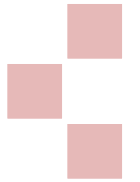
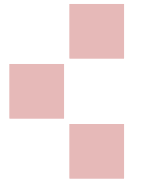


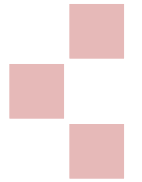
Java Array

Nurochman



Array

- **Arrays** are data structures consisting of related data items of the same type
- Arrays are fixed-length entities, they remain the same length once they are created,
- although an array variable may be reassigned such that it refers to a new array of a different length.
- Arrays are objects, so they are considered reference types.
- The elements of an array can be either primitive types or reference types.



Deklarasi dan membuat objek Array reference

- Deklarasi array kemudian membuat objek array

  String[] hari;

  hari= new String[7];

  atau (deklarasi dan membuat array sekaligus)

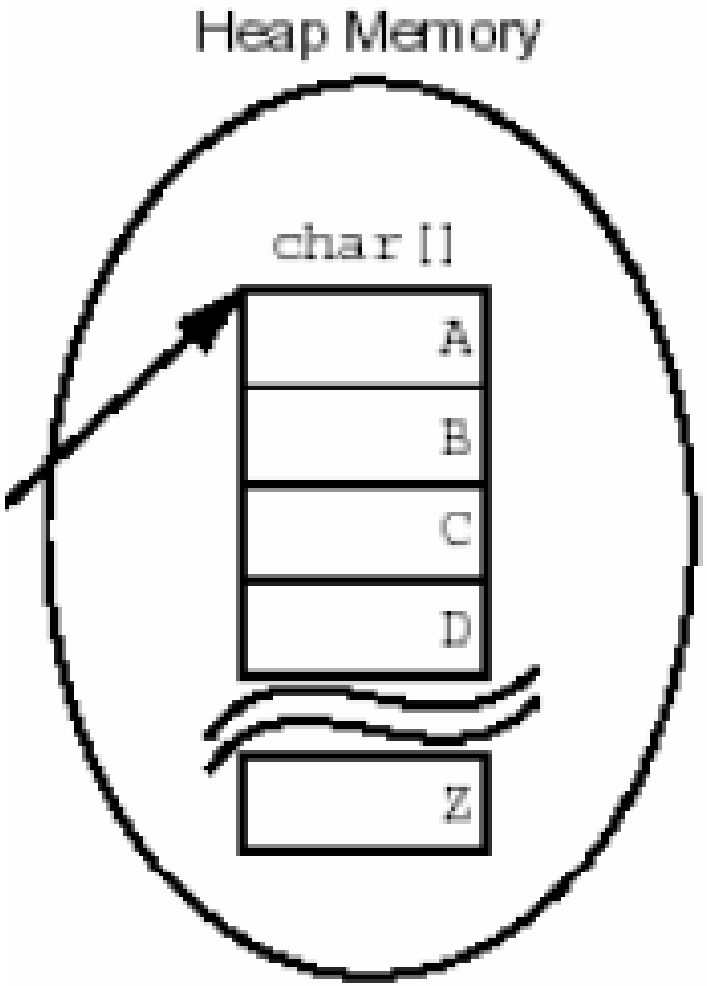
  String[] hari= new hari[7];

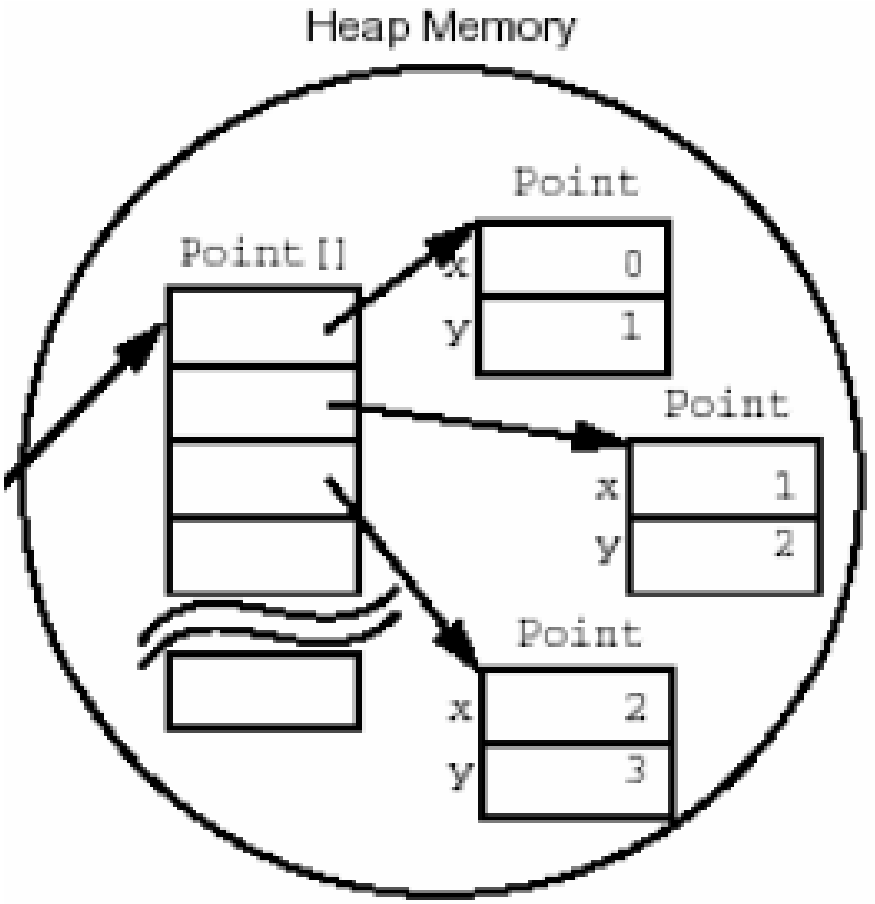
  atau (deklarasi array dg langsung menginisialisasi elemennya)

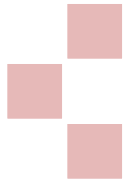
  String[] hari= new String[] {“senin”, “selasa”, …}

- Tanda [] dapat ditulis setelah nama identifier

Array primitif VS Array reference







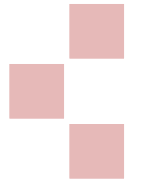
Mengakses Elemen Array

- arrBilPrima[0] = 1;
- System.out.println(arrBilPrima[0]);
- hari[0] = “senin”;
- System.out.println(hari[0]);

Catatan: indeks dari array dimulai dari 0.

- Panjang array dari hari

  hari.length -> return 7

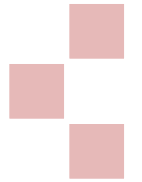


Merubah Elemen Array

- Kita dapat merubah elemen array dengan menyebutkan indeks dari elemen yg akan dirubah

  hari[5] = “jumat”;

  arrBilPrima[5] = 11;



Array Multidimensi

- Array dalam array
- Array yg menyimpan nilai pixel dari sebuah gambar:

  int[] gambar = new int[600][800];
- Array yg berisi 10 data Mhs (nim, nama)

  String[] mhs = new String[10][2];

Mengakses array multidimensi

- Mengisi data mhs pertama:

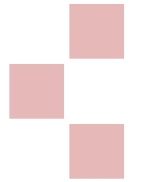
  mhs[0][0] = “08650001”

  mhs[0][1] = “Ahmad”;

- Mengisi data mhs kedua:

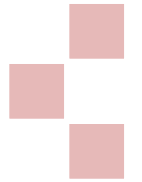
  mhs[1][0] = “08650002”

  mhs[1][1] = “Fatimah”;



Non-rectangular array of arrays

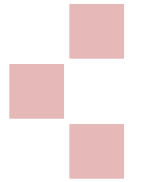
- int[][] piramida = new int[5][];
- Piramida[0] = new int[1];
- Piramida[1] = new int[2];
- Piramida[2] = new int[3];
- Piramida[3] = new int[4];
- Piramida[4] = new int[5];



Array Resizing

- Cannot resize an array
- Can use the same reference variable to refer to an entirely new array

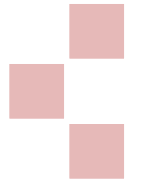
```
int[] arrBilPrima = new int[5];
arrBilPrima = new int[10];
```



Copying Array

- The System.arraycopy() method:

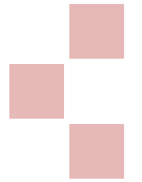
```
int[] prima1 = {1, 2, 3, 5, 7, 9};
int[] prima2 = new int[10];
System.arraycopy(prima1, 0, prima2, 0, prima1.length);
```



Tambahan

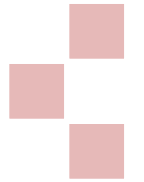
Lihat Modul JENI-Intro1-BAB-07-Java Array hal 1 - akhir

Pertanyaan???



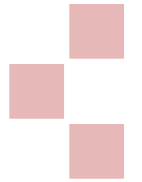
Arrays

- Arrays are objects.
  - Sort-of.
  - With special support in the language.
- Array syntax is very similar to C/C++
- Every array access is checked (at run time) to make sure it is within the bounds of the array.
  - arrays have fixed sizes (like in C/C++).



Array of what?

- You can have an array of anything, but all the elements of an array are the same type.
  - any primitive type.
  - any reference type.
- You need to declare the type of an array (the type of each element).
- You also need to explicitly create an array
  - just declaring the type is not enough.



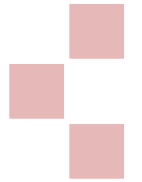
Array variable declaration

```
element_type [] variablename;
```

Examples:

```
int [] foo;
String [] studentNames;
```

These variable now exist, but there are not yet any arrays!



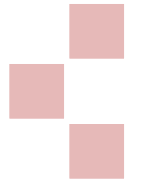
Creating an Array

- You use the Java **new** operator:

```
foo = new int [100];
studentNames = new String[20];
```

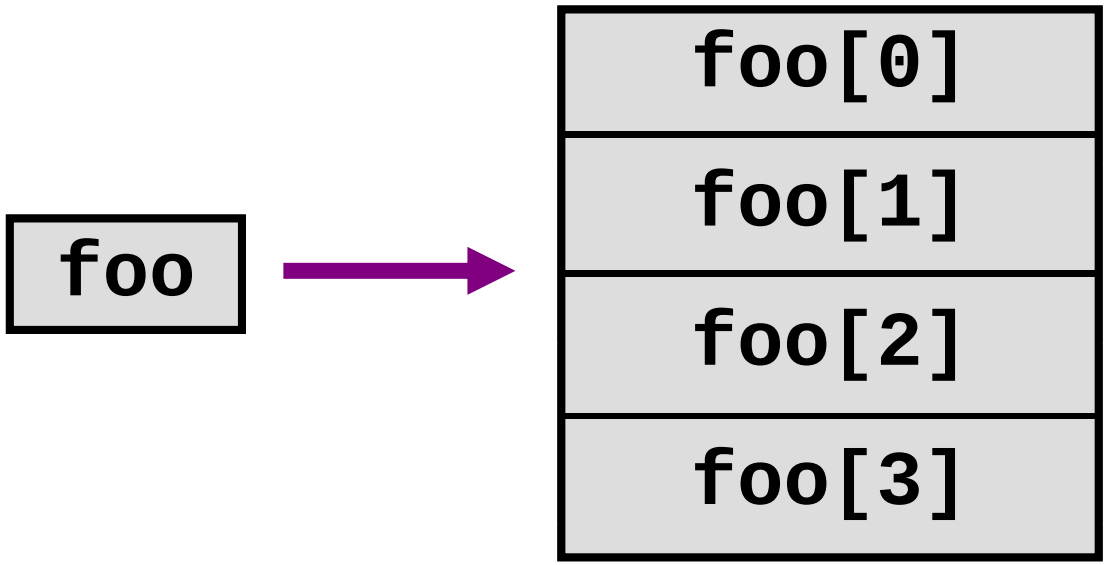
- You often see both declaration an creation like this:

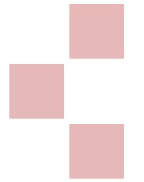
```
int [] foo = new int [100];
String [] studentNames = new String[20];
```



An important picture

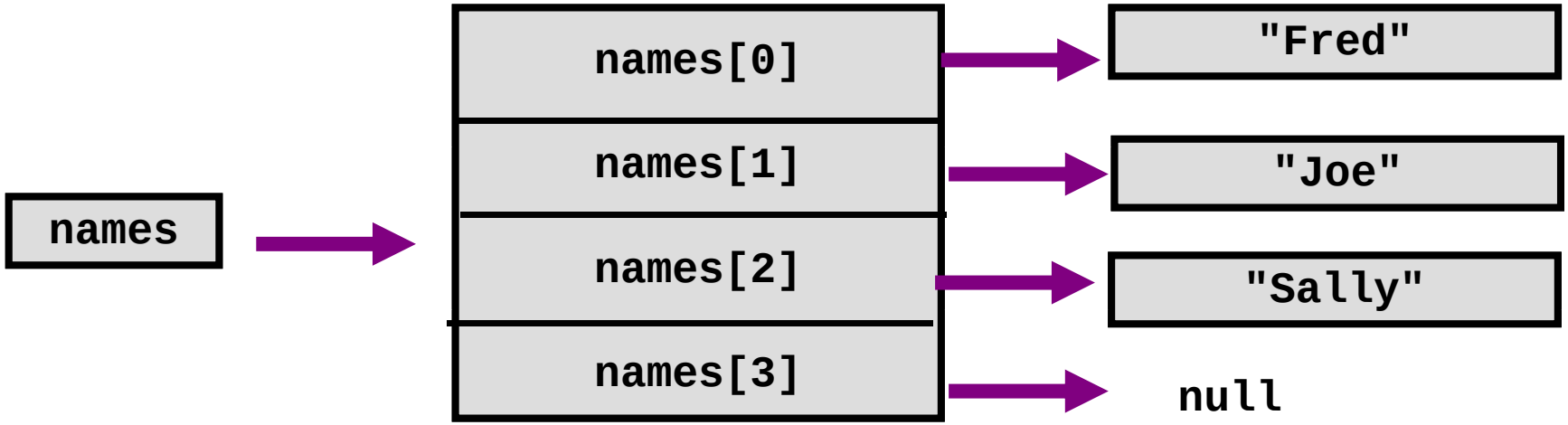
```
int [] foo = new int [4];
```

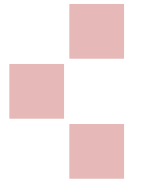




An even more important picture

```
String [] names = new String [4];
```





Array initializers

- You can use an *array literal* to create an array
  - you assign the literal to the array variable.

```
int [] foo = { 8, 2, 1, 4, 5 };
for (int i=0;i<foo.length;i++)
   System.out.println("foo["+i+"]="+foo[i]);
```