Belajar Cara Memahami Konsep Dasar Bahasa Program Java

**A). Sekilas tentang Java**

Java merupakan bahasa pemrograman tingkat tinggi (high level) , artinya bahasa ini mudah dipahami oleh manusia pada umumnya , karena menggunakan bahasa sehari-hari manusia. Java dulu dibuat oleh perusahaan **Sun Microsystems** , yang sekarang dimiliki oleh perusahaan**Oracle.** Bahasa java mirip dengan bahasa C++ ,  namun dikembangkan agar mudah dipahami dan dipakai.Java menggunakan konsep OOP (Object Oriented Programming) ,artinya semua inti program dari Java yaitu mengacu ke sebuah Obyek. Java termasuk platform independent , yang artinya kalian bisa menjalankan dengan berbagai Macam OS(Sistem Operasi ) dan arsitektur komputer. Seperti pada perangkat Android dari Google ,dan Beberapa Aplikasi Desktop ,seperti program AntiVirus , program media player, Aplikasi Enterprise dll.

**B). Keunggulan Java**

* **Mudah dipelajari**

Java memiliki salah satu keunggulan , yaitu mudah dipelajari oleh semua orang , karena syntax (tata bahasa ) yang mirip dengan bahasa manusia. Dengan banyaknya artikel atau tutorial yang membahas bahasa pemrograman Java di  Internet. Semakin memudahkan kita untuk pemula , untuk mempelajari semua hal yang ada di bahasa pemrograman Java.

* **Merupakan bahasa OOP**

Salah satu alasan ,yang membuat Java begitu populer karena mempunyai konsep bahasa *(OOP) Object Oriented Programming.*Oleh karena itu , untuk mengembangkanya jauh lebih mudah , serta menjaga system tetap modular, flexible and extensible.

* **Banyaknya API**

Java memiliki banyaknya  *API (Application programming interface ) ,*yang siap dikembangkan untuk keperluan para programer Java.

**C).  Konsep OOP :**

Konsep di OOP (*Object Oriented Programming*) , memiliki beberapa Konsep salah satunya:

* **Class**merupakan sebuah kerangka/model yang berfungsi untuk tempat menaruh,  dan mendeksripsikan variabel, method (perilaku ) dari sebuah obyek. Penamaan nama class diprogram ,harus sama dengan nama di struktur file  extention **.java**. jadi jangan salah. Contoh : class binatang ,class mobil ,class buah dll.

* **Object**merupakan sebuah representasi dan instance dari Class. Object adalah sebuah inti dan wujud real dari sebuah Class. Contoh : dari class binatang ,kita bisa mengambil obyek , yaitu ,kucing,ayam,anjing dll.

* **Atribute**merupakan sebuah unsur data yang ada di class, Atribut biasanya terdiri dari sebuah data,variabel,propertie dan field. Contoh dari class binatang bntng  , class mobil mbl ,dll.

* **Method** merupakan sebuah perilaku (behavior) dari sebuah class. Method terbagi menjadi dua jenis ,method void dan methon non void. Method void artinya method yang tidak mengembalikan nilai. Method non void sebaliknya method yang mengembalikan nilai. Contoh method void seperti : public static void main(String[] args)

|  |
| --- |
| [Contoh Atribute dan Behavior Java](https://2.bp.blogspot.com/-sx2UaUHi7wk/VwckJL-5v5I/AAAAAAAAFRM/Uto_NaYfXecathpf6cT254xTttBhuvKRw/s1600/Contoh-Atribute-dan-Behavior-Java.png) |
| Atribute dan Behavior |

* [**Encapsulation**](http://www.okedroid.com/2016/06/belajar-encapsulation-di-bahasa-pemrograman-java.html)(Enkapsulasi)adalah suatu mekanisme membungkus suatu data (variabel) , agar tidak dapat di akses oleh class lain, dengan menggunakan modifier private atau protected (untuk Class Turunan).Di dalam konsep ini beberapa variabel akan disembunyikan oleh class lain ,dan hanya bisa di akses di main class ,dengan menggunakan method modifier public.
* [**Inheritance**](http://www.okedroid.com/2016/06/belajar-inheritance-pewarisan-bahasa-pemrograman-java.html)(Pewarisan)adalah suatu proses dimana , suatu class yang bisa disebut*super class ,*dapat mewarisi sifat turunan ke dalam class turunanya yaitu *sub class*. Super class akan mewarisi nilai dari atribut atau behavior ke Class turunananya.
* [**Polymorphism**](http://www.okedroid.com/2016/06/belajar-polymorphism-overriding-overloading-di-bahasa-pemrograman-java.html) (Banyak bentuk )adalah suatu kemampuan yang dimiliki sebuah method ,yang memiliki nama sama ,namun dengan perilaku yang berbeda-beda. Kemampuan objek agar melakukan perilaku atau tindakan  yang secara konsep sama namun dengan cara yang berbeda-beda.

**D)**. **Modifier**

**Modifier** merupakan sebuah ijin hak akses untuk penggunaan suatu atribut atau method, pada suatu Package dan Class. Terdapat empat 4 jenis Modifier yaitu public ,private,protected ,no modifier. Contohnya nya kalian bisa melihat gambar berikut:

|  |
| --- |
| [Modifier Akses Java](https://3.bp.blogspot.com/-QSuEAsAcDkU/VwYJtviuBdI/AAAAAAAAFQ8/JTFhtzmg_HIMzbHoOuyJmui-mDqiY7hgg/s1600/jenis-jenis-modifier-akses-java.png) |
| Modifier Akses Java |

* **Public**dapat di akses dari class yang sama,package yang sama ,subclass ,dan package lain.
* **Protected**dapat diakses dari class yang sama ,package yang sama ,subclass, tapi tidak dapat diakses dari package lain.
* **Private** hanya dapat di akses oleh class yang sama.
* **No Modifiers** dapat diakses dari class dan package yang sama.

**Contoh Program Java :**

*latihan1.java*

2. **package** latihan1;  *// nama package latihan1*

5. **import** latihan2.latihan2;
6. *//import class bernama latihan dua dari package lain bernama latihan2*

9. ***/\*\****
10. ***\* @author FATHUR (okedroid.com)***
11. ***\*/***
12. **public** **class** latihan1 { *//nama class latihan1 dengan modifier public*
14. ***/\*\****
15. ***\* @param args the command line arguments***
16. ***\*/***
17. **public** **static** **void** main(String[] args) {
19. latihan2 lth = **new** latihan2(); *//setiap statement selalu diakhiri ;*
20. *//membuat obyek baru dengan variabel lth dari class latihan2*
21. *//yang berada di package lain bernama package latihan2*
23. lth.tampilhello();*//setiap statement selalu diakhiri ;*
24. *//memanggil method void bernama tampillhello()*
25. *//dengan variabel lth yang baru dibuat*


29. }
31. }

*latihan2.java*

1. **package** latihan2; *// nama package latihan2*
3. ***/\*\****
4. ***\****
5. ***\* @author FATHUR (okedroid.com)***
6. ***\*/***
7. **public** **class** latihan2 { *//nama class latihan2*
9. **public** **void** tampilhello ()
10. *//method void dengan nama tampillhello()*
11. *//sehingga dapat di akses oleh package lain*
12. *//menggunakan modifier public agar dapat diakses oleh package lain*
13. {
14. System.out.println("Hello Java!!"); *//setiap statement selalu diakhiri ;*
15. *//obyek cetak text dengan tipe data string Hello Java*
16. }
18. }

**Output:**  
  
Hello Java !!  
  
  
  
  
**Dasar Syntax Java:**  
Dasar dari penulisan syntax dalam bahasa pemrograman Java ,yang perlu kalian ketahui salah satunya adalah :

* Bersifat **Case Sensitivity** : artinya penamaan dari kata "HelloWorld" dengan "helloworld" sangat berbeda.
* Nama Class :  untuk nama class huruf pertama harus  menggunakan huruf besar. (contoh :  class *NamaClass* )
* Nama Method : untuk nama method huruf pertama harus menggunakan huruf kecil. ( contoh : *public void namaMethodSaya()*  )
* Nama Program :  nama program java harus sama dengan nama class. Jadi jika kalian mempunyai nama class *NamaClass* , program file harus sama dengan nama class yaitu *NamaClass.java.*

# Belajar Mengenal Tipe Variable Bahasa Pemrograman Java

**Variable** di dalam bahasa pemrograman Java, ibarat sebuah wadah atau tempat  di dalam memory , yang digunakan untuk menyimpan sebuah nilai/value (seperti data atau informasi) , yang bersifat temporary (sementara). Jadi dapat berubah-ubah atau tidak tetap tergantung ,perintah si programmer tersebut.  Sama halnya seperti bahasa pemrograman lain , variabel di Java juga memiliki sebuah nama  , tipe data , dan nilai/value (seperti data atau informasi) ,yang akan menentukan sebuah ukuran variabel di dalam memori.

Macam-macam basic tipe data dalam Java salah satunya :

* String =  teks ( ex satu,dua,dll).
* int(integer) ,long ,short ,byte = bilangan bulat (0,1,2,3,dll).
* char  = karakter (ex A,B,C).
* float,double = bilangan desimal.
* boolean = nilai/value dari boolean hanya mengenal true dan false.

*Baca sebelumnya :*[***Belajar Cara Memahami Konsep Dasar Bahasa Program Java***](http://www.okedroid.com/2016/04/belajar-cara-memahami-konsep-dasar-bahasa-program-java.html)  
  
Ada 3 Macam Tipe Variabel di dalam Java yaitu:

* Local Variables
* Instance Variables (Non-static variables)
* Class Variables (Static Variables)

**1) Local Variables**

* Tipe data variable ini penggunaan atau pengaksesanya berada didalam method ,constructor atau statement block.
* Local Variables penggunaanya tidak memerlukan access modifier ( public , dll.) .
* Local Variables penggunaanya wajib di inisialisasi terlebih dahulu ,untuk mencegah terjadinya erorr, karna Local Variables tidak memiliki default value.

Contoh  code program :

2. **package** contoh2; *// nama package*
4. ***/\*\****
5. ***\****
6. ***\* @author FATHUR(Okedroid.com)***
7. ***\*/***

10. **public** **class** Contoh1 {   *//nama class Contoh1*
12. **public** **void** cetakNamaLengkap(){
13. *// menginisialisasi variable nama*
14. String nama = "Fathurrahman";

17. *//menginisialisasi variable kota*
19. String kota = "Bogor";
20. System.out.println("Nama :"+nama);
21. System.out.println("Kota :"+kota);
22. }

25. **public** **static** **void** main(String[] args) {  *//method void main utama*
26. *// TODO code application logic here*
28. *//membuat obyek baru cth dari class Contoh1*
29. Contoh1 cth = **new** Contoh1();
31. *//memanggil method void cetakNamaLengkap() dari obyek cth yang telah dibuat*
32. cth.cetakNamaLengkap();

35. }
36. }

Output dari program diatas :  
  
Nama : Fathurrahman  
  
Kota : Bogor

**2) Instance Variables (Non-static variables)**

* Tipe data variable ini penggunaan atau pengaksesanya berada di dalam kelas , namun diluar dari method , constructor atau statement block.
* Instance Variables memerlukan access modifier.
* Karna Instance Variables memiliki default value , seperti variable yang bertipe data boolean default valuenya adalah false, variable yang bertipe data number default valuenya adalah 0, serta yang bertipe object reference

Contoh code program:

1. **package** contoh2; *// nama package*
3. ***/\*\****
4. ***\****
5. ***\* @author FATHUR(Okedroid.com)***
6. ***\*/***



11. **public** **class** Contoh1 {   *//nama class Contoh1*
13. String namasiswa;
14. **char**  kelas;
15. **double**  nilaiujian;
17. *//inisialisasi di dalam kelas tapi diluar method*

20. **public** **void** cetakNamaSiswa(){
22. namasiswa = "Andi Wijayanto";
23. kelas = 'A';
24. nilaiujian = 7.89;
26. System.out.println("Nama Siswa :"+namasiswa);
27. System.out.println ("Kelas di :"+kelas);
28. System.out.println("Nilai Ujian : "+nilaiujian);



33. }

36. **public** **static** **void** main(String[] args) {  *//method void main utama*
37. *// TODO code application logic here*
39. Contoh1 cth = **new** Contoh1();
41. cth.cetakNamaSiswa();

44. }
45. }

Output dari program diatas :  
  
Nama Siswa :Andi Wijayanto  
Kelas di :A  
Nilai Ujian : 7.89

**3) Class Variables (Static Variables)**

* Tipe data variable ini menggunakan keyword static , yang berada didalam kelas ,namun diluar dari method , constructor atau statement block.
* Tipe data variable ini  biasanya digunakan untuk membuat sebuah variable bertipe konstanta, artinya nilai/value didalam variable bersifat tepat dan tidak dapat diubah.
* Tipe data variable ini memiliki default value yang sama seperti Instance Variables.

Contoh code program :

1. **package** contoh2; *// nama package*
3. ***/\*\****
4. ***\****
5. ***\* @author FATHUR(Okedroid.com)***
6. ***\*/***



11. **public** **class** Contoh1 {   *//nama class Contoh1*

14. **public** **static** **double** nilaiujian;
16. **public** **static** **final** String namasiswa = "Andi Wijayanto";

19. **public** **static** **void** main(String[] args) {  *//method void main utama*
20. *// TODO code application logic here*
22. nilaiujian = 7.89;
24. System.out.println("Nama Siswa : "+namasiswa);
25. System.out.println("NilaI Ujianya :"+nilaiujian);

28. }
29. }

Variable dapat di deklarasikan (declare) , di tentukan nilainya ( assign ), dan ditentukan kembali nilainya ( reassign) . Seperti yang saya sudah jelaskan diatas, variable merupakan tempat sementara untuk menyimpan nilai/value.

|  |
| --- |
| [Mendeclare,assign,reassign variable Java](https://1.bp.blogspot.com/-XkOYxsMvsWA/V0Kt0kjrSjI/AAAAAAAAFdE/t1JjfK6pggUjX_1VYd0DsIpzf_7GZ03FQCLcB/s1600/tipe-variable-java-declare-assign-reassign.png) |
| Mendeclare,assign,reassign variable Java |

Ketika kita mendeklarasikan tipe data**integer** dengan variable **x**, maka belum ada nilai/value yang tersimpan di variable **x**. Nah ketika kita menentukan nilai/value dari angka**5** ke variable**x**, maka kita dapat menentukan kembali variable **x**untuk menentukan nilai/value dari angka 7. Hasilnya ketika kita ingin melihat output lalu mengcompile atau menjalankan program dengan  perintah*System.out.println (x);*Output nya akan menampilkan angka : 7.

# Belajar Mengenal Array Bahasa Pemrograman Java

# Array merupakan sebuah deretan atau susunan , nama-nama variable( element)    , yang memiliki tipe data sama dalam daftar, yang dapat diakses secara berurutan(sequentially) . Untuk membedakan nama variable dan tipe data yang sama ,kita dapat mengaksesnya dengan menggunakan indeks yang dimulai dari angka [0] lalu [1,2,3,4..].dst. Array digunakan untuk menampilkan sebuah variable , dengan jumlah lebih dari 1 atau lebih banyak lagi. Jadi intinya jika kalian ingin menampilkan sebuah variabel dalam list ,kalian tidak perlu mendeklarasikan satu per satu,  karna di nilai tidak efisien untuk penulisan coding, kalian cukup menggunakan Array.

# Ketika kita mendefinisikan sebuah Array ,kita harus menspesifikasikan jumlah item dari sebuah tipe data. Contohnya seperti :

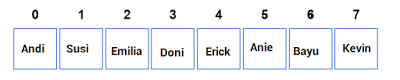
1. **int**[] nomor = **new** **int**[10];
2. *// membuat array dari 10 tipe data integer*
3. String[] nama = { "Andi", "Susi", "Emilia" };
4. *// membuat array dari 3 tipe data string*

Setiap kali kita mendefinisikan sebuah Array , element(atau data) dapat diakses dengan mengetahui posisi dari index yang dimulai dari angka 0. Dimulai dari sebelah kiri untuk baris elemen horizontal ,dan dari atas untuk kolom elemen vertikal.

Contohnya seperti (lanjutan program diatas):

1. System.out.println(numbers[0]); *// mencetak elemen/data pertama*
2. System.out.println(numbers[1]); *// mencetak elemen/data kedua*
4. String judul  = nama[0] + " dan " + nama [2];
6. *// Andi yang berada di posisi 0  dan Emilia yang berada di posisi 2*
7. *//posisi 1 dari susi digantikan dengan text dan*
9. System.out.println(judul);
10. *//akan menampilkan output = Andi dan Emilia*

Array merupakan daftar item , yang disimpan secara berurutan di dalam memory.

  
Array

Contoh code program :

2. **package** contoh2; *// nama package*
4. ***/\*\****
5. ***\****
6. ***\* @author FATHUR(Okedroid.com)***
7. ***\*/***







16. **public** **class** Contoh1 {   *//nama class Contoh1*



21. **public** **static** **void** main(String[] args) {  *//method void main utama*
22. *// TODO code application logic here*



27. String[] nama = **new** String[] {
28. "Andi", "Susi", "Emilia", "Doni", "Erick", "Anie", "Bayu", "Kevin" };
29. *//index array dimulai dari Andi = 0 1 2.. --->*
31. nama[3] = "Aldi";
32. *// menentukan atau mengubah kembali (reassign) nilai dari index array 3 Doni  menjadi Aldi*
33. System.out.println(nama[3]);
34. *//mencetak index array posisi 3*
35. *//output = Aldi*

38. }
39. }

Contoh code program iteration Array :

1. String[] arr2 = {"Hello","Saya","Java"};
2. *//deklarasi dan insialisai string array*
3. **int**[] arr = {1,3,5,6,7};
4. *//deklarasi dan insialisai int array*
6. **for** (**int** i=0;  i<arr2.length;  i++){
7. *//i dimulai dari index atau posisi 0*
8. *//kondisi berdasarkan < length = panjang atau jumlah array yaitu 3*
9. *//increment + 1*
10. System.out.println(arr2[i]);
11. *//menampilkan isi nilai/value array dari variabel j*
12. }
14. **for** (**int** j=0;  j<arr.length;  j++){
15. *//j dimulai dari index atau posisi 0*
16. *//kondisi berdasarkan < length = panjang atau jumlah array yaitu 5*
17. *//increment + 1*
19. System.out.println(arr[j]);
20. *//menampilkan isi nilai/value array dari variabel j*
22. }

**Output :**  
Hello  
Saya  
Java  
1  
3  
5  
6  
7

# Belajar tentang ArrayList Bahasa Pemrograman Java

**ArrayList**termasuk ke dalam *collection* di bahasa pemrograman Java ,yang digunakan untuk menampilkan daftar atau list dari nilai/value, yang dapat dimodifikasi dan juga bersifat dinamis. Sifat dinamis ini artinya , ukuran ArrayList dapat diubah dengan ditambahkan dan juga di hapus. Perlu kalian ketahui juga ,untuk membuat obyek baru dengan ArrayList.  ArrayList tidak mengenal tipe data primitive.

# Untuk lebih jelasnya kalian bisa lihat contoh syntax berikut:

# *List<String> stringdaftar = new ArrayList<String>();  List <Integer> angkadaftar = new ArrayList<Integer>(); //pendeklarasian yang benar ̶L̶i̶s̶t̶ ̶<̶i̶n̶t̶>̶ ̶l̶i̶s̶t̶k̶u̶ ̶=̶ ̶n̶e̶w̶ ̶A̶r̶r̶a̶y̶L̶i̶s̶t̶<̶i̶n̶t̶>̶(̶)̶;̶ //pendeklarasian yang salah*

Jadi untuk mendeklarasikan variable dari nilai/value angka , kita tidak menggunakan**int ,** akan tetapi kita menggunakan **Integer.** Dan juga **double** , **float** ,**long** ,**char** , dan **boolean** adalah tipe data primitif. Yang artinya sudah ada secara*built-in*di dalam bahasa pemrograman Java. Mereka bukan termasuk obyek atau class. Bagaimanapun juga , **Integer** , **Double** , **Float,** ,**Long** ,**Character** , dan **Boolean** adalah class bukan tipe data primitif, jadi kita harus menggunakanya di dalam ArrayList.

# Contoh code program ArrayList :

1. **import** java.util.ArrayList;
2. **import** java.util.List;
4. ***/\*\****
5. ***\****
6. ***\* @author FATHUR(Okedroid.com)***
7. ***\*/***
9. **public** **class** Cetak {   *//nama class Contoh1*


13. **public** **static** **void** main(String[] args) {  *//method void main utama*


17. List<String> phrase = **new** ArrayList<String>();
18. *//membuat obyek baru dengan nama phrase*
19. *//dengan memanggil class ArrayList dan tipe data String*
20. phrase.add("Hello");
21. phrase.add("Saya");
22. phrase.add("Java");
23. *//menambahkan nilai/value ke var phrase*
24. List<String> phrase2 = **new** ArrayList<String>();
25. *//membuat obyek baru dengan nama phrase2*
26. *//dengan memanggil class ArrayList dan tipe data String*
27. phrase.add("Hello");
28. phrase.add("Saya");
29. phrase.add("Java");
31. List<Integer> nomor = **new** ArrayList<Integer>();
33. *//membuat obyek baru dengan nama var nomor*
34. *//dengan memanggil class ArrayList dan tipe data Integer*
35. nomor.add(4);
36. nomor.add(6);
37. nomor.add(9);
38. *//menambahkan nilai/ value ke var nomor*
40. System.out.println(nomor);
41. *//menampilkan isi dari nilai/value variable nomor*
42. System.out.println(phrase.get(0));
43. System.out.println(phrase.get(1));
44. System.out.println(phrase.get(2));
45. *//mengakses Arraylist di variable phrase*
46. *//sesuai dengan posisi atau index array*

49. phrase.addAll(phrase2);
51. *//menambahkan semua nilai/value di variable phrase2*
52. *//ke dalam variable phrase*
53. System.out.println(phrase);
54. *//menampilkan variable phrase setelah ditambahkan*
56. phrase.remove(2);
57. *//menghapus salah satu nilai/value di variable phrase*
58. *//berdasarkan index atau posisi*
59. System.out.println(phrase);
60. *//menampilkan variable phrase setelah dihapus*
61. phrase.clear();
62. *//menghapus semua  nilai/value di variable phrase*
63. System.out.println(phrase);
64. *//menampilkan variable phrase setelah dihapus*
66. }
67. }

# Output: [4, 6, 9] Hello Saya Java [Hello, Saya, Java, Hello, Saya, Java] [Hello, Saya, Hello, Saya, Java] []

# Belajar tentang Method di Bahasa Pemrograman Java

**Method**(dikenal juga sebagai Function atau Prosedur) adalah suatu perintah kode , yang diberi nama untuk dijalankan di dalam sebuah program Java. Method merupakan sebuah cara untuk mempermudah programmer , dalam mengelompokan sebuah baris code atau pernyataan(statement). Jadi dengan Method , kita bisa menjalankan beberapa baris code atau pernyataan sekaligus , hanya dengan memanggil (calling) nama method tersebut. Hal ini  agar program kita terlihat terstruktur dan efisien.

Contoh code program dari penerapan method di java:

1. **package** contoh2; *// nama package*
3. ***/\*\****
4. ***\****
5. ***\* @author FATHUR(Okedroid.com)***
6. ***\*/***







15. **public** **class** Contoh1 {   *//nama class Contoh1*



20. **public** **void** cetakInfo() { *//method cetakInfo()*
21. System.out.println("Nama saya Okedroid "); *//perintah untuk menampilkan text*
22. System.out.println("Saya adalah seorang Blogger");*//perintah untuk menampilkan text*
23. }
25. *//mendefinisikan 2 baris code atau pernyataan untuk menampilkan text*
26. *//didalam method cetakInfo()*
28. **public** **static** **void** main(String[] args) {  *//method void main utama*

31. Contoh1 ctk = **new** Contoh1();
32. *//konstruktor untuk membuat obyek baru ctk*
34. ctk.cetakInfo(); *//memanggil method yang bernama cetakInfo() di dalam class Contoh1*



39. }
40. }

**Output :**  
Nama saya Okedroid  
Saya adalah seorang Blogger

Seperti yang kalian lihat , dari Contoh syntax program diatas. Dengan method kita hanya memanggil nama method dari *cetakInfo() .*Untuk menampilkan dua baris perintah atau statement , yang berfungsi  menampilkan text di layar output.

Penting untuk diketahui , kita harus memahami perbedaan antara mendefinisikan (Defining) method dengan  memanggil (Calling) method.

* Mendefinisikan (Defining) method artinya , kita memberi nama dari grup code , tapi tidak menjalankan code program , yang berada di dalam grup code tersebut. Menefinisikannya di dalam class.

*public void cetakInfo() {  
System.out.println("Nama saya Okedroid ");  
System.out.println("Saya adalah seorang Blogger");  
}*

* Memanggil (Calling) method artinya memanggil grup code,  dan menjalankan code program  di dalamnya , berdasarkan nama yang telah di definisikan. Memanggilnya menggunakan object.

*ctk.cetakInfo();*

Method dapat menyimpan suatu nilai/values yang disebut **parameters**, yang mana dapat diakses ketika dijalankan.

Contoh code program:

1. **package** contoh2; *// nama package*
3. ***/\*\****
4. ***\****
5. ***\* @author FATHUR(Okedroid.com)***
6. ***\*/***







15. **public** **class** Contoh1 {   *//nama class Contoh1*



20. **public** **void** cetakInfo(String nama) {
21. *//membuat paramater yang memempunyai nilai/values text bertipe data string*
22. System.out.println("Hi "+nama);
23. *//menambahkan nilai/value yaitu nama yang berada di dalam parameter*
25. }
27. **public** **static** **void** main(String[] args) {  *//method void main utama*

30. Contoh1 ctk = **new** Contoh1();
32. ctk.cetakInfo("Okedroid");
34. *//memanggil method yang berisi parameter*
36. }
37. }

Output:  
  
Hi Okedroid

Terdapat 2 jenis method yang perlu kalian ketahui, Method void dan Method non void. Method yang menggunakan keyword void , yang contoh syntaxnya seperti yang ada di atas, adalah method yang di buat agar tidak dapat mengembalikan nilai. Sedangkan method non void , method yang tidak menggunakan keyword void ,merupakan method yang dapat mengembalikan nilai.

Serta method non void wajib menggunakan keyword **return**, dan tipe data nya harus sama dengan nilai yang akan dikembalikanya (return). Keyword return value dapat disimpan ke dalam variable , lalu digunakan di dalam program.

Contoh code program method non void :

2. **package** contoh2; *// nama package*
4. ***/\*\****
5. ***\****
6. ***\* @author FATHUR(Okedroid.com)***
7. ***\*/***
8. Co






16. **public** **class** Contoh1 {   *//nama class Contoh1*





23. **public** **static** **void** main(String[] args) {  *//method void main utama*


27. System.out.println(sum); *// output 11*
28. }
30. **public** **static** **int** add(**int** n1, **int** n2) {
31. *//method non void yang tidak menggunakan keyword void*
32. *//langsung dengan mendefiniskan tipe data*
33. **return** n1 + n2;
34. }

37. **static** **int** sum = add(5, 6); *// variabel sum diguanakan untuk mengembalikan nilai*
38. }

# Belajar tentang Looping (Perulangan) di Bahasa Pemrograman Java

**Looping**  adalah suatu perintah ,yang digunakan dalam bahasa pemrograman Java, untuk melakukan tugas secara berulang kali, dengan nilai/value yang sudah ditentukan .Looping juga dapat membatasi baris instruksi (codingan) sampai syarat/kondisi tertentu. Penggunaan looping , di dalam bahasa pemrograman bertujuan untuk mengurangi baris code yang telah ditulis. Looping juga biasanya digunakan bersamaan dengan **Method**. Looping membantu para programer , untuk menjalankan sekelompok pernyataan baris instruksi (codingan)  , secara berulang-ulang sampai kondisi tertentu terpenuhi.

Di dalam bahasa pemrogaman Java , terdapat 3 Keyword yang digunakan untuk melakukan tugas Looping. yaitu :

* For
* While
* Do While

**1. For**

**For**merupakan jenis looping yang jumlah dari perulanganya sudah di tentukan di awal.

Contoh code program for :

//PROGRAM 4-13

class DemoFor1 {

public static void main(String[] args) {

for (int i=0; i<10; i++) {

System.out.println("Java");

}

}

}

Contoh For 2 :

//PROGRAM 4-15

class DemoFor2 {

public static void main(String[] args) {

int n = 5; // 5 bilangan positif pertama

int hasil = 0;

for (int i=1; i<=n; i++) {

hasil += i;

if (i != n) {

System.out.print(i + " + ");

} else {

System.out.print("= ");

}

}

System.out.println(hasil);

}

}

Contoh For 3 :

//PROGRAM 4-16

class DemoFor3 {

public static void main(String[] args) {

int bilangan = 7;

boolean prima = true;

for (int i=2; i <= (bilangan/2); i++) {

if ((bilangan % i) == 0) {

prima = false;

break; // menghentikan pengulangan

}

}

if (prima) {

System.out.println(bilangan + " merupakan bilangan prima");

} else {

System.out.println(bilangan + " bukan bilangan prima");

}

}

}

Demo For 4 :

//PROGRAM 4-17

class DemoFor4 {

public static void main(String[] args) {

int j=4;

for (int i=0; i < 5; i++) {

System.out.println("Nilai i: " + i);

System.out.println("Nilai j: " + j);

System.out.println();

j--;

}

}

}

Demo ForBersarang :

//PROGRAM 4-29

class DemoForBersarang {

public static void main(String[] args) {

for (int i=1; i<=5; i++) {

for (int j=1; j<=i; j++) {

System.out.print(i\*j + " ");

}

System.out.println();

}

}

}

**2. While**  
 **While** adalah jenis looping yang lebih sederhana dibandingkan dengan for. While akan menjalankan pengulangan sampai kondisi tidak lagi benar(false) .

Contoh code program while :

//PROGRAM 4-23

class DemoWhile2 {

public static void main(String[] args) {

int n=5; // 5 bilangan positif pertama

int hasil=0;

int i=1;

while (i<=n) {

System.out.print(i);

if (i != n) {

System.out.print(" + ");

} else {

System.out.print(" = ");

}

hasil += i;

i++;

}

System.out.println(hasil);

}

}

Contoh program while 2 :

//PROGRAM 4-24

class DemoWhile3 {

public static void main(String[] args) {

int bilangan=5; // bilangan yang akan dihitung

// nilai faktorialnya

int hasil=1;

int i=bilangan;

System.out.print(bilangan + "! = ");

while (i>=1) {

System.out.print(i);

if (i != 1) {

System.out.print(" x ");

} else {

System.out.print(" = ");

}

hasil \*= i; // dapat ditulis: hasil = hasil \* i;

i--;

}

System.out.println(hasil);

}

}

**3. Do While**

Do While adalah jenis looping yang dimana statement atau pernyataan,  di lakukan di awal syntax (Do). Lalu syarat atau kondisi di lakukan di akhir syntax (While). Meskipun syarat atau kondisi  dari while bernilai salah (false), program tetap akan menampilkan output.

Contoh code program while :

//PROGRAM 4-27

class DemoDoWhile1 {

public static void main(String[] args) {

int n=5; // 5 buah bilangan positif pertama

int hasil=0;

int i=1;

do {

System.out.print(i);

if (i != n) {

System.out.print(" + ");

} else {

System.out.print(" = ");

}

hasil += i;

i++;

} while (i <= n);

System.out.println(hasil);

}

}

# Belajar IF ELSE Bahasa Pemrograman Java

# IF ELSE merupakan suatu perintah conditional atau percabangan,  yang digunakan dalam bahasa pemrograman Java, untuk memeriksa  suatu kondisi,lalu akan menjalankan program sesuai dengan syarat/kondisi yang akan ditampilkan. Jika (if) kondisi bernilai benar (true) , maka program akan menjalankan pernyataan (statement) A. Namun jika tidak (else if) ,maka program akan menjalankan pernyataan (statement ) B. Atau jika kedua kondisi (if dan ,else if ) tidak ada yang benar , maka program akan menjalankan pernyataan (statement) C.

Contoh Syntax :

# *if ( syarat kondisi ) { //menjalankan statement A }else if ( syarat kondisi ) { //menjalankan statement B }else { //menjalankan statement c }*

# Jadi penggunaan  IF ELSE di Bahasa Pemrograman Java , sangat penting jika kita ingin membuat program, dengan situasi atau kondisi berbeda-beda. Agar mencegah terjadinya erorr atau bug di program kita.

# Contoh Kode Program :

# //PROGRAM 4-1

# class DemoIfSatuKondisi {

# public static void main(String[] args) {

# int a=1, b=10;

# if (a < 5) {

# System.out.println("Nilai a lebih kecil dari 5");

# }

# if (b < 5) {

# System.out.println("Nilai b lebih kecil dari 5");

# } } }

# Belajar Switch Case Bahasa Pemrograman Java

# Switch Case merupakan suatu perintah conditional atau percabangan, yang fungsinya kurang lebih sama seperti IF ELSE , hanya berbeda syntaxnya saja. Jika penggunaan Switch Case di awali dengan keyword Switch lalu di bagian kondisi berisi variable ,kemudian percabangan case di isi dengan beberapa pernyataan/statement , lalu dengan keyword default ketika semua kondisi pernyataan tidak ada yang benar(sama seperti else ). Penggunaan Switch Case ,biasanya digunakan untuk menentukan alternatif dari sebuah pilihan.

# Contoh Syntax :

# *switch (variable ) { case 1: //statement 1 break; case 2: //statement 2 break; default: //akan dijalankan ketika semua kondisi statement salah break; }*

# Nah disini kita akan membuat contoh code program dari Switch Case , yang berisi schedule .

# import java.util.Scanner;

# public class contoh\_switchcase{

# public static void main(String [] args){

# Scanner input = new Scanner (System.in);

# System.out.println(">>>>>>>>>>>>>#################################<<<<<<<<<<<<<<");

# System.out.println("\t \t \"Welcome To My Schedule\"");

# System.out.println("\t \t \"UMAR AJI PRATAMA\"");

# System.out.println("\t \t \"IMKOM UNIVERSITY\"");

# System.out.println(">>>>>>>>>>>>>#################################<<<<<<<<<<<<<<");

# System.out.println("");

# System.out.println("Jadwal Kuliah");

# System.out.println(">>>==Hari==<<<");

# System.out.println("1. Senin");

# System.out.println("2. Selasa");

# System.out.println("3. Rabu");

# System.out.println("4. Kamis");

# System.out.println("5. Jum'at");

# System.out.println("6. Sabtu");

# System.out.println("7. Minggu");

# System.out.println("Silahkan Pilih Hari = ");

# int x = input.nextInt();

# switch (x)

# {

# case 1 : System.out.println(" >>> Anda Memilih Hari Senin <<< \n \"Hari Senin Anda Tidak Memiliki Jam Kuliah\" \"I Feel FREE!!! :D \" ") ;

# break;

# case 2 : System.out.println(">>> Anda Memilih Hari Selasa <<< \n \"Hari Selasa Anda Kuliah >>> Pemograman Berbasis Objek > (08.20 WIB)\" ");

# break;

# case 3 : System.out.println(" >>> Anda Memilih Hari Rabu <<< \n \"Hari Rabu Anda Kuliah dan Mengajar di IMKOM >>> Pemrograman Web > (08.30 WIB)\" \n \t \t \t >>> Mengajar Android Programming > (16.00 WIB)\"");

# break;

# case 4 : System.out.println(">>> Anda Memilih Hari Kamis <<< \n \"Hari Kamis Anda Kuliah >>> Pemrograman Lanjut > (09.10 WIB)\" \n \t \t \t >>> Pemrograman Berorientasi Objek > (13.10 WIB)\" ");

# break;

# case 5 : System.out.println(">>> Anda Memilih Hari Jum'at <<< \n \"Hari Jum'at Anda Kuliah >>> Program Aplikasi Mobile > (09.10 WIB)\" ");

# break;

# case 6 : System.out.println(">>> Anda Memilih Hari Sabtu <<< \n \"Hari Sabtu Anda Tidak Memiliki Jam Kuliah\" \"I Feel FREE!!! :D \"");

# break;

# case 7 : System.out.println(">>> Anda Memilih Hari Minggu <<< \n \"Hari Minggu Anda Tidak Memiliki Jam Kuliah\" \"I Feel FREE!!! :D \"");

# break;

# default : System.out.println("Pilihan Anda Salah!!! \"Ingat Dalam Satu Minggu Hanya terdapat 7 Hari!!!\"");

# break;

# }

# System.out.println("");

# System.out.println(">>>>>>>>>>>>>#################################<<<<<<<<<<<<<<");

# System.out.println(">>>>>>>>>>>>>#################################<<<<<<<<<<<<<<");

# System.out.println("");

# System.out.println("Silahkan Lihat Kegiatan Anda Yang Lainnya");

# System.out.println("");

# System.out.println(">>>==Kegiatan==<<<");

# System.out.println("1. Senin");

# System.out.println("2. Selasa");

# System.out.println("3. Rabu");

# System.out.println("4. Kamis");

# System.out.println("5. Jum'at");

# System.out.println("6. Sabtu");

# System.out.println("7. Minggu");

# System.out.println("Silahkan Pilih Hari = ");

# int y = input.nextInt();

# switch (y)

# {

# case 1 : System.out.println(" >>> Anda Memilih Hari Senin <<< \n \"Kegiatan Hari Senin Anda >>> Asisten Dosen Pemrograman Mobile > (10.00 WIB)\" ") ;

# break;

# case 2 : System.out.println(">>> Anda Memilih Hari Selasa <<< \n \"Kegiatan Hari Selasa Anda >>> Latihan Musik \'Rockstar Studio\' > (16.00 WIB)\" ");

# break;

# case 3 : System.out.println(" >>> Anda Memilih Hari Rabu <<< \n \"Kegiatan Hari Rabu Anda >>> Membuat Website > (13.00 WIB)\" ");

# break;

# case 4 : System.out.println(">>> Anda Memilih Hari Kamis <<< \n \"Kegiatan Hari Kamis Anda >>> Membuat Aplikasi Android > (15.30 WIB)\" ");

# break;

# case 5 : System.out.println(">>> Anda Memilih Hari Jum'at <<< \n \"Kegiatan Hari Jum'at Anda >>> Latihan Seni Beladiri > (16.00 WIB)\"");

# break;

# case 6 : System.out.println(">>> Anda Memilih Hari Sabtu <<< \n \"Kegiatan Hari Sabtu Anda >>> Membuat Rancangan Kegiatan Organisasi > (13.00 WIB)\" ");

# break;

# case 7 : System.out.println(">>> Anda Memilih Hari Minggu <<< \n \"Kegiatan Hari Minggu Anda >>> Melanjutkan Tugas yang Belum Terselesaikan ");

# break;

# default : System.out.println("Pilihan Anda Salah!!! \"Ingat Dalam Satu Minggu Hanya terdapat 7 Hari!!!\"");

# break;

# }

# System.out.println("");

# System.out.println(">>>>>>>>>>>>>#################################<<<<<<<<<<<<<<");

# System.out.println(">>>>>>>>>>>>>#################################<<<<<<<<<<<<<<");

# System.out.println("");

# System.out.println(" \"Waktu ibarat pedang, jika engkau tidak menebasnya maka ialah yang akan menebasmu\" \n \"Maka Manfaatkanlah Waktumu Sebaik - Baiknya, dan Jangan Lupa Beribadahlah\" ");

# System.out.println("");

# System.out.println(">>>>>>>>>>>>>#################################<<<<<<<<<<<<<<");

# System.out.println(">>>>>>>>>>>>>#################################<<<<<<<<<<<<<<");

# }

# }

Scanner merupakan perintah untuk membuat objek atau menginisialisasi object yang diinginkan pengguna atau yang diinputkan pengguna

Perintah-Perintah yang digunakan :

1. import java.util.Scanner;  
   Ini untuk memasukan fungsi Scanner pada program.
2. Scanner data = new Scanner (System.in);

Ini membangun/membuat sebuah Scanner baru yang menghasilkan nilai-nilai yang dipindai dari file yang ditentukan. Data adalah nama scanner yang akan di buat , kita bisa merubah sesuai dengan data yang akan kita buat , yang terpenting adalah tidak membuat eror atau anda kebingungan.

1. A = data.nextInt();

Berfungsi untuk penginputan data yang akan di buat dari kita, A adalah nama variabel . Syntax tersebut berfungsi pada tipe data integer , contohnya pada tipe data double maka gunakan A = input.nextDouble(); .

**ARRAY DALAM JAVA**

Array adalah sekumpulan variabel yang memiliki tipe data yang sama dan dinyatakan dengan nama yang sama. Array merupakan konsep yang penting dalam pemrograman, karna array memungkinkan untuk menyimpan data maupun referensi objek dalam jumlah banyak dan terindeks. Array menggunakan indeks integer untuk menentukan urutan elemen-elemennya, dimana elemen pertamanya dimulai dari indeks 0,elemen kedua memiliki indeks 1, dan seterusnya.

·         Contoh :

-          Angka untuk menyimpan sederetan bilangan

-          Buku untuk menyimpan sekumpulan data buku

-          Mahasiswa untuk menyimpan beberapa data mahasiswa

·         Sebagai contoh jika A merupakan sebuah array dengan tipe integer, maka notasi dari array A adalah: A[n], dengan n merupakan angka index dari array tersebut misal:

A[0]=100  
A[1]=200  
A[2]=300  
A[3]=400

**A.**   **Mendeklarasikan Variabel Array**

·         Mendeklarasikan variabel array dengan tipe data yang diinginkan dengan cara yang hampir sama dengan variabel biasa. Misalnya untuk mendeklarasikan variabel bertipe integer, dapat dilakukan dengan cara :

**int [ ] bilangan;**atau **int bilangan [ ];**

Jadi perbedaan utama pendeklarasian variabel array dengan variabel biasa adalah adanya tanda kurung [ ] di akhir tipe data atau di akhir nama variabel array. Pada tahap pendeklarasian variabel array ini belum ada alokasi memory untuk menyimpan data.

B.   **Mendefenisikan Array**

·         Setelah mendeklarasikan array, kita perlu mendefenisikan array, dalam arti menentukan besar array yang diinginkan. Misalnya dengan cara :

**Bilangan = new int [5];**

·         Array memiliki ukuran yang tetap dalam arti tidak dapat membesar atau mengecil ukurannya setelah didefenisikan. Setelah didefenisikan, maka variabel dengan nama bilangan dapat menyimpan 5 nilai integer yang dapat diakses melalui indeks 0 sampai indeks 4. Setelah pendefenisian array, maka memori akan dialokasikan untuk menyimpan data dari array. Besar memori yang dialokasikan tergantung dari tipe data variabel array dan jumlah elemen array yang didefenisikan.

·  **C.   Array Dua Dimensi**

      Pada java juga menyediakan fasilitas untuk membuat array dua dimensi yang dapat membantu dalam pemrograman apabila array datu dimensi tidak mencukupi dalam menghasilkan suatu solusi. Array dua dimensi sebenarnya adalah array yang berisi array.

**D.   Array Multidimensi**

Selain array satu dimensi dan array dua dimensi, dapat juga membuat array multi dimensi pada java. Array multidimensi merupakan array yang terdiri dari array yang tidak terbatas hanya dua dimensi saja. Kita dapat menggunakan kode berikut untuk mendapatkan array 3 dimensi :

**Int [ ] [ ] array dimensi = new int [ 5 ] [ 10 ] [ 5 ] ;**

  Dan pada array multidimensi , kita dapat menetukan ukuran array yang berbeda pada tiap array. Misalnya :

**Int [ ] [ ] [ ] mdimensi = new int [ 5 ] [ ] [ ] ;**

   Dari kode diatas, kita mendapatkan array pertama dengan 5 elemen, tetapi kita belum mendefinisikan ukuran array dimensi kedua dan ketiga.

Contoh ;

// Elemen 512 x 128 dari integer array

int[][] twoD = new int[512][128];

// karakter array 8 x 16 x 24

char[][][] threeD = new char[8][16][24];

// String array 4 baris x 2 kolom

String[][] dogs = {{ "terry", "brown" },

{ "Kristin", "white" },

{ "toby", "gray"},

{ "fido", "black"}

};

            Untuk mengakses sebuah elemen didalam array multidimensi, sama saja dengan mengakses array satu dimensi. Misalnya saja, untuk mengakses element pertama dari baris pertama didalam array dogs, kita akan menulis,

System.out.print( dogs[0][0] );

Kode diatas akan mencetak String “terry” di layar.

Contoh program array :

Mendeklarasikan variabel array tanpa menyebutkan berapa jumlah elemen yang diperlukan.

**public class Array**

**{**

**public static void main(String[] args)**

**{**

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

**x[0]= 1;**

**x[1]= 2;**

**x[2]= 3;**

**System.out.println(x[0]);**

**System.out.println(x[1]);**

**System.out.println(x[2]);**

**}**

**}**

**Contoh Program array 2 Mendeklarasikan variabel array dengan menyebutkan jumlah elemen yang diperlukan.**

**public class Array1**

**{**

**public static void main(String[] args)**

**{**

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

**x[0]= 1;**

**x[1]= 2;**

**x[2]= 3;**

**System.out.println(x[0]);**

**System.out.println(x[1]);**

**System.out.println(x[2]);**

**}**

**}**

**Contoh Program array**

**//program array dengan menggunakan ‘length’**

**//program array dengan menggunakan ‘length’**

**public class ArrayBunga2**

**{**

**public static void main(String[] args)**

**{**

**String[] bunga = {"Mawar", "Melati", "Anggrek"};**

**for(int i=0;i<bunga.length; i++)**

**System.out.println(bunga[i]);**

**}**

**}**

# Contoh Program Array :

# import java.util.Random;

# class StatistikBlnLahir {

# public static void main (String [] args) {

# String[] bulan = {"Jan","Feb","Mar","Apr","Mei","Jun","Jul","Ags","Sep",

# "Okt","Nov","Des"};

# int jmlLahir[] = new int[12];

# Random bln = new Random();

# for (int i=0; i<12; i++)

# jmlLahir[i]=0;

# // deklarasi jmlLahir bisa juga seperti di bawah (tidak perlu for untuk inisialisasi)

# // int jmlLahir[] = {0,0,0,0,0,0,0,0,0,0,0,0};

# int n=0;

# do {

# int blnLahir=bln.nextInt(12);

# jmlLahir[blnLahir]=jmlLahir[blnLahir]+1;

# n++;

# } while (n<1000);

# for (int i=0; i<12; i++)

# System.out.println("Lahir bulan "+bulan[i]+" "+jmlLahir[i]+" orang");

# int maks=0;

# for (int i=0; i<12; i++)

# if (jmlLahir[i]>maks)

# { maks=jmlLahir[i]; n=i; }

# System.out.println("Terbanyak lahir bulan "+bulan[n]+" "+jmlLahir[n]+" orang");

# }

# }

# Penjelasan :

## java.util.Random (I)

kelas ini berfungsi sebagai random number generatordi Java. Kelas Random bisa diakses lewat packaage java.util atau melalui static class, Math. Objek dari kelas Random ternyata tidak hanya bisa menghasilkan value berupa angka, namun class Random bisa menghasilkan random boolean, byte, float, double, int, longdanGaussian, dimana method random pada Math hanya bisa memproduksi value double dalam range 0.0 ≤x < 1.0, dimana x adalah value yang dihasilkan oleh methodRandom tersebut.

**Array 2 Dimensi**

Array multidimensi tidaklah jauh berbeda dengan array 1 dimensi, hanya saja pada array multi dimensi akan terdapat semacam kolom dan baris, anggap saja seperti bagan matriks pada al-jabar matematika.

Contoh program :

**public class Array2D {**

**public static void main(String[] args) {**

**// membuat isi elemen array**

**String [][] country ={**

**// keterangan {kolom1,kolom2,kolom3}**

**{"No","Name","Country"}, // baris ke-0**

**{"1.","David","Australia"}, // baris ke-1**

**{"2.","Ahmad","Indonesia"}, // baris ke-2**

**{"3.","Sudaish","Mesir"}, // baris ke-3**

**{"4.","Kim","Korea"} //baris ke-4**

**};**

**// mendeklarasikan baris dan kolom**

**int i, j; // i = baris, j = kolom**

**for ( i=0;i<5; i++){ // menampilkan elemen sejumlah baris**

**for ( j=0;j<3; j++){ // menampilkan elemen sejumlah kolom**

**// menampilkan isi elemen baris dan kolom**

**System.out.print(country[i][j]+" ");**

**}**

**System.out.println(""); //pindah baris**

**}**

**}**

**}**

**Penjelasan :**  
1. Deklarasikan nama dan tipe data String serta model array 2 dimensi dengan tanda ( [] [] ).  
2. Tuliskan isi elemen array seperti contoh koding diatas.  
3. Membuat variabel dan tipe data*int i = baris* dan *int j = kolom*.  
4. Gunakan perulangan (for) untuk proses menampilkan isi elemen.  
5. Untuk mencetak gunakan nama\_variabel\_arrya [i][j].  
6. Jangan lupa untuk memberi sipasi System.out.println("") agar terlihat seperti gambar diatas.  
7. Perhatikan tutup kurawa dari setiap masing-masing blok baik dari kelas, main, array dan for.

Array 3 Dimensi

***public class Array3D {  
    public static void main(String[] args) {  
    // membuat isi elemen array  
        int [][][] angka ={     // {kolom1,kolom2,kolom3}  
                                {{1,2,3},    // baris ke-0  
                                 {4,5,6},    // baris ke-1  
                                 {7,8,9}},   // baris ke-2  
                                {{10,11,12},    // baris ke-3  
                                 {13,14,15},    // baris ke-4  
                                 {16,17,18}},   // baris ke-5  
                                {{19,20,21},    // baris ke-6  
                                 {22,23,24},    // baris ke-7  
                                 {25,26,27}}    // baris ke-8  
                           };  // kurung kurawa array penutup    
       // mendeklarasikan baris dan kolom  
        int i, j,k;  // i = baris, j = kolom  
        for ( i=0;i<3; i++){        // menampilkan elemen sejumlah baris  
            for ( j=0;j<3; j++){    // menampilkan elemen sejumlah kolom  
                for ( k=0;k<3; k++){     
                    // menampilkan isi elemen baris dan kolom  
                    System.out.print(angka[i][j][k]+" ");    
                }  
                    System.out.println("");         //pindah baris  
            }  
                    System.out.println("");         //pindah baris  
        }  
    }  
}***

**PENJELASAN**:

1. Deklarasikan nama dan tipe data serta model array 3 dimensi dengan tanda ( [] [] [] ).
2. Tuliskan isi elemen array seperti contoh koding diatas.
3. Membuat variabel dan tipe data*int i = baris* dan *int j = kolom* dan *int k.*
4. Gunakan perulangan (for) untuk proses menampilkan isi elemen (lihat koding diatas).
5. Untuk mencetak gunakan nama\_variabel\_array [i][j][k].
6. Jangan lupa untuk memberi sipasi System.out.println("") agar terlihat seperti gambar diatas.
7. Perhatikan tutup kurawa dari setiap masing-masing blok baik dari kelas, main, array dan for.

# Belajar tentang Method di Bahasa Pemrograman Java

**Method**(dikenal juga sebagai Function atau Prosedur) adalah suatu perintah kode , yang diberi nama untuk dijalankan di dalam sebuah program Java. Method merupakan sebuah cara untuk mempermudah programmer , dalam mengelompokan sebuah baris code atau pernyataan(statement). Jadi dengan Method , kita bisa menjalankan beberapa baris code atau pernyataan sekaligus , hanya dengan memanggil (calling) nama method tersebut. Hal ini  agar program kita terlihat terstruktur dan efisien.