

# PERULANGAN PEMROGRAMAN JAVA

## PERTEMUAN 4

## Pernyataan Perulangan

Pernyataan Perulangan digunakan untuk melakukan proses yang sifatnya mengulang pada pemrograman java. Untuk keperluan perulangan proses, Java menyediakan beberapa perintah perulangan, yaitu: for, while dan do-while.

### Pernyataan for

Perulangan yang pertama adalah for. Bentuk umum pernyataan for sebagai berikut :

for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )

Bila pernyataan didalam for lebih dari satu maka pernyataan-pernyataan tersebut harus diletakan didalam tanda kurung.

```
for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )  
{  
    pernyataan / perintah;  
    pernyataan / perintah;  
    pernyataan / perintah;  
}
```

## Pernyataan Perulangan(lanjutan)

Kegunaan dari masing-masing argumen for diatas adalah :

- Inisialisasi : merupakan bagian untuk memberikan nilai awal untuk variabel-variabel tertentu.
- Syarat Pengulangan : memegang kontrol terhadap pengulangan, karena bagian ini yang akan menentukan suatu perulangan diteruskan atau dihentikan.
- Pengubah Nilai Pencacah : mengatur kenaikan atau penurunan nilai pencacah.

Contoh :

Sebagai contoh program untuk mencetak bilangan dari 1 hingga 10 secara menaik, secara selengkapnya seperti dibawah ini:

```
3
4  /**
5   *
6   * @author rony
7   */
8  public class contohFor {
9      public static void main(String[] args)
10 {
11     int a;
12     for(a = 1; a <= 10; ++a)
13         System.out.println(a);
14 }
15 }
```

# Pernyataan nested - for

Pernyataaan Nested for adalah suatu perulangan for didalam perulangan for yang lainnya. Bentuk umum pernyataan Nested for sebagai berikut :

```
for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )  
{  
  for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )  
  {  
    pernyataan / perintah;  
  }  
}
```

# Contoh Program nested - for

Didalam penggunaan nested-for, perulangan yang didalam terlebih dahulu dihitung hingga selesai, kemudian perulangan yang diluar diselesaikan. Sebagai contoh program menerapkan perintah nested - for, selengkapnya seperti dibawah ini:

```
0
7 /**
8  *
9  * @author rony
10 */
11 public class NestedFor {
12     public static void main(String[] args)
13     {
14         int a, b;
15
16         for(a = 1; a <= 5; a++)
17         {
18             System.out.println();
19             for(b = a; b <= 5; b++)
20             {
21                 System.out.print(a);
22             }
23         }
24         System.out.println("\n");
25     }
26 }
```

## Pernyataan while

Pernyataan perulangan while merupakan instruksi perulangan yang mirip dengan perulangan for. Bentuk perulangan while dikendalikan oleh syarat tertentu, yaitu perulangan akan terus dilaksanakan selama syarat tersebut terpenuhi. Bentuk umum perulangan while, sebagai berikut :

```
while ( syarat )  
Pernyataan / perintah ;
```

Bentuk umum perulangan while, dengan lebih dari perintah / pernyataan, sebagai berikut :

```
while ( syarat )  
{  
Pernyataan / perintah ;  
Pernyataan / perintah ;  
}
```

# Contoh Kode Program While

```
7  /**
8   *
9   * @author rony
10  */
11  public class ContohWhile {
12      public static void main(String[] args)
13      {
14          int bil=10;
15          while(bil>=1)
16          {
17              System.out.println(bil);
18              --bil;
19          }
20      }
21  }
```

## Pernyataan do-While

Pernyataan perulangan do - while merupakan bentuk perulangan yang melaksanakan perulangan terlebih dahulu dan pengujian perulangan dilakukan dibelakang. Bentuk umum perulangan do - while, sebagai berikut :

```
do  
pernyataan / perintah ;  
while ( syarat );
```

Bentuk umum perulangan do - while, dengan lebih dari perintah / pernyataan, sebagai berikut:

```
do  
{  
Pernyataan / perintah ;  
Pernyataan / perintah ;  
}  
while ( syarat );
```



# Contoh Kode Program Do While

```
7  /**
8   *
9   * @author rony
10  */
11  public class ContohDoWhile {
12      public static void main(String[] args)
13      {
14          int bil=1;
15          do
16          {
17              System.out.println(bil);
18              ++bil;
19          }
20          while(bil<=10);
21      }
22  }
```

# ARRAY PEMROGRAMAN JAVA

# Apa itu Array?

Array merupakan kumpulan data pada sebuah variabel dengan tipe data yang sama

Array adalah sekelompok data sejenis yang disimpan ke dalam variabel dengan nama yang sama, dengan memberi indeks pada variabel untuk membedakan antara yang satu dengan yang lain.

## ***VARIABEL ARRAY***

nama\_variabel[indeks]

ketentuan nama variabel array sama dengan nama variabel biasa.  
indeks menunjukkan nomor dari variabel .

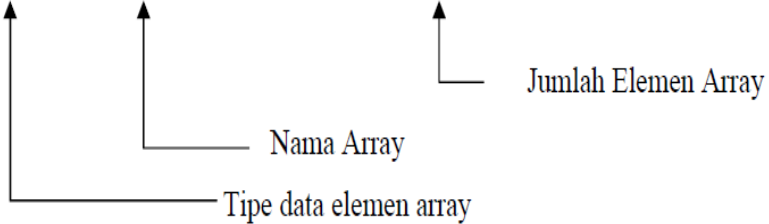
## ***DEKLARASI VARIABEL ARRAY***

BU : tipe nama\_variabel[indeks];

Contoh : float bil[10];

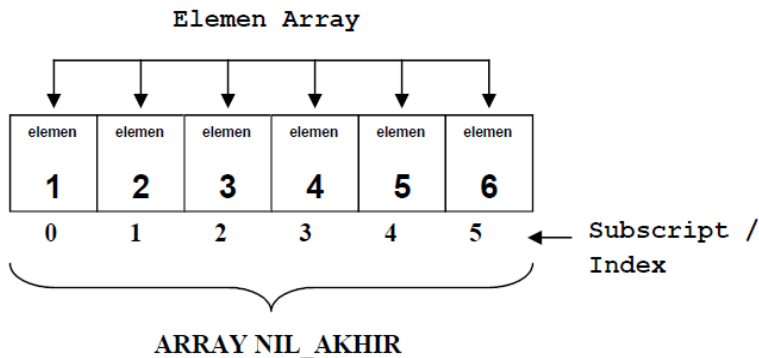
# DEKLARASI VARIABEL ARRAY

```
int[] nil_akhir = new int[6];
```



Contoh Pendeklarasian Array sekaligus membuat objek array.

Pada pendeklarasian diatas, variabel `nil_akhir` sebagai array-of-int, banyak elemen yang dapat ditampung sebanyak 6 (enam) elemen. Suatu array dapat digambarkan sebagai kotak panjang yang berisi kotak-kotak kecil didalam kotak panjang tersebut.



# Contoh Program Array

```
0
7 import java.util.Scanner;
8
9 /**
10  *
11  * @author rony
12  */
13 public class LatArray1 {
14     public static void main(String[] args)
15     {
16         int i;
17         int[] nil_akhir;
18         // deklarasi variabel array
19         nil_akhir = new int[6]; // membuat objek array
20         Scanner input = new Scanner(System.in);
21         for(i=0; i<6; i++)
22         {
23             System.out.print("Masukkan Array ke " + i + " = ");
24             nil_akhir[i] = input.nextInt();
25         }
26         System.out.println("\n\nData Yang Diinput ke Elemen Array \n");
27         for(i=0; i<6; i++)
28         {
29             System.out.print("Nilai Akhir Index " + i );
30             System.out.println(" = " + nil_akhir[i]);
31         }
32     }
33 }
```

Sebagai contoh program memasukan data kedalam elemen array dan mengambil data dari dalam elemen array, secara selengkapnya seperti dibawah ini:

## Inisialisasi Array Berdimensi Satu

```
6
7 /**
8  *
9  * @author rony
10 */
11 public class InisialisasiArray1 {
12     public static void main(String[] args)
13     {
14         int i;
15         double[] nil_akhir = {56.5, 66.7, 87.6, 98.5, 78.9, 85.4};
16
17         System.out.println("\nData Yang Diinput ke Elemen Array \n");
18         //menampilkan data dari elemen array
19         for(i=0; i<6; i++)
20         {
21             System.out.print("Nilai Akhir Index " + i );
22             System.out.println(" = " + nil_akhir[i]);
23         }
24     }
25 }
```

Sebagai contoh program memasukan data kedalam elemen array dengan cara inisialisasi objek array dan mengambil data dari dalam elemen array, secara selengkapnya seperti dibawah ini:

Inisialisasi adalah memberikan nilai awal terhadap suatu variabel. Bentuk pendefinisian suatu array dapat dilihat dari contoh berikut :

type\_data[] nama\_array = {  
nilai array};

Contoh:

float[] nilai= {56.5, 66.7,  
87.45, 98.5, 78.9};

# Array dimensi dua

Array dimensi dua tersusun dalam bentuk baris dan kolom, dimana indeks pertama menunjukkan baris dan indeks kedua menunjukkan kolom. Array dimensi dua dapat digunakan seperti pendataan penjualan, pendataan nilai dan lain sebagainya.

Bentuk Umum pendeklarasian array :

Deklarasi Array Bentuk Umum pendeklarasian array :

**tipe\_data[][] nama\_var\_array;**

**nama\_var\_array = new tipe\_data[ukuran\_baris][ukuran\_kolom];**

Atau bisa juga secara langsung seperti dibawah ini :

**tipe\_data[][] nama\_var\_array = new tipe\_data[ukuran\_baris][ukuran\_kolom];**

Keterangan :

Type Data : Untuk menyatakan type data yang digunakan.

Ukuran\_baris : Untuk menyatakan jumlah maksimum elemen baris dalam array.

Ukuran\_kolom : Untuk menyatakan jumlah maksimum elemen kolom dalam array.

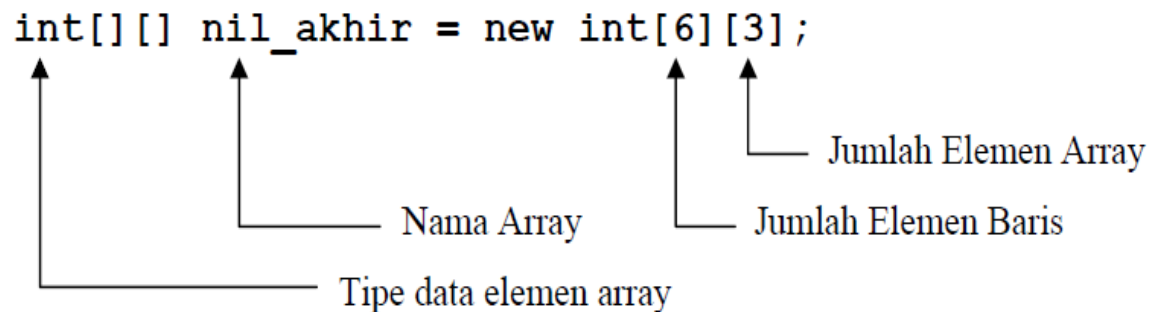
# Pendeklarasikan Array

Berikut contoh pendeklarasian array dan membuat objek array;

```
int[][] nil_akhir;  
nil_akhir = new int[6][3];
```

Contoh Pendeklarasian Array sekaligus membuat objek array.

```
int[][] nil_akhir = new int[6][3];
```



Tipe data elemen array

Nama Array

Jumlah Elemen Baris

Jumlah Elemen Array



## Contoh Program Array

Sebagai contoh program memasukkan data kedalam elemen array berdimensi dua dan mengambil data dari dalam elemen array berdimensi dua, secara selengkapnya seperti dibawah ini:

```
4 import java.util.Scanner;
5 /**
6  *
7  * @author rony
8  */
9 public class LatArray2 {
10     public static void main(String[] args)
11     {
12         int i, j;
13         int[][] data_jual;
14         data_jual = new int[3][3];
15         Scanner input = new Scanner(System.in);
16         for(i=0; i<3; i++)
17         {
18             for(j=0; j<3; j++)
19             {
20                 System.out.print("Masukkan Data Jual ke ["+i+"] ["+j+"] =");
21                 data_jual[i][j] = input.nextInt();
22             }
23             System.out.println();
24         }
25         System.out.println("\n\nData Jual Yang Diinput ke Elemen Array \n");
26         for(i=0; i<3; i++)
27         {
28             for(j=0; j<3; j++)
29             {
30                 System.out.print("Nilai Data Jual ["+i+"] ["+j+"] ");
31                 System.out.println(" = " + data_jual[i][j]);
32             }
33             System.out.println();
34         }
35     }
36 }
```

## Inisialisasi Array Berdimensi Dua

Suatu array, dapat diakses dengan menggunakan subscript atau indexnya. Bentuk umum pengaksesan dengan bentuk :

```
tipe_data[][] nama_array = { nilai array };
```

Contoh:

```
int[][] nil_akhir = {{150, 159, 230}, {100,125,150}, {210,125,156}};
```

Sebagai contoh program memasukan data kedalam elemen array dengan cara inisialisasi objek

array dan mengambil data dari dalam elemen array, secara selengkapnya seperti dibawah ini:

```
0
7  /**
8   *
9   * @author rony
10  */
11  public class InisialisasiArray2 {
12      public static void main(String[] args)
13      {
14          int i, j;
15          int[][] nil_akhir = {{150, 159, 230}, {100,125,150}, {210,125,156}};
16
17          System.out.println("Data Yang Diinput ke Elemen Array \n");
18          //menampilkan data dari elemen array
19          for(i=0; i<3; i++)
20          {
21              for(j=0; j<3; j++)
22              {
23                  System.out.print("Nilai Akhir Index ["+i+"] ["+j+"] = ");
24                  System.out.println(" = " + nil_akhir[i][j]);
25              }
26          }
27      }
28  }
```