Belajar Java Dasar

Maulana Ifandika

Struktur Program

Kode java memiliki struktur atau standar tersendiri dalam penulisan kode.

public = Ini dapat diakses secara bebas.

class = Ini untuk tipe class.

Main = nama dari file kode java.

static = Ini dapat diakses secara langsung.

void = Kosong atau tidak menerima hasil nilai.

String[] args = Array tipe String dengan nama ‘args’ untuk argumen ketika menjalankan program java.

main = Fungsi utama dari program java, gerbang utama program di jalankan.

public class Main {

public static void main(String[] args) {

…

}

}

Package

Package hanyalah istilah untuk folder pada program java, jika program sederhana maka tidakmasalah jika semua file program pada 1 folder, tapi jika program besar maka tidak efisien maka perlu pengelompokan berdasar tipe program, semisal tipe layanan, tampilan, penyimpana data.

Contoh pada gambar di atas, ada package ui, databse, service, dll. Pada file kode java kata kunci ‘package’ untuk memberitahu file ini berada di package tersebut.

package database;

public class Database {

…

}

package service;

public class Service {

…

}

package ui;

public class UI {

…

}

Jika ingin memasukan file pada lokasi berbeda cukup tulis ‘import <lokasi/nama\_file>’.

import database.\*; // Semua file pada folder database

import ui.UI; // File UI pada folder UI

import service.Service; // File service pada folder service

public class All {

…

}

IO/Input & Output

Output

Input/masukan & Output/Keluaran pada java untuk input nilai lalau menampilkan hasil/output. Untuk IO harus memanggil ke sistem lalu jenisnya, apakah untuk input atau output dengan kata kunci ‘System’ lalu tipe ‘System.in’ atau ‘System.out’. Untuk output/out sendiri ada beberapa jenis, ada ‘println’, ‘print’, ‘printf’. Jika ‘print’ maka hanya tampilkan hasil tanpa membuat baris baru/enter maka output selanjutnya akan berada di sampingnya. Jika ‘println’ maka akan membuat baris baru, output selanjutnya ada di bawah. Jika ‘printf’ untuk output dengan format output lalu memasukan nilai yang akan di tampilkan.

public class InputOutput {

public static void main(String[] args) {

System.out.println("Baris 1: println"); // .println = print dan menambahkan baris baru

System.out.print("Baris 2: print"); // .print = hanya print saja, tidak menambahkan baris baru

System.out.print(" Baris 3: print");

System.out.print("\nBaris 4: print dengan enter \n"); // \n = new line, enter secara manual

System.out.printf("Baris 5: 1 + 1 = %d", 2); // .printf = Untuk output format

System.out.print("\n");

}

}

Input

Input menggunakan ‘.in’, objek untuk menangkap dari input contoh Scanner.

public class Input {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int a = scanner.nextInt();

System.out.println(a);

}

}

Tipe Data

Sebuah jenis wadah yang digunakan untuk menyimpan sebuah nilai, semisal tipe data karakter menyimpan karakter/kalimat, tipe data bilangan menyimpan nilai bilangan 1, 100, 2.500, dll. Tipe data boolean menyimpan nilai true atau false. Tipe data dibagi menjadi 2 yaitu tipe data primitif dan objek.

Tipe primitif tipe biasa yang hanya dapat menyimpan nilai jika tipe objek dapat melakukan operasi terhadap nilai yang disimpan seperti membalikkan nilai, dan operasi lainnya. Penulisan tipe data primitif pada Java bertipe lowercase.

1. Tipe Data Primitif

- Short (short) Tipe Data Bilangan

- Integer (int)

- Floating (float)

- Double (double)

- Long (long)

- Character (char) Tipe Data Karakter/Huruf

- Boolean (boolean)

2. Tipe Data Objek

- Integer (Integer)

- Double (Double)

- String (String)

- Boolean (Boolean)

public class TipeData {

public static void main(String[] args) {

int varInt = 100;

System.out.println("=== integer ===");

System.out.println("nilai = " + varInt);

System.out.println("Min Range = " + Integer.MIN\_VALUE);

System.out.println("Max Range = " + Integer.MAX\_VALUE);

System.out.println("size = " + Integer.BYTES + " byte");

System.out.println("size = " + Integer.SIZE + " bit");

byte varByte = 10;

System.out.println("\n=== byte ===");

System.out.println("nilai = " + varByte);

System.out.println("Min Range = " + Byte.MIN\_VALUE);

System.out.println("Max Range = " + Byte.MAX\_VALUE);

System.out.println("size = " + Byte.BYTES + " byte");

System.out.println("size = " + Byte.SIZE + " bit");

short varShort = 120;

System.out.println("\n=== short ===");

System.out.println("nilai = " + varShort);

System.out.println("Min Range = " + Short.MIN\_VALUE);

System.out.println("Max Range = " + Short.MAX\_VALUE);

System.out.println("size = " + Short.BYTES + " byte");

System.out.println("size = " + Short.SIZE + " bit");

long varLong = 1234567L;

System.out.println("\n=== long ===");

System.out.println("nilai = " + varLong);

System.out.println("Min Range = " + Long.MIN\_VALUE);

System.out.println("Max Range = " + Long.MAX\_VALUE);

System.out.println("size = " + Long.BYTES + " byte");

System.out.println("size = " + Long.SIZE + " bit");

float varFloat = 12.3f;

System.out.println("\n=== float ===");

System.out.println("nilai = " + varFloat);

System.out.println("Min Range = " + Float.MIN\_VALUE);

System.out.println("Max Range = " + Float.MAX\_VALUE);

System.out.println("size = " + Float.BYTES + " byte");

System.out.println("size = " + Float.SIZE + " bit");

double varDouble = 15.51d;

System.out.println("\n=== double ===");

System.out.println("nilai = " + varDouble);

System.out.println("Min Range = " + Double.MIN\_VALUE);

System.out.println("Max Range = " + Double.MAX\_VALUE);

System.out.println("size = " + Double.BYTES + " byte");

System.out.println("size = " + Double.SIZE + " bit");

char varChar = 'C';

System.out.println("\n=== char ===");

System.out.println("nilai = " + varChar);

System.out.println("Min Range = " + Character.MIN\_VALUE);

System.out.println("Max Range = " + Character.MAX\_VALUE);

System.out.println("size = " + Character.BYTES + " byte");

System.out.println("size = " + Character.SIZE + " bit");

boolean varBoolean = true;

System.out.println("\n=== boolean ===");

System.out.println("nilai = " + varBoolean);

System.out.println("Min Range = " + Boolean.FALSE);

System.out.println("Max Range = " + Boolean.TRUE);

// System.out.println("size = " + Boolean.BYTES + " byte");

// System.out.println("size = " + Boolean.SIZE + " bit");

int numDecimal = 35;

int hexaDecimal = 0xFFFFF;

int binaryDecimal = 0b01010110;

System.out.println(numDecimal);

System.out.println(hexaDecimal);

System.out.println(binaryDecimal);

// mengunakan underscore

double harga = 1\_000\_000\_000;

System.out.println(harga);

System.out.println("\n");

/\* Tipe Data Bukan Primitive/Object \*/

// pada tipe data bukan primitive memiliki method karena ia adalah object

// nilai default pada tipe data bukan primitive adalah null(kosong)

Byte iniByte = 100;

System.out.println(iniByte);

Short iniShort = 1000;

System.out.println(iniShort);

Integer iniInteger = 10000;

System.out.println(iniInteger);

Long iniLong = 100000L;

System.out.println(iniLong);

Float iniFloat = 1.123F;

System.out.println(iniFloat);

Double iniDouble = 1.12345;

System.out.println(iniDouble);

Character iniCharacter = 'A';

System.out.println(iniCharacter);

Boolean iniBoolean = true;

System.out.println(iniBoolean);

String iniString = "kipli";

System.out.println(iniString);

}

}

Output

$ javac TipeData.java && java TipeData

=== integer ===

nilai = 100

Min Range = -2147483648

Max Range = 2147483647

size = 4 byte

size = 32 bit

=== byte ===

nilai = 10

Min Range = -128

Max Range = 127

size = 1 byte

size = 8 bit

=== short ===

nilai = 120

Min Range = -32768

Max Range = 32767

size = 2 byte

size = 16 bit

=== long ===

nilai = 1234567

Min Range = -9223372036854775808

Max Range = 9223372036854775807

size = 8 byte

size = 64 bit

=== float ===

nilai = 12.3

Min Range = 1.4E-45

Max Range = 3.4028235E38

size = 4 byte

size = 32 bit

=== double ===

nilai = 15.51

Min Range = 4.9E-324

Max Range = 1.7976931348623157E308

size = 8 byte

size = 64 bit

=== char ===

nilai = C

Min Range =

Max Range = ?

size = 2 byte

size = 16 bit

=== boolean ===

nilai = true

Min Range = false

Max Range = true

35

1048575

86

1.0E9

100

1000

10000

100000

11.23

1.12345

A

true

kipli

Variabel

Variabel adalah wadah yang digunakan untuk menyimpan nilai atau tipe data yang sudah ditentukan, disebut variable karena data yang disimpan dapat diubah nilainya. Setiap deklarasi variable diberi nama agar membedakan variable 1 dan lainnya. Penulisan variabel adalah

<tipe\_data> <nama\_variabel> = <nilai\_yang\_dimasukan>

Contoh

int iniVariabel = 100;

public class Variabel {

public static void main(String[] args) {

// inisialisasi variabel

int variabel\_1 = 100; // deklarasi variabel dan langsung memberikan nilai

System.out.println("nilai variabel 1 = " + variabel\_1);

// merubah data/nilai variabel\_1

variabel\_1 = 50;

System.out.println("nilai baru variabel 1 = " + variabel\_1);

// hanya deklarasi variabel saja

int variabel\_2;

variabel\_2 = 200;

System.out.println("nilai variabel 2 = " + variabel\_2);

// kata kunci var, finally

// var = sebuah keyword untuk variabel bertipe apapun, namun harus langsung di assign

// finally = keyowrd mirip const/konstanta pada bahasa p lain, yaitu sebuah variabel konstan

// yang tidak dapat diubah.

// java 10

// var num; -> eror

// num = 10;

var varOtomatis = "ini keyword var"

System.out.println(varOtomatis);

finally int varFinally = 100;

// varFinally = 200; -> error

System.out.println(varFinally);

}

}

Output

$ java Variabel

nilai variabel 1 = 100

nilai baru variabel 1 = 50

nilai variabel 2 = 200

ini keyword var

100

Konversi Tipe Data

public class Konversi {

public static void main(String[] args) {

int varInt = 100;

System.out.println("varInt = " + varInt);

// Konversi ke tipe data/rentang data yang lebih besar

long varLong = varInt;

System.out.println("varLong = " + varLong);

short varShort = 300;

// Konversi ke tipe data/rentang data yang lebih kecil

// ketika konversi ke yang lebih kecil maka data yang akan di konversi harus di casting

byte varByte = (byte)varShort;

System.out.println("varByte = " + varByte);

/\* casting operator \*/

// merubah suatu tipe data ke tipe data lain

int pembagi = 10;

int dibagi = 4;

// jika salah satu bilangan pecahan maka hasil berupa bilangan pecahan

float hasil = (float)pembagi / dibagi;

// % = untuk format

// %d = desimal

// %f = float

System.out.printf("%d / %d = %f\n", pembagi, dibagi, hasil);

short val\_1 = 20;

float val\_2 = 3f;

float result = val\_1 / val\_2;

System.out.printf("%d / %f = %f\n", val\_1, val\_2, result);

}

}

Aritmatika

Aritmatika adalah operasi seperti penjumlahan, pengurangan, perkalian, dan pembangian. Pada bahasa pemprograman ada beberapa operasi yang simbol dirubah jika pejumlahan menggunakan simbol + maka pada bahasa pemprograman tetap +, jika operasi perkalian maka menggunakan simbol x jika pada bahasa pemprograman menggunakan simbol \*.

public class Aritmatika {

public static void main(String[] args) {

int x = 10;

int y = 3;

int hasil;

// penjumlahan +

hasil = x + y;

System.out.printf("%d + %d = %d\n", x, y, hasil);

// pengurangan -

hasil = x - y;

System.out.printf("%d - %d = %d\n", x, y, hasil);

// perkalian \*

hasil = x \* y;

System.out.printf("%d \* %d = %d\n", x, y, hasil);

// pembagian /

hasil = x / y;

System.out.printf("%d / %d = %d\n", x, y, hasil);

// modulu %

hasil = x % y;

System.out.println(x + " % " + y + " = " + hasil);

}

}

Unary Operator

Sebuah operasi kepada nilai tunggal yang digunakan untuk menghasilkan nilai baru. Unary digunakan ketika ingin merubah suatu nilai dengan hanya 1 operasi, atau penyederhanaan opersi. Jika sebuah nilai a = 10 lalu kita ingin merubah nilai tersebut, semisal a + 5 maka a = 15. Jenis unary operator ada 4:

1. Unary minus

2. Increment

3. Decrement

4. NOT

public class Unary {

public static void main(String[] args) {

// operator unary = operator yang tidak merubah nilai

// + (menjadi positif)

// - (menjadi negatif)

int ageUcup = 97;

int ageKipli = 55;

// before

System.out.println("ageUcup " + +ageUcup);

System.out.println("ageKipli " + -ageKipli);

// after

System.out.println("ageUcup " + ageUcup);

System.out.println("ageKipli " + ageKipli);

// operator increment dan decrement, operator yang dpt merubah nilai

// prefix = ++data , --data

// post = data++ , data--

int num1 = 5;

int num2 = 5;

// prefix

System.out.println("prefix ++ = " + ++num1);

System.out.println("prefix -- = " + --num2);

num1 = 5;

num2 = 5;

// postfix

System.out.println("postfix ++ = " + num1++);

System.out.println("postfix -- = " + num2--);

// operator boolean !

boolean numBool1 = true;

boolean numBool2 = false;

System.out.println("numBool1 = " + !numBool1);

System.out.println("numBool2 = " + !numBool2);

}

}

Scope

// global scope file

public class Scope {

// global scope class

int dataGlobal = 12345;

public static void main(String[] args) {

// scope / jangkuan sebuah data / variabel

// - global scope

// - local scope

// - block scope

// local scope yang hanya bisa di akses di dalam method main

int nilaiInt = 12345;

{

// block scope, hanya bisa di akses di dalam block ini saja

String name = "Kipli";

System.out.println(name);

}

System.out.println(nilaiInt);

// System.out.println(name); // error karena data berada pada block

printHello();

printWord();

}

static void printHello() {

// ini scope local pada method

String output = "Hello";

System.out.print(output);

}

static void printWord() {

// ini scope local pada method

String output = " Word";

System.out.println(output);

}

}

Operator Logika

Jika suatu kondisi dimana kondisi terpenuhi dengan syarat harus terpenuhi, jika salah satu syarat tidak terpenuhi maka kondisi tidak bisa dilakukan. Semisal kita ingin minum maka syarat yang harus dipenuhi adalah ada air dan wadah, jika salah satu tidak ada maka kita tidak bisa minum secara normal. Operator Logika digunakan untuk membuat sebuah kondisi, jenis operator logika ada 5 yang utama NOT, AND, OR, NAND, NOR, XOR, XNOR.

// Operator Logika Java

public class Logika {

public static void main(String[] args) {

// operasi logika

int a = 10;

int b = 5;

boolean hasil;

// AND = &&

System.out.println(">> AND = && <<");

hasil = (a == 5 && b == 10);

System.out.println("false && false = " + hasil);

hasil = (a == 10 && b == 9);

System.out.println("true && false = " + hasil);

hasil = (a == 3 && b == 5);

System.out.println("false && true = " + hasil);

hasil = (a == 10 && b == 5);

System.out.println("true && true = " + hasil);

100

// OR = ||

System.out.println(">> OR = || <<");

hasil = (a == 5 || b == 10);

System.out.println("false && false = " + hasil);

hasil = (a == 10 || b == 9);

System.out.println("true && false = " + hasil);

hasil = (a == 3 || b == 5);

System.out.println("false && true = " + hasil);

hasil = (a == 10 || b == 5);

System.out.println("true && true = " + hasil);

// XOR = ^

System.out.println(">> XOR = ^ <<");

hasil = (a == 5 ^ b == 10);

System.out.println("false && false = " + hasil);

hasil = (a == 10 ^ b == 9);

System.out.println("true && false = " + hasil);

hasil = (a == 3 ^ b == 5);

System.out.println("false && true = " + hasil);

hasil = (a == 10 ^ b == 5);

System.out.println("true && true = " + hasil);

// NOT = !

System.out.println(">> NOT = ! <<");

hasil = !(a == 10);

System.out.println("true = " + hasil);

hasil = !(b == 9);

System.out.println("false = " + hasil);

}

}

// Output

>> AND = && <<

false && false = false

true && false = false

false && true = false

true && true = true

>> OR = || <<

false && false = false

true && false = true

false && true = true

true && true = true

>> XOR = ^ <<

false && false = false

true && false = true

false && true = true

true && true = false

>> NOT = ! <<

true = false

false = true

Komparasi