Belajar Cara Memahami Konsep Dasar Bahasa Program Java

Oke artikel tentang belajar kali ini saya buat ,khusus untuk kalian yang ingin membuat dan mengembangkan Aplikasi Android. Namun masih terkendala dalam logika pemrograman Java. Karena Android dikembangkan dengan menggunakan bahasa pemrograman Java, intinya kita harus mengenali dan memahami dulu ,syntax(tata bahasa) dalam pemrograman Java. Belajar *syntax* bahasa java sama halnya ,saat kita ingin belajar *grammar*di bahasa inggris untuk berbicara. Seperti saat kita ingin ke luar negeri  ,minimal kita harus punya kemampuan bahasa inggris yang cukup ,untuk memahami budaya dan orang-orang sekitar

**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 atau bentuk awal (prototype), 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.  Atribut bisa juga disebut state dari objek tersebut. Misal Manusia Mempunyai atribut atau state : Nama ,Tinggi,Berat,Umur dan Jenis Kelamin.
* **Method** merupakan sebuah perilaku (behavior) dari sebuah class. Method bisa disebut juga perilaku dari objek tersebut : Misal Manusia mempunyai method : Berjalan ,Berlari ,Loncat,Berbicara ,dan Tidur.
* 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)

*Jadi maksud dari Java berorientasi pada objek itu , semua status dan method mengacu pada obyek ,yang dimana itu adalah wujud real atau nyata dari sebuah Class ,yang ada pada bahasa pemrograman Java.*

|  |
| --- |
|  |
| Atribute dan Behavior |

* **[Encapsulation](http://www.okedroid.com/2016/06/belajar-encapsulation-di-bahasa-pemrograman-java.html%22%20%5Ct%20%22_blank)**(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%22%20%5Ct%20%22_blank)**(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%22%20%5Ct%20%22_blank)** (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 |

* **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*

DATA HOSTED WITH ♥ BY [PASTEBIN.COM](http://pastebin.com/%22%20%5Ct%20%22_blank) - [DOWNLOAD RAW](http://pastebin.com/raw/xx8uCzEL%22%20%5Ct%20%22_blank) - [SEE ORIGINAL](http://pastebin.com/xx8uCzEL%22%20%5Ct%20%22_blank)

1.
2. **package** latihan1;  *// nama package latihan1*
3.
4.
5. **import** latihan2.latihan2;
6. *//import class bernama latihan dua dari package lain bernama latihan2*
7.
8.
9. ***/\*\****
10. ***\* @author FATHUR (okedroid.com)***
11. ***\*/***
12. **public** **class** latihan1 { *//nama class latihan1 dengan modifier public*
13.
14. ***/\*\****
15. ***\* @param args the command line arguments***
16. ***\*/***
17. **public** **static** **void** main(String[] args) {
18.
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*
22.
23. lth.tampilhello();*//setiap statement selalu diakhiri ;*
24. *//memanggil method void bernama tampillhello()*
25. *//dengan variabel lth yang baru dibuat*
26.
27.
28.
29. }
30.
31. }

*latihan2.java*

DATA HOSTED WITH ♥ BY [PASTEBIN.COM](http://pastebin.com/%22%20%5Ct%20%22_blank) - [DOWNLOAD RAW](http://pastebin.com/raw/VU15RHv0%22%20%5Ct%20%22_blank) - [SEE ORIGINAL](http://pastebin.com/VU15RHv0%22%20%5Ct%20%22_blank)

1.
2. **package** latihan2; *// nama package latihan2*
3.
4. ***/\*\****
5. ***\****
6. ***\* @author FATHUR (okedroid.com)***
7. ***\*/***
8. **public** **class** latihan2 { *//nama class latihan2*
9.
10. **public** **void** tampilhello ()
11. *//method void dengan nama tampillhello()*
12. *//sehingga dapat di akses oleh package lain*
13. *//menggunakan modifier public agar dapat diakses oleh package lain*
14. {
15. System.out.println("Hello Java!!"); *//setiap statement selalu diakhiri ;*
16. *//obyek cetak text dengan tipe data string Hello Java*
17. }
18.
19. }

**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.*