**Belajar java**

**Pengenalan java**

Dibuat oleh james gosling pada 1995. Java merupakan pemrograman berorientasi objek dan mendukung pengelolaan memori secara otomatis. Java dikenal dengan write once, run anywhere. Artinya Ketika sebuah code java dibuat dan decompile akan menghasilkan sebuah binary program. Nah binary program ini itu independent tidak terkait dengan system operasi. Untuk menjalankan binary java menggunakan jvm (java virtual machine).

JRE (java runtime environment) (Ketika akan melakukan deploying)

JDK (java development kit) (masih tahap pengembangan)

Proses development program java

Ketika kita menuliskan code java dengan extension File.java Ketika decompile akan menghasilklan java binary (File.class) file ini lah yang nanti akan diproses oleh JRE. Namun Ketika File.class sudah banyak maka akan disatukan menggunakan java archive (File.jar) yang nantinya JRE akan memprosesnya.

**Instalasi java**

Download jdk di openjdk karena free dan opensource. Bisa didownload di dalam jdk.java.net. dan dalam environment variable ubah JAVA\_HOME ke path folder yang sudah di download. Dan path nya ke bin di dalam folder java tersebut.

Ketika memasuki pertama kali intellij idea terdapat proses indexing yaitu untuk memudahkan IDE dalam pencarian file.

Tipe data di java

Variable dalam java yaitu static type yaitu Ketika membuat variable harus ditentukan tipe datanya jadi tidak dinamis seperti javascript.

Tipe data terbagi menjadi dua primitive dan non primitive. Primitive yaitu tipe data bawaan di program seperti int, char, Boolean sedangkan non primitive seperti String, Integer, Long, Float, Double. Note tipe data primitive selalu memiliki default value sedangkan non primitive default value selalu null. Non primitive data type mempunyai method atau function untuk memanipulasi variable tersebut.

Integer number

Byte (1 byte) -128 -> 127  
Short (2 byte) -32,769 -> 32, 767  
Int (4 byte) 2.10 pangkat 9  
Long (8 byte) 2.10 pangkat 18

Float (4 bytes) 3.4 pangkat 038  
Double (8 bytes) 1.7 pangkat 308

// kode literal  
int decimalInt = 10;  
int hexInt = 0xFFFFFF;  
int binaryDecimal = 0b10101010;  
  
// for readibiilty  
int satuJuta = 1\_000\_000;

Char merupakan tipe data untuk satu karakter saja   
Bool binary tipe data yang berisi true dan false (default value = false)

char firstWord = 'a'; // harus dengan single tik  
  
System.*out*.println(firstWord);  
  
boolean iniBool = false;

Tipedata non primitive

Integer angka = 5;  
Long angkaLong = 100L;  
  
Byte iniByte;  
iniByte = 50;

Array

// kumpulan tipe data yang sama  
// jumlah array tidak bisa dirubah ketika pertama kali diinisialisasi  
  
// 1st method to declare an array  
String[] stringArray;  
stringArray = new String[3];  
  
stringArray[0] = "Mujahid ";  
stringArray[1] = "Ansori ";  
stringArray[2] = "Majid ";  
  
System.*out*.println(stringArray[1]);  
  
int[] intArray = new int[]{  
 10,10,10  
};  
  
long[] longArray = {10L, 10L};  
  
System.*out*.println(intArray.length);  
  
// 2 dimensions array  
  
String[][] members = {  
 {"Mujahid", "Ansori"},  
 {"Farhan", "Nurzaman"},  
 {"Hilmy", "Apriadi"}  
  
};  
  
System.*out*.println(members[0][1]); // ansori

Expression = a piece of code that result a value;  
statement = instruction or action (condition and loops);

We can’t put a statement when an expression is expected

Example  
System.out.println(const x); // error  
var x = if(a < 2) { return 2}; // error

Percabganan

var absen = 70;  
var ujianAkhir = 75;  
  
if(absen >= 75 && ujianAkhir >= 75){  
 System.*out*.println("Selamat Anda lulus");  
}  
  
char nilai = 'A';  
switch(nilai){  
 case 'A':  
 System.*out*.println("lulus dengan sangat baik");  
 break;  
 case 'B':  
 System.*out*.println("lulus dengan sangat baik");  
 break;  
 case 'C':  
 System.*out*.println("lulus dengan sangat baik");  
 break;  
 case 'D':  
 System.*out*.println("lulus dengan sangat baik");  
 break;  
 case 'E':  
 System.*out*.println("lulus dengan sangat baik");  
 break;  
 default:  
 System.*out*.println("jelek banget ku");  
  
}

// dimulai pada java 14 kita dapat menggunakan lambda dalam swith case  
switch (nilai){  
 case 'A' ->System.*out*.println("bagus banget gila");  
 case 'B', 'C' -> System.*out*.println("gila sih");  
 default -> {System.*out*.println("jelek");}  
}  
  
// menggunakan yeild  
String ucapan = switch (nilai) {  
 case 'A':  
 yield "Bagus banget gila";  
 case 'B', 'C':  
 yield "mayanlah";  
 default:  
 yield "jelek banget gila";  
};  
  
System.*out*.println(ucapan);

Perulangan

for(int i=0;i<5;i++){  
 for(int j=0;j<i;j++){  
 System.*out*.print("\*");  
 }  
 System.*out*.println();  
}  
  
var nilai = 50;  
boolean masihRemed = nilai >= 70 ? false:true;  
while(masihRemed){  
 System.*out*.println("remedke");  
 nilai+=5;  
 masihRemed = nilai >= 70 ? false:true;  
}  
  
do{  
 System.*out*.println("akan dijalankan meskipun kondisi tidka terpenuhi");  
} while(masihRemed);  
  
String[] nama = {  
 "Mujahid", "ansori", "majid"  
};  
  
for (var name: nama  
) {  
 System.*out*.println(name);  
}

method

public static void main(String[] args) {  
 *sayHelloWorld*();  
 System.*out*.println(*addTwoNumbers*(2,3));  
  
 // tanpa variable arguments  
 int[] values = {80,80};  
 *sayCongrats*("Mujahid", values);  
 *sayCongratsVarargs*("Mujahid", 80, 80);  
}  
static void sayHelloWorld(){  
 System.*out*.println("hello world");  
}  
static int addTwoNumbers(int a, int b){  
 return a + b;  
}  
// method variable arguments, jumlah argument yang tidak tentu  
// harus berada di akhir  
static void sayCongrats(String name, int[] values){  
 int total = 0;  
 for(var value: values){  
 total += value;  
 }  
  
 int finalValue = total / values.length;  
 if(finalValue >= 75){  
 System.*out*.println("Selamat " + name + " Anda lulus");  
 } else {  
 System.*out*.println("Mohon maaf anda tidak lulus");  
 }  
}  
  
static void sayCongratsVarargs(String name, int... values){  
 int total = 0;  
 for(var value: values){  
 total += value;  
 }  
  
 int finalValue = total / values.length;  
 if(finalValue >= 75){  
 System.*out*.println("Selamat " + name + " Anda lulus");  
 } else {  
 System.*out*.println("Mohon maaf anda tidak lulus");  
 }  
}

**Overloading =** method dengan nama yang sama dengan parameter yang berbeda.

sebuah block program yang akan dijalankan Ketika dipanggil