**Sintaks dan java**

Case sensitive

* Variabel x berbeda dengan variable X.
* Nama class HelloWord berbeda dengan nama kelas helloWord.
* Nama method hitung berbeda dengan Hitung().

Harus diperhatikan juga hal-hal berikut ini:

* System.out.print berbeda dengan ini system.out.print.

“S” yang benar harus menggunakan huruf “S” kapital.

**Nama kelas**

* Untuk nama kelas sebaiknya di awali dengan huruf kapital.

Contoh: class Hello atau class HelloWorld.

* Nama kelas harus sama dengan nama file,jika tidak terjadi error.

Contoh: jika nama class HelloWorld,maka harus HelloWorld.java.

* Jika nama class tidak sama dengan nama file,maka program tidak akan di compile.

**Nama method**

* nama method sebaiknya di awali dengan huruf kecil.

Contoh:hitung () atau hitungAngka().

* Method hitung () berbeda dengan method Hitung().

**Main method**

* Sebuah method yang akan dipanggil setiap program java dijankan.
* Main method wajib ada di setiap program java.
* Beberapa contoh versi main method.

Contoh versi main method: public static void main(String[]args)

public static void main(String args[])

public static void main(String argv[])

**identifier**

* Identifier adalah nama yang digunakan untuk kelas,method dan variabel.
* Identifier sebaiknya diawali A-Z atau a-z, dollar($), atau underscore(\_).
* Setelah di awali A-Z atau a-z, dollar($) atau underscore(\_) identifier boleh dikombinasikan dengan karakter lain.
* Contoh:\_nama123;

**Identifier(2)**

* sebuah keyword di java seperti; class,public,int dll tidak dapat dijadikan sebuah identifier.
* Identifier bersifat case sensitive.
* Contoh identifier yang salah; 1angka atau –nama.
* Contoh identifier yang benar; \_1angka atau $nama.

**Java**

Java adalah bahasa pemrograman yang dikembangkan oleh sun microsystem,yang diinisialisasi oleh james gosling tahun 1991 dan dirilis tahun 1995. Sebagai komponen inti dari dari system sun microsystem, java berjalan di berbagai macam platform system operasi seperti; linux,macOs, dan windows. Versi java terakhir adalah java 8. Java juga di kembangkan untuk aplikasi enterprise (javaEE) dan aplikasi mobile(java ME).

**Java Development Kit**

JDK adalah aplikasi yang digunakan untuk membuat program java, di dalam JDK terdapat java compiler yang digunakan untuk mengkompilasi kode java menjadi bytecode. Saat menginstal JDK, terinstal jga JRE. Java compiler akan mengubah kode-kode java menjadi bytecode(class).pada JVM terdapat java interpreter yang mengkonveksi bytecode menjadi program yang executable di platform tertentu dengan cara menginterpretasikan di platform mana program java di jalankan. Java interpreter mengkonveksi bytecode menjadi bahasa mesin yang sesuai dengan platform yang digunakan.

**Sejarah bahasa java**

James gosling menginisialisasi java project pada tahun 1991.awalnya disebut dengan Oak,lalu menjadi green dan akhir nya menjadi java prinsip java adalah write once,run anywhere. Pada tahun 2007,sun menjadikan java sebagai open-source project.

**Variabel di java**

Variabel merupakan sebuah tempat untuk menanpung data. Variabel adalah sebuah penyimpanan yang memiliki nama, yang dimana nama tersebut dapat dimanipulasi di dalam program setiap variabel memiliki tipe data yang spesifik untuk menentukan ukuran dan layout data yang disimpan di memori. Tipe data dari variabel juga menentukan berapa range nilai yang dapat di tamping di dalam memori.

**Variabel lokal**

Dideklarasikan di dalam method. Variabel lokal akan memesan memori ketika method tempat variabel lokal tersebut dipanggil dan dihancurkan ketika method tempat variabel lokal tersebut selesai di gunakan, modifier seperti public,protected dan private tidak dapat digunakan pada variabel lokal.

**Variabel instance**

* Variabel instance dideklarasikan di dalam kelas,tetapi di luar method.
* Variabel instance akan memesan memori ketika kelas tempat variabel instance tersebut diinstansiasi kan kepada sebuah objek.

Contoh; variabel var = new Variabel();

* Variabel instance dapat di gunakan oleh banyak method yang ada di dalam kelas.
* Modifier public,private,dan protected dapat di gunakan pada variabel instance.

**Variabel instance (2)**

* Disarankan memberikan akses private pada variabel instances.
* Akses variabel di dalam method static seperti main,dilakukan dengan cara(contoh1)

namaObjek.namaVariabel.

* Tetapi pada method bukan static,variabel dapat diakses secara langsung (contoh2).

**Modifiers di java**

Modifiers

* Modifiers adalah keyword yang ditambahkan kepada definisi tertentu untuk merubah arti definisi tersebut.
* Contoh modifier,public,protected,private,final,dll.
* Contoh penggunaan modifier;

Public.class Z {}

* Di java ada 2 jenis modifier,yaitu

Access modifiers

Non access modifiers

**Default**

* Defult tidak secara ekspilit dituliskan pada kelas,atribut,method dan kontruktor.
* Sebuah kelas,atribut,method,atau kontruktor yang tidak di beri keterangan access modifier akan menjadi default.
* Ketika kelas modifiers nya adalah default,maka kelas tersebut dapat diakses oleh kelas mana aja dalam package yang sama.

Contoh;

Int x=2,y=2;

Int tambah(int a, int y);

**Private**

* Private secara eksplisit di tuliskan pada kelas,atribut,method,atau kontruktor
* Ketika method modifiersnya adalah private,maka method tersebut hanya dapat diakses dari kelas itu saja.
* Private adalah access modifier yang sangat restrictive.
* Private tidak dapat di sematkan pada kelas dan interface,tetapi bisa pada inner-class.
* Dengan menggunakan private,objek mengbungkus dirinya dan menyembunyikan data dari kelas lain.

**Public**

Sebuah kelas,method atau atribut yang di set public akan dapat di akses oleh kelas lain.

Walaupun kelas adalah public,tetapi jika diakses dari package yang berbeda,maka perlu dilkukan meng-import package tempat kelas berada.

Contoh penggunaan public adalah “public class mahasiswa{}.

Method main pada java harus di set public.hal ini dilakukan agar java interpreter dapat mengenali class yang akan di jlankan.

**protected**

protected secara eksplisit dituliskan pada kelas,atribut,method atau kontruktor.

Ketika method,variabel dan kontruktor modifiersnya adalah protected,maka method,variabel dan kontruktor tersebut hanya dapat diakses dari kelas itu,turunan kelas dan kelas lain yang berada dalam 1 package.

Protected tidak dapat disematkan pada kelas dan interface,tetapi dapat disematkan pada inner class.

Method dan atribut pada interface juga tidak dapat di set protected,sebaliknya,method dan atribut pada kelas dapatt di set protected

**Variabel static**

Static keyword yang disematkan pada variabel menunjukan bahwa variabel tersebut dapat diakses tanpa menggunakan objek kelas dimana variabel tersebut berada.

Variabel static hanya dapat dideklarasikan di dalam kelas,bukan di dalam method atau sebagai variabel lokal.

Variabel static diakses dengan cara memanggil kelas tempat variabel static lalu dilanjutkan dengan nama variabel static.

Contoh;VariabelStatic.nama;

“nama”adalah variabel static.

**Method static**

* Static keyword yang disemtkan pada method menunjukan bahwa method tersebut dapat diakses tanpa mengunaka objek dari kelas di mana method tersebut berada.
* Method static tidak menggunakan instance variabel (variabel global) dari sebuah kelas,melainkan dari parameter method static tersebut.
* Method static diakses dengan cara memanggil kelas tempat method static lalu dilanjutkan dengan nama method static
* Contoh;VariableStatic.tanbah1()

tambah1()adalah method static.

**Kelas dan objek**

Kelas merupakan suatu bentuk klasidikasi dari objek objek yang ada di dunia ini.

Objek objek yang memiliki kesamaan dari segi data dan fungsi diklasifikasikan kepada kelas ternteu.

Contoh;ribuan kendaraan roda empat ukuran sedang di njalanan, yang dapat berjalan,berhenti,ngebut diklasifikasikan sebagai kelas mobil.

Objek didalamnya adalah Nissan,Toyota,Honda dll

**Kelas dan objek(2)**

Di dalam kelas terdefinisi data dan operasi dari setiap objek yang diklasifikasikan pada kelas tersebut.

* Kelas mobil memiliki data(ban,rem,body)dan operasi(ngebut,berhenti).
* Maka,objek mazda dan Toyota memiliki ban,rem,body,ngebut dan berenti
* Objek adalah things benda ,motor bebek,apapun dapat dikatan objek
* Objek adalh isntansi atau perwujudan dari kelas
* Artinya sebuah objek memiliki data dan fungsi yang memiliki oleh kelas

**Inhenritance**

* Salah satu konsep penting pada OOP adalah pewarisan atau inheritance
* Inheritance memumgkinkan kita membuat kelas yang di turunkan dari kelas lain.
* Inhenritance juga memungkinkan untuk menggunakan ulng kode program dan mempercept prosese implemtasi
* Contoh konsep inherintance adalah hubungan seorang ayah dan anak nya, anak memiliki data dan fungsi yang memiliki oleh ayah.
* Warna rambut jenis rambut,warna kulit hingga sifat

**Konsep dasar PBO**

Pemrograman berorientasi objek

* Menurut meyer(1997) system yang dibangun berdasarkan metode berorientasikan objek adalah sebuah system yang komponen komponen dienkapulasi(dibungkud)menjadi kelompok data dan fungsi yang dapat mewarisi atrribut dan sifat dari komponen lain nya dan komponen komponen tersebut berinteraksi satu sama lain.
* Membuat program secra OO tidk menggunakan bahsa OO.
* OO adalah cara pkikir nya.

**Overriding**

* Pada sub kelas dari hirarki inheritance,dimungkinkan un tuk memodifikasi atau mengimplement suatu method yang diturunkan dari super kelas.
* Proses ini di sebut dengan method overriding.
* Proses overriding disesuaikan dengan kebutuhan sub kelas.
* Method yang di overriding adalah method yang tidak bermodifier final.
* Overriding method berarti memodifikasi/ meng implement/mengupdate/menulis kembali/menggnti isi method yang sudah ada di suoer kelas