



**Universitas
Padjadjaran**

Array

diselenggarakan oleh

Erick Paulus, S.Si., M.Kom
Prodi Teknik Informatika

Jalan Raya Bandung Sumedang KM 21
Jatinangor Sumedang
www.fmipa.unpad.ac.id

From West Java for Indonesia to the World through SDGs

www.unpad.ac.id





Array

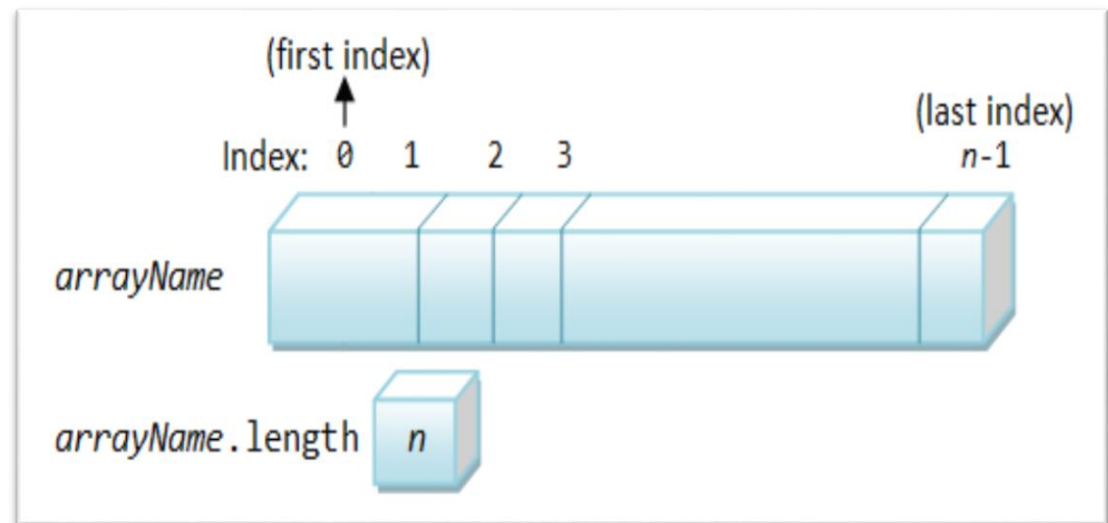
Array : Sekumpulan data (n elemen) dengan tipe data sama

Deklarasi :

- `tipe_array nama_array[];`
- `tipe_array[] nama_array;`

Contoh :

- `int nilai[];`
- `char[] huruf;`





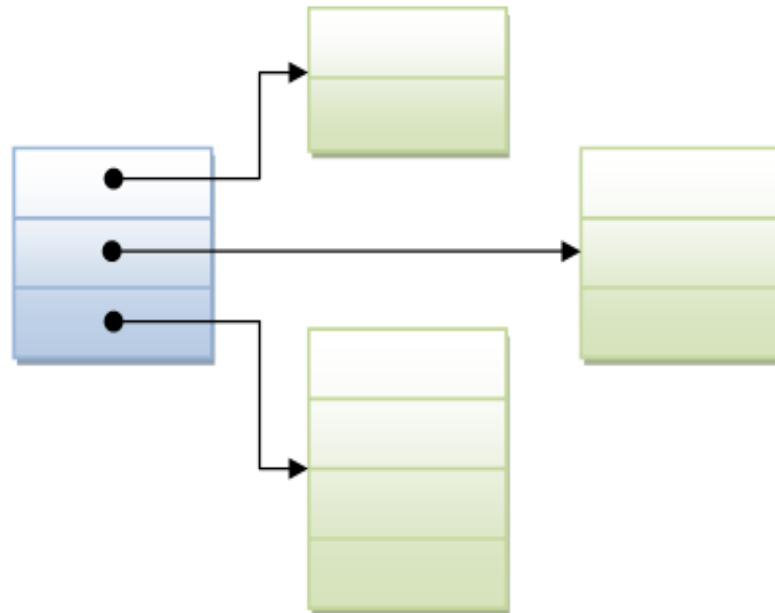
Instasiasi Array

```
nama_array = new tipe_array[total_elemen_array];  
int nilai[] = new int[5];
```



Array of arrays

- `int x[][] = new int[3][4];`





Parsing by value vs Ref

```
System.arraycopy(array1, p1, array2, p2, n);
```

```
int nilai[] = {10, 20, 30};  
int result[];  
result = nilai;
```



String Class

- String class merupakan salah satu contoh penerapan konsep array
- Class String memiliki method untuk menangani individual karakter dari suatu string, membandingkan string, mencari string, mendapatkan substring, membuat kopi isi string dan dijadikan lowercase atau uppercase.
- Java menyediakan operator spesial untuk menggabungkan dua string yaitu operator plus (+)



Method String

// Length

int length() // returns the length of the String

boolean isEmpty() // same as *thisString.length == 0*

// Comparison

boolean equals(String another)

// CANNOT use '==' or '!=' to compare two Strings in Java

boolean equalsIgnoreCase(String another)

int compareTo(String another)

// return 0 if this string is the same as another;

// <0 if lexicographically less than another; or >0

int compareToIgnoreCase(String another)

boolean startsWith(String another)

boolean startsWith(String another, int fromIndex) // search begins at fromIndex

boolean endsWith(String another)



Method String

```
// Searching & Indexing
int indexOf(String search)
int indexOf(String search, int fromIndex)
int indexOf(int character)
int indexOf(int character, int fromIndex)    // search
forward starting at fromIndex
int lastIndexOf(String search)
int lastIndexOf(String search, int fromIndex) // search
backward starting at fromIndex
int lastIndexOf(int character)
int lastIndexOf(int character, int fromIndex)
```




Method String

```
// Extracting a char or part of the String (substring)
char charAt(int index) // index from 0 to String's length - 1
String substring(int fromIndex)
String substring(int fromIndex, int endIndex) // exclude endIndex

// Creating a new String or char[] from the original (Strings are
immutable!)
String toLowerCase()
String toUpperCase()
String trim() // create a new String removing white spaces from front
and back
String replace(char oldChar, char newChar) // create a new String with
oldChar replaced by newChar
String concat(String another) // same as thisString + another
char[] toCharArray() // create a char[] from this string
void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin) //
copy into dst char[]
```



Method String

```
// Static methods for converting primitives to String
static String ValueOf(type arg)  // type can be primitives or char[]

// Static method resulted in a formatted String using format specifiers
static String format(String formattingString, Object... args)  // same
as printf()

// Regular Expression (JDK 1.4)
boolean matches(String regexe)
String replaceAll(String regexe, String replacement)
String replaceAll(String regexe, String replacement)
String[] split(String regexe)           // Split the String using
regexe as delimiter,                    // return a String array

String[] split(String regexe, int count) // for count times only
```



StringBuffer

- Class StringBuffer mirip dengan String tetapi bersifat mutable, atau dapat diubah atau dimodifikasi dengan menggunakan beberapa method yang dimilikinya.
- String buffer aman digunakan oleh beberapa thread. Method-methodnya bersifat synchronized sehingga beberapa operasi yang terjadi pada suatu obyek string buffer akan diselesaikan secara serial sesuai urutan pemanggilan.
- Operasi utama class StringBuffer adalah method append dan insert.



```
StringBuffer sb1 = new StringBuffer("start");  
sb1.append("le");
```

Maka isi sb1 adalah "startle"

```
StringBuffer sb2 = new StringBuffer("start");  
sb2.insert(4, "le");
```

Maka isi sb2 adalah "startlet"



String Builder

- StringBuilder adalah string yang bersifat mutable.
- Operasi yang dimiliki class StringBuilder mirip dengan class StringBuffer. Perbedaannya adalah StringBuilder tidak mendukung sinkronisasi.



Pertanyaan

- Apakah Array diinisialisasi 2 kali bisa

```
Int nilai[];
```

```
Nilai = new int[5];
```

```
Nilai = new int[10];
```



Pertanyaan

- StringBuffer apa lebih cepat dari String builder?