



PEMOGRAMAN JAVA

Yoannita, S.Kom

- **Tipe Data (ii) : Array**
- **Loncatan**



Array

- Misalkan dibutuhkan 100 variabel bertipe int untuk suatu perhitungan
- Array satu dimensi merupakan kumpulan dari variabel yang memiliki tipe data sama
- Pendeklarasian :
 - Tipe-data nama-array[];
 - Tipe-data[] nama-array;
- Contoh :
 - `int arr1[];` // atau :
 - `int[] arr2;`



Array

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

x[0]	x[1]	x[2]
------	------	------

```
x[0] = 20;
```

```
x[1] = 10;
```

```
x[2] = 0;
```

20	10	0
----	----	---

```
Nama-array = new tipe=data[ukuran-array]
```

```
int[] arr;
```

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



Array

```
class singleArray {  
    public static void main (String[] args) {  
        int[] x = new int[3];  
        x[0] = 20;  
        x[1] = 10;  
        x[2] = 0;  
        System.out.println ("Nilai x[0] : " + x[0]);  
        System.out.println ("Nilai x[1] : " + x[1]);  
        System.out.println ("Nilai x[2] : " + x[2]);  
    }  
}
```


Array Error

- Error yang menyebutkan **Java.lang.ArrayIndexOutOfBoundsException**

```
int [] x;
```

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

```
x = [10] = 100; -----> ERROR!
```

- Deklarasi array baru :

BENAR → **int[] x = new int [3];** atau

int[] x = {0,1,2}

ERROR -----> { **int [] x = new int [3];**
x = {0,1,2}



Panjang Dimensi Array

...

```
int x = {0,1,2}
```

```
System.out.print (x.length);
```

...

```
// akan mencetak angka 3
```

```
String y = "hallo";
```

```
System.out.println(y.length());
```



Loncatan

BREAK

- digunakan untuk menghentikan eksekusi dari suatu blok dan keluar dari blok yang sedang dieksekusi tersebut.
- sering digunakan dalam blok perulangan (while, do...while) dan blok kondisi (switch...case).

CONTINUE

- sering digunakan dalam perulangan, berfungsi untuk melanjutkan perulangan tanpa mengeksekusi baris-baris pernyataan setelah pernyataan continue. Pemakaian pernyataan continue biasanya bersama-sama dengan pernyataan kondisi (if)

RETURN

- digunakan dalam method (subroutine), Dengan menggunakan pernyataan return, maka eksekusi program akan kembali ke baris program yang memanggil method (subroutine) tersebut.



Contoh Penggunaan Break

```
// nama file : break1.java

class break1{
    public static void main(String[] args){
        char na = 'A';
        switch (na){
            case 'A' : System.out.println("Memuaskan");
                break;
            case 'B' : System.out.println("Baik");
                break;
            case 'D' : System.out.println("Cukup");
                break;
            case 'E' : System.out.println("Jelek");
                break;
            default :
                System.out.println("Nilai tidak ada");
        }
    }
}
```

```
// nama file : break2.java

class break2 {
    public static void main(String[] args){
        for (int a=5; a>0; a--){
            System.out.println("a = " + a );
            if (a==3)
                break;
        }
    }
}
```

berikut hasil dari program di atas.

a = 5

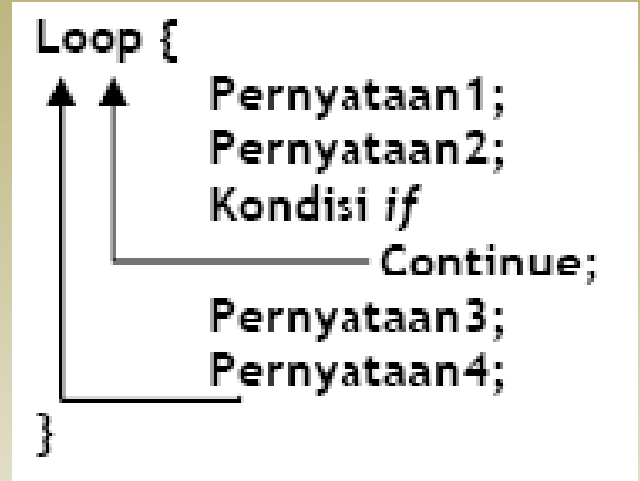
a = 4

a = 3

Contoh Penggunaan Continue

```
// nama file : continue1.java
```

```
class continue1{  
    public static void main(String[] args) {  
        for (int a=0 ; a < 10 ; a++){  
            if (a % 3==0) {  
                continue;  
            }  
            System.out.println("a = " + a );  
        }  
    }  
}
```



berikut hasil dari program

a = 1

a = 2

a = 4

a = 5

a = 7

a = 8



Contoh Penggunaan Return

```
// nama file : return1.java
```

```
class return1{  
    public static int hitung(int a){  
        return(a*a);  
    }  
  
    public static void main (String args[]){  
        int b = hitung(3);  
        System.out.println("cara satu = " + b);  
        System.out.println("cara dua = " + hitung(5));  
    }  
}
```

Hasil program di atas.
cara satu = 9
cara dua = 25