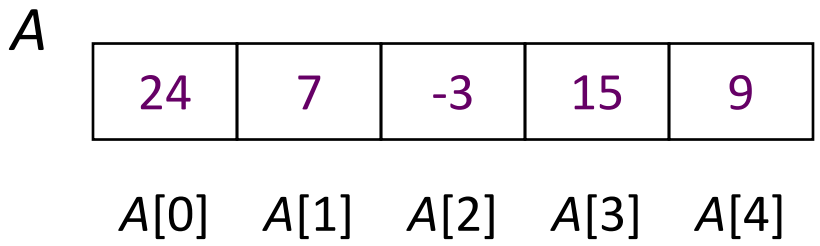
- **Data Abstraction and Problem Solving with Java: Walls and Mirror**
  - **Array:** Chapter 1, Section 1.1, pages 35 to 38
- http://www.comp.nus.edu.sg/ ~cs1020/2_resources/lectures.html

Array

Kumpulan data homogen

Pengantar

- Array adalah cara paling sederhana untuk menyimpan kumpulan data yang berjenis sama (homogen)
- Ini menyimpan elemennya dalam memori yang berdekatan
  - Indeks array dimulai dari nol
  - Contoh array bilangan bulat 5 elemen A dengan elemen yang diisi



Array in C (1/2)

```c
#include <stdio.h>
#define MAX 6

int scanArray(double [], int);
void printArray(double [], int);
double sumArray(double [], int);

int main(void) {
  double list[MAX];
  int size;

  size = scanArray(list, MAX);
  printArray(list, size);
  printf("Sum = %f\n",
         sumArray(list, size));

  return 0;
}
```

```c
// To read values into arr and return
// the number of elements read.
int scanArray(double arr[], int max_size) {
  int size, i;

  printf("How many elements? ");
  scanf("%d", &size);
  if (size > max_size) {
    printf("Exceeded max; you may only enter");
    printf(" %d values.\n", max_size);
    size = max_size;
  }
  printf("Enter %d values: ", size);
  for (i=0; i<size; i++) {
    scanf("%lf", &arr[i]);
  }
  return size;
}
```

sum_array.c

Array in C (2/2)

```
// To print values of arr
void printArray(double arr[], int size) {
  int i;

  for (i=0; i<size; i++)
    printf("%f ", arr[i]);
  printf("\n");
}

// To compute sum of all elements in arr
double sumArray(double arr[], int size) {
  int i;
  double sum = 0.0;

  for (i=0; i<size; i++)
    sum += arr[i];
  return sum;
}
```

sum_array.c

Array in Java (1/2)

- Di Java, array adalah sebuah objek.
- Setiap array memiliki atribut panjang bersifat publik (ini bukan method!)

TestArray1.java

```
public class TestArray1 {

  public static void main(String[] args) {
    int[] arr; // arr is a reference

    // buat array integer baru dengan 3 elemen
    // arr sekarang merujuk (poin) ke array baru ini
    arr = new int[3];

    // menggunakan panjang atribut
    System.out.println("Length = " + arr.length);

    arr[0] = 100;
    arr[1] = arr[0] - 37;
    arr[2] = arr[1] / 2;
    for (int i=0; i<arr.length; i++)
      System.out.println("arr[" + i + "] = " + arr[i]);
  }
}
```

Deklarasi array:
`datatype[] array_name`

Membuat sebuah array:
`array_name = new datatype[size]`

Mengakses elemen array individu.

```
Length = ?
arr[0] = ?
arr[1] = ?
arr[2] = ?
```

Array in Java (2/2)

- Sintaks loop alternatif untuk mengakses elemen array
- Ilustrasikan metode **toString ()** di kelas **Array** untuk mencetak larik

TestArray2.java

```
public class TestArray2 {
  public static void main(String[] args) {
    // Membuat dan inisialisasi array
    double[] arr = { 35.1, 21, 57.7, 18.3 };

    // menggunakan atribut panjang
    System.out.println("Length = " + arr.length);

    for (int i=0; i<arr.length; i++) {
      System.out.print(arr[i] + " ");
    }
    System.out.println();

    // Cara alternatif
    for (double element: arr) {
      System.out.print(element + " ");
    }
    System.out.println();
    System.out.println(Arrays.toString(arr));
  }
}
```

```
panjang = 4
35.1 21.0 57.7 18.3
35.1 21.0 57.7 18.3
[35.1, 21.0, 57.7, 18.3]
```

Syntax (enhanced for-loop):

```
for (datatype e: array_name)
```

Telusuri semua elemen dalam larik. “e" secara otomatis merujuk ke elemen array secara berurutan di setiap iterasi

Menggunakan toString() method in class Arrays

Array sebagai Parameter

- Referensi ke larik diteruskan ke sebuah method
  - Setiap modifikasi elemen dalam metode ini akan mempengaruhi array yang sebenarnya

TestArray3.java

```
public class TestArray3 {
  public static void main(String[] args) {
    int[] list = { 22, 55, 33 };

    swap(list, 0, 2);

    for (int element: list)
      System.out.print(element + " ");
    System.out.println();
  }

  // To swap arr[i] with arr[j]
  public static void swap(int[] arr, int i, int j) {
    int temp = arr[i]; arr[i] = arr[j]; arr[j] = temp;
  }
}
```

Jalan memutar : String[] di main() method

- main() method berisi parameter yang merupakan larik objek String
- Kita bisa menggunakan ini untuk argumen baris perintah

TestCommandLineArgs.java

```
public class TestCommandLineArgs {
  public static void main(String[] args) {

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

```
java TestCommandLineArgs Seri "Harry Potter" memiliki 7 buku.
args[0] = The
args[1] = Harry Potter
args[2] = series
args[3] = has
args[4] = 7
args[5] = books.
```

Mengembalikan Array

- Array bisa dikembalikan dari sebuah method

TestArray4.java

```
public class TestArray4 {
  public static void main(String[] args) {
    double[] values;

    values = makeArray(5, 999.0);

    for (double value: values) {
      System.out.println(value + " ");
    }
  }

  // Untuk membuat sebuah array dan mengembalikannya ke pemanggil
  public static double[] makeArray(int size, double limit) {
    double[] arr = new double[size];

    for (int i=0; i < arr.length; i++)
      arr[i] = limit/(i+1);

    return arr;
  }
}
```

Kesalahan Umum (1/3)

- length versus length()
  - Untuk mendapatkan panjang **`String`** object str, kita menggunakan **length()** method
    - Contoh: **`str.length()`**
  - Untuk mendapatkan panjang (ukuran) dari array arr, kita menggunakan atribut **length**
    - Contoh: **`arr.length`**
- Indeks larik di luar rentang
  - Hati-hati terhadap ArrayIndexOutOfBoundsException

```
public static void main(String[] args) {
  int[] numbers = new int[10];
  . . .
  for (int i = 1; i <= numbers.length; i++)
    System.out.println(numbers[i]);
}
```

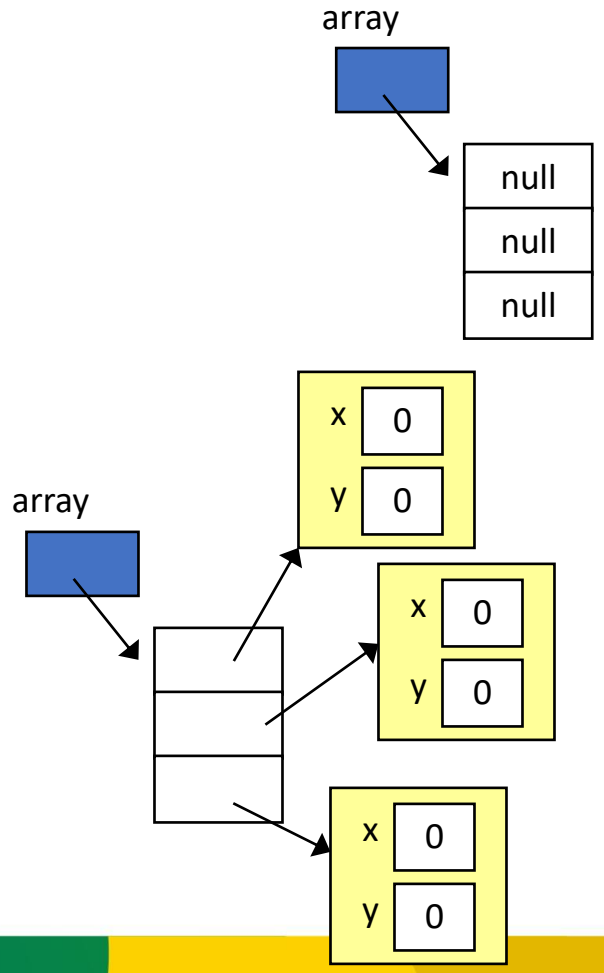
Kesalahan Umum (2/3)

- Jika Anda memiliki array of objects, sangat umum untuk lupa membuat instance objek array.
- Pemrogram sering membuat instance array itu sendiri dan kemudian berpikir bahwa mereka sudah selesai - itu mengarah ke java.lang.NullPointerException
- Contoh pada slide berikutnya
  - Ini menggunakan kelas Point di API
  - Refer to the API documentation for details

Kesalahan Umum (3/3)

```
Point[] array = new Point[3];
for (int i=0; i<array.length; i++) {
   array[i].setLocation(1,2);
}
```

Tidak ada objek yang dirujuk oleh array[0], array[1], and array[2], jadi bagaimana cara memanggil setLocation()?!



Kode yang diperbaiki:

```
Point[] array = new Point[3];
for (int i=0; i<array.length; i++) {
   array[i] = new Point();
   array[i].setLocation(1,2);
}
```

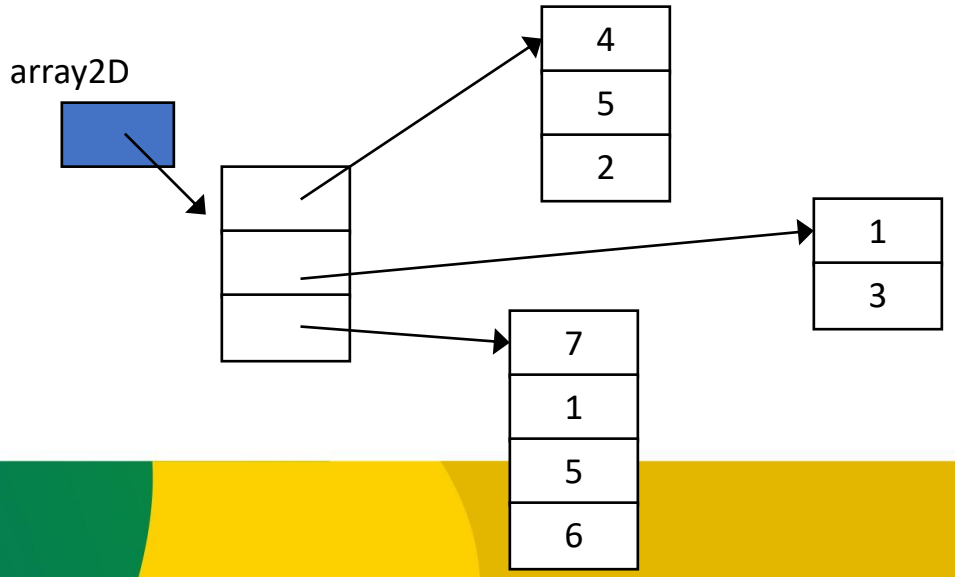
2D Array

2D Array (1/2)

- Larik dua dimensi (2D) adalah larik dari larik.
- Ini memungkinkan baris dengan panjang yang berbeda.

```
// an array of 12 arrays of int
int[][] products = new int[12][];
```

```
int[][] array2D = { {4,5,2}, {1,3},
                    {7,1,5,6} };
```



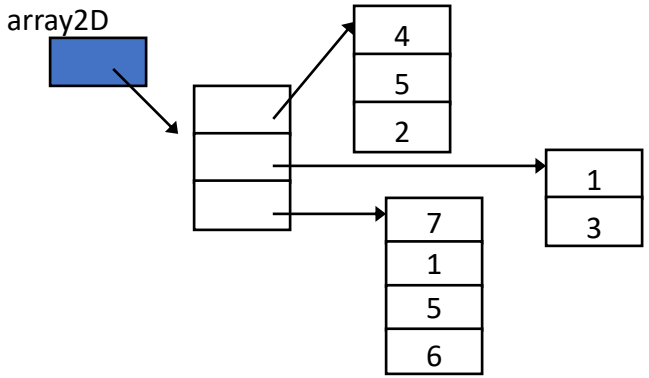
2D Array (2/2)

```
public class Test2DArray {
  public static void main(String[] args) {
    int[][] array2D = { {4, 5, 2}, {1, 3}, {7, 1, 5, 6} };

    System.out.println("array2D.length = " + array2D.length);
    for (int i = 0; i < array2D.length; i++)
      System.out.println("array2D[" + i + "].length = "
                         + array2D[i].length);

    for (int row = 0; row < array2D.length; row++) {
      for (int col = 0; col < array2D[row].length; col++)
        System.out.print(array2D[row][col] + " ");
      System.out.println();
    }
  }
}
```

Test2DArray.java




```
array2D.length = 3
array2D[0].length = ?
array2D[1].length = ?
array2D[2].length = ?
?
?
?
```

Kekurangan

- Array memiliki satu kelemahan utama :
  - Setelah diinisialisasi, ukuran array tetap
  - Rekonstruksi diperlukan jika ukuran larik berubah
  - Untuk mengatasi keterbatasan tersebut, kita dapat menggunakan beberapa kelas yang berhubungan dengan array
- Java memiliki class **Array**
  - Periksa dokumentasi API dan jelajahi sendiri
- Namun, kami tidak akan menggunakan banyak class Array; kami akan menggunakan beberapa kelas lain seperti **Vector** atau **ArrayList** (untuk dibahas nanti)
- Sebelum melakukan Vector/ArrayList, kami akan memperkenalkan konsep lain yang nanti disebut **Generics**

Terima Kasih