

# Tutorial Bahasa Java

## Bagian 1 (Dasar-Dasar Java)

Lab Basdat Ilmu Komputer dan Pendidikan Ilmu Komputer UPI

Yudi Wibisono ([yudi@upi.edu](mailto:yudi@upi.edu))

Versi dok (2.0.0 / Feb 2016)



<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Modul ini bebas dicopy, didistribusikan, ditransmit dan diadaptasi/modifikasi/diremik dengan syarat tidak untuk komersial, pembuat asal tetap dicantumkan dan hasil modifikasi dishare dengan lisensi yang sama.

# Daftar Isi

## Daftar Isi

### Prasyarat

Mengapa belajar Java?

Kelebihan Java

Java Platform

IDE

### Instalasi Java dan IntelliJ

### Program Pertama: Hello World

### Dasar-Dasar Bahasa Java

Tipe

Operator

Konversi Antar Type

String

Membentuk String

Input dari Keyboard

Formating Output

Conditional Statement (IF-THEN-ELSE)

LOOP (WHILE-DO)

LOOP (DO-WHILE )

LOOP (FOR)

Keluar Paksa dari Loop: Break

Melewati Loop: Continue

Array

Array: For-Each

Array: Sorting

Array: Dua Dimensi

Generic

### Collection

## Prasyarat

Diasumsikan pengguna modul ini telah mengetahui dasar-dasar pemrograman dan struktur data.

## ***Mengapa belajar Java?***

Java diperkenalkan sejak tahun 1995 dan berkembang menjadi salah satu bahasa pemrograman yang paling populer baik di kalangan industri maupun akademik. Sejak 2007, Java juga digunakan untuk mengembangkan native Android app. Walaupun dapat digunakan untuk membuat aplikasi desktop dan front end web (via applet), Java saat ini umumnya digunakan dibagian backend (server side).

## Kelebihan Java

Ada beberapa keunggulan Java yang membuatnya menjadi salah satu bahasa pemrograman utama [Horstmann]:

**Sederhana:** perbaikan dari C++. Tidak ada header file, pointer, operator overloading, multiple inheritance. Walaupun demikian, Java masih lebih rumit dan lebih sulit dipelajari dibandingkan VB dan PHP.

**Object Oriented:** Mirip dengan C++, kecuali untuk multiple inheritance yang diganti Java dengan Interface.

**Berkinerja tinggi:** JIT (Just In Time) compiler merubah bytecode menjadi machine code sehingga aplikasi dapat dijalankan lebih cepat.

**Network-Savvy:** Mudah untuk membuat program yang memanfaatkan TCP/IP, HTTP dan FTP.

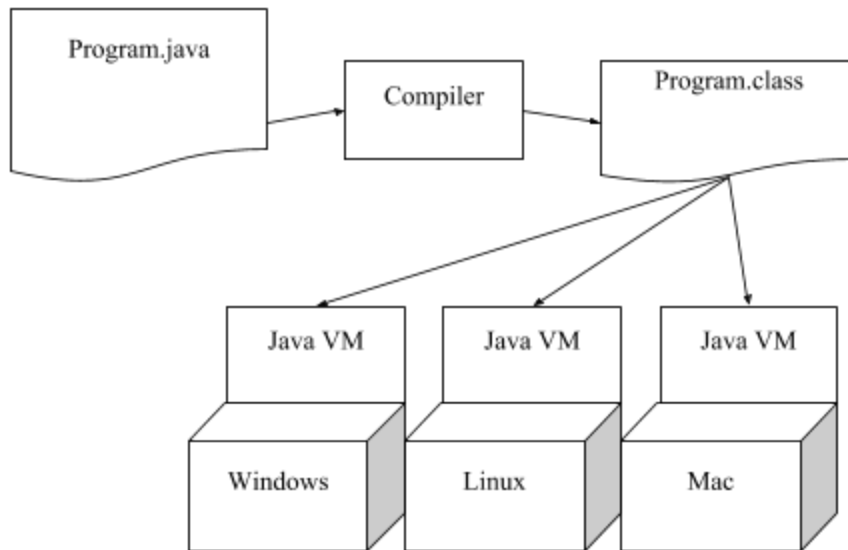
**Robust:** Mengurangi error dengan menghilangkan pointer dan melakukan early checking.

**Multithreaded:** Multithreading mudah dilakukan di Java, fitur ini sangat bermanfaat untuk memaksimalkan multicore processor.

**Aman:** mencegah overrunning runtime stack, corrupting memory, read dan write file tanpa ijin.

**Architecture Neutral dan Portable:** Menggunakan virtual machine dengan byte code, memudahkan Java untuk digunakan di mesin lain. Code Java akan dicompile menjadi

byte code (.class) yang kemudian dijalankan oleh JVM (Java Virtual Machine), lihat gambar di bawah.



## ***Java Platform***

Java mencakup banyak aspek sehingga dapat membingungkan. Istilah-istilah seperti J2SE, J2EE, J2ME, Struts, Spring, Seam, Servlet, Applet, Hibernate, Ibatis, Pojo, JSF, JPA dapat terlihat mengerikan. Kuncinya adalah tidak semua hal mengenai Java harus dipelajari sekaligus. Yang terpenting adalah dasarnya dulu, kemudian yang lain dapat dipelajari sesuai dengan kebutuhan.

## ***IDE***

IDE (*Integrated Development Environment*) dapat digunakan untuk membantu pengembang dalam proses edit, compile sampai debug. IDE Java yang paling populer adalah IntelliJ, Netbeans dan Eclipse. Dalam modul ini yang digunakan adalah IntelliJ, walaupun tidak masalah jika digunakan IDE lain.

Kelebihan IntelliJ adalah stabil, cepat dan mempunyai banyak fitur. IntelliJ juga digunakan di Android Studio. Kelemahannya? bukan aplikasi opensource dan hanya gratis untuk versi community.

## Instalasi Java dan IntelliJ

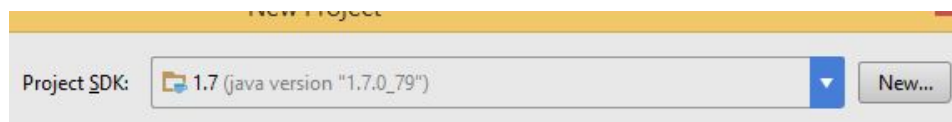
Download JDK (Java Development Kit) untuk Java SE (Standard Edition) dari <http://www.oracle.com/technetwork/java/javase/downloads/index.html> jangan lupa jika Anda mempunyai prosesor 64bit, pilih yang sesuai. Lakukan instalasi. Catat direktori tempat JDK ini disimpan.

Selanjutnya download IntelliJ versi community di <https://www.jetbrains.com/idea/download> Lakukan instalasi.

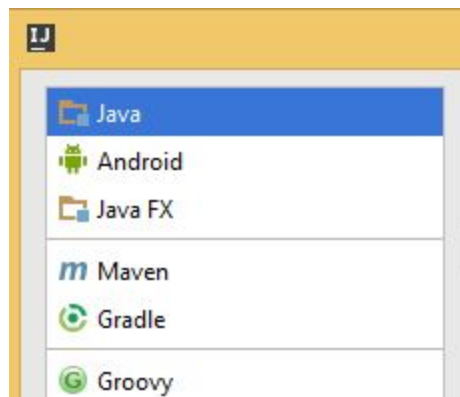
## Program Pertama: Hello World

Sesuai tradisi, pertama kita akan membuat aplikasi HelloWorld.

Buka IntelliJ pilih New Java Project. Selanjutnya pilih versi JDK dengan mengklik “New” (gambar bawah) lalu pilih direktori Java SDK yang sudah diinstall sebelumnya. Ini hanya perlu dilakukan sekali.



Pilih jenis project dengan “Java” (gambar bawah) lalu tekan Next.

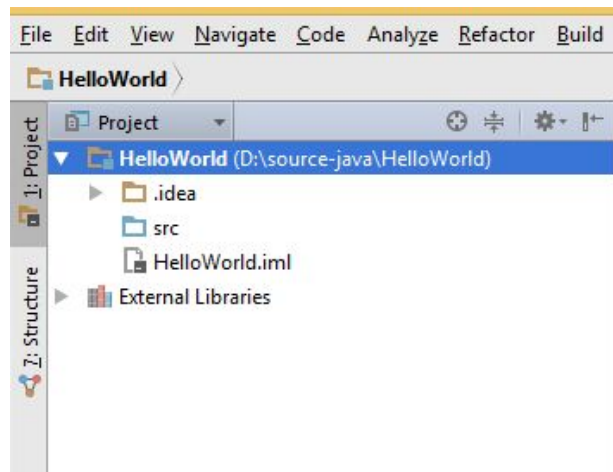


Tekan lagi Next saat ditanya template.

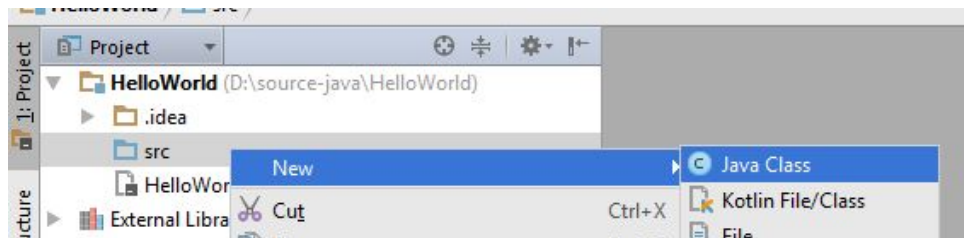
Isi project name dengan HelloWorld, pilih lokasi tempat source code ini (gambar bawah) kemudian tekan button “Finish”



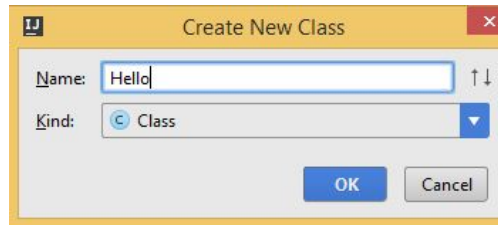
Kita akan masuk ke editor. Lihat project explorer dibagian kiri atas, expand project “HelloWorld”



Pilih src (source), klik kanan, New → Java Class (gambar bawah)



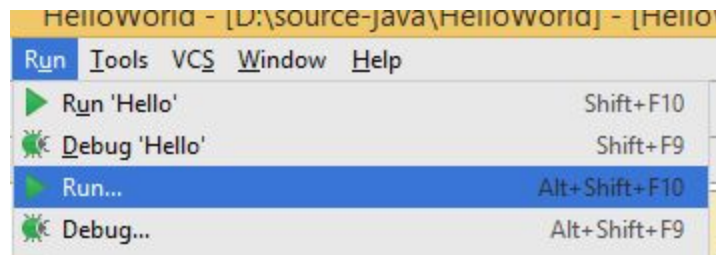
Isi nama class dengan “Hello” (gambar bawah)



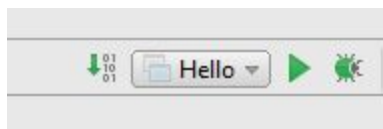
IDE akan menambahkan class Hello yang kosong. Isi class ini dengan method main sebagai berikut:

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Jalankan program dengan menekan Alt-Shift-F10 atau Run → Run dan pilih class Hello



Untuk selanjutnya, menjalankan program dapat dilakukan dengan mengklik tombol play di kanan atas (gambar bawah)



## Tugas 1

Modifikasi program tersebut sehingga menampilkan tiga baris:

Selamat pagi,

Nama saya [nama anda]

Java tidak sulit, saya akan menguasainya!.

## Dasar-Dasar Bahasa Java

Sebelum memulai ada beberapa hal yang penting tentang bahasa Java:

- Case sensitive untuk semuanya, bahkan nama file. Contoh: `namaMahasiswa` berbeda dengan `NamaMahasiswa`!
- Semua variabel harus dideklarasikan tipenya.
- Komentar di program:  
satu baris, gunakan double slash: `//`  
lebih dari satu baris, gunakan `/* ... */`

### Tipe

Seperti halnya C dan C++, Java adalah *strongly typed language*, yang berarti setiap variabel harus dideklarasikan tipenya (berbeda dengan PHP dan Javascript). Untuk program yang kecil memang kadang terasa mengganggu harus mendeklarasikan tipe setiap variabel, tapi saat program semakin besar, deklarasi tipe akan mengurangi kemungkinan terjadinya kesalahan.

Di Java terdapat delapan tipe dasar: 4 integer, 2 float, 1 character dan boolean.

Tipe integer terdiri atas:

- **int**: mempunyai rentang antara -2 milyar sd 2 milyar
- **short**: -32768 sd 32768
- **byte**: -128 sd 128
- **long**: -9 juta trilyun sd 9 juta trilyun

Int adalah tipe yang paling umum digunakan, short dan byte biasanya digunakan dalam array berjumlah besar untuk mengurangi penggunaan memori.

Tipe float digunakan untuk merepresentasikan bilangan pecahan. Tipe float terdiri atas:

- **float**: 6 sampai 7 angka signifikan
- **double**: 15 angka signifikan

Double adalah tipe data float yang paling umum digunakan

Tipe boolean memiliki hanya memiliki dua nilai: true dan false.

### Variabel

Variabel digunakan untuk menampung nilai di dalam memori. Deklarasi variabel di Java berformat: `tipe-variabel nama-variabel;`

Berikut adalah contoh deklarasi variabel.



Buatlah project baru seperti langkah sebelumnya. Beri nama project AplikasiTipe. Kemudian masukan program sebagai berikut:

```
public static void main(String[] args) {
    int kode;
    int umur=25; //variabel dapat langsung diinisiasi
    boolean isDibawahUmur; //perhatikan penulisan nama variael

    kode = 1234; //pengisian variabel (assignment)
    double gaji; //deklarasi variabel dapat dimana saja
    gaji = 5500000.23;
    isDibawahUmur = true;
    System.out.println("Umur:"+umur);
    System.out.println("Gaji:"+gaji);
}
```

Penulisan nama variabel di Java umumnya menggunakan model Camel, yaitu dimulai dari huruf kecil, lalu setiap pergantian kata menggunakan huruf besar. Misalnya totalGaji, jumlahSemuaBilangan.

## Operator

Java menyediakan operator standard: <, >, <=, >=, +, \*, -, /, dan % untuk mod.

Operator increment dan decrement: ++, --

Operator boolean:

```
== (sama) ,
!= (tidak sama),
&& (AND)
|| (OR)
! (NOT) misalnya !(x > y) artinya not(x > y)
```

Modifikasi contoh di atas sebagai berikut:

```
public static void main(String[] args) {
    int kode;
    int umur=25; //variabel dapat langsung diinisiasi
    boolean isDibawahUmur; //perhatikan penulisan nama variael

    kode = 1234; //pengisian variabel
    System.out.println("Kode:"+kode);
    kode++; //increment, sama dengan kode=kode+1
    System.out.println("Kode setelah ditambah:"+kode);

    isDibawahUmur = ( umur < 17 ); //operator boolean
}
```

```

    System.out.println("Dibawah umur?" + isDibawahUmur);
    boolean is20Tahun = (umur == 20);
    System.out.println("Dua puluh tahun?" + is20tahun);
}

```

Operator ++ dan -- dapat diletakkan di belakang dan depan variabel (a++ dan ++a). Perbedaan variabel++ dan ++variabel dapat dilihat pada contoh berikut:

```

public class Hello {
    public static void main(String[] args) {

        int a,b,c;
        a = 1;

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

        ++a; //tidak ada bedanya dengan a++ kalau seperti ini
        System.out.println("nilai a="+a);

        b = a++; //a masuk dulu ke b, baru ditambah 1
        System.out.println();
        System.out.println("nilai a="+a);
        System.out.println("nilai b="+b);

        c = ++a; //a ditambah 1 dulu, baru masuk ke b
        System.out.println();
        System.out.println("nilai a="+a);
        System.out.println("nilai c="+c);
    }
}

```

## Konversi Antar Type

Cobalah contoh dibawah, berapa nilai c?

```

public static void main(String[] args) {
    int a = 10;
    int b = 3;
    double c = a/b;
    System.out.println("Nilai C="+c);
}

```

Nilai c tidak sesuai harapan karena hasilnya menjadi integer. Untuk memperoleh hasil yang diinginkan harus dilakukan casting terlebih dulu. Casting adalah “memaksa” suatu tipe menjadi tipe yang lain.

```
double c = (double) a/b; //a dicasting menjadi double
```

Cara lain adalah dengan membuat tipe b menjadi double

```
double c = a / (double) b; //b dicasting menjadi double
```

## **Tugas 2**

Apa yang terjadi jika statement diatas diubah menjadi:

```
double c = (double) (a / b);
```

Mengapa?

## **Tugas 3**

Buatlah program yang menghitung luas ( $p \times l$ ) dan keliling segi empat ( $2p + 2l$ ). Variabel panjang dan lebar diisi dalam program. Keluaran program adalah sebagai berikut:

```
panjang = xxx  
lebar = yyy  
luas = xxx  
keliling = xxx
```

## ***String***

Berbeda dengan beberapa bahasa seperti Objek Pascal Delphi, string di Java bukan merupakan tipe dasar tetapi sebuah class. Class akan dibahas lebih detail pada materi object oriented. Tapi untuk sekarang, anggap saja class adalah sebuah tipe data khusus.

Contoh berikut memperlihatkan cara mendeklarasikan variabel bertipe String, yaitu mengambil sebagian string (substring) dan menggabungkan (concatenation).

```
public static void main(String[] args) {  
    String nama, gelar; //nama dan gelar adalah objek String  
    nama = "Ahmad Aulia";  
    gelar = "MT";  
    //substring  
    //ambil nama mulai dari pos 0, sebanyak 4 huruf  
    String namaPotong = nama.substring(0,4);  
    System.out.println("Nama Potong="+namaPotong);  
}
```

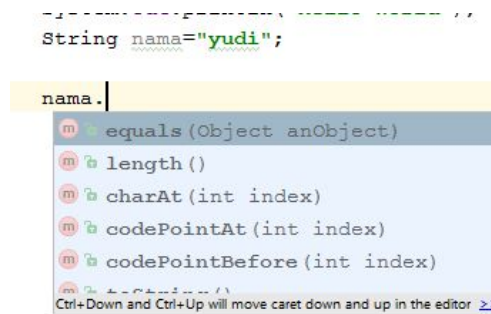
```

//penggabungan
String namaLengkap = nama+", "+gelar; // + untuk menggabung
System.out.println("Nama Lengkap: "+namaLengkap);

}

```

Perhatikan penggunaan **nama.substring**. Substring merupakan method (fungsi) yang disediakan class String. Selain substring masih banyak lagi method lain, misalnya uppercase, lowercase. Untuk info lengkapnya, ketik `nama.` (nama lalu diakhiri dengan titik) dan tunggu sebentar. IntelliJ akan menampilkan method yang dapat digunakan (gambar bawah).



Untuk membandingkan dua string, tidak dapat digunakan operator `==` (dua kali samadengan) melainkan harus dengan method `equals`. Cobalah code berikut:

```

public static void main(String[] args) {
    String nama1="Budi Martami";
    String nama2="Akhmad Aulia";

    if (nama1==nama2) { //BERBAHAYA, JANGAN GUNAKAN ==
        System.out.println("nama sama");
    } else System.out.println("nama beda");
}

```

Pada contoh diatas, penggunaan `==` untuk membandingkan string tidak menemui masalah. Tapi cobalah program berikut, apa hasilnya?

```

public static void main(String[] args) {
    String s = "satu";
    s = s + ""; // objek s dicreate lagi
    System.out.println(s);
    if (s=="satu") {
        System.out.println("string sama");
    }
    else {

```

```

        System.out.println("string berbeda");
    }
}

```

Hasilnya akan salah karena operator `==` membandingkan lokasi objek (apakah dua variabel menunjuk ke lokasi objek yang sama), bukan kesamaan string. Kalau kebetulan lokasinya sama, maka hasilnya akan benar, tapi ini akan berbahaya.

Untuk membandingkan string, selalu gunakan method `equals` seperti contoh dibawah:

```

public static void main(String[] args) {
    String nama1="Budi Martami";
    String nama2="Akhmad Aulia";

    if (nama1.equals(nama2)) { //BENAR!!
        System.out.println("nama sama");
    } else System.out.println("nama beda");
}

```

## Membentuk String

Seringkali kita membutuhkan penggabungan string dari string yang lain. Contoh berikut bekerja, tetapi tidak efisien (lebih lambat) karena setiap kali nama diisi (di-assign), maka akan dicreate variabel baru.

```

public static void main(String[] args) {
    String nama1="Budi Martami";
    String nama2="Ahmad Aulia";
    String nama3="Elfan Nov";

    String nama;
    nama = nama1;
    nama = nama + nama2; //nama dicreate lagi di memori
    nama = nama + nama3; //nama dicreate lagi di memori
    System.out.println("Nama:"+nama);
}

```

Solusi yang lebih efisien adalah dengan menggunakan class `StringBuilder`. Code diatas dapat ditulis ulang sebagai berikut:

```

public static void main(String[] args) {
    String nama1="Budi Martami";
    String nama2="Ahmad Aulia";
    String nama3="Elfan Noviari";

    StringBuilder nama = new StringBuilder();
    nama.append(nama1);
}

```

```

        nama.append(nama2);
        nama.append(nama3);
        System.out.println("Nama:"+nama);
    }

```

Perhatikan pembuatan objek nama:

```

StringBuilder nama = new StringBuilder();

```

Artinya kita menciptakan sebuah objek nama yang bertipe `StringBuilder`. Sebuah objek sebelum dapat digunakan, harus dicreate terlebih dulu dengan menggunakan keyword `new`. (Mengapa object bertipe `String` tidak perlu dicreate? Karena class `String` adalah class khusus yang mendapat perlakuan berbeda)

Inisialisasi objek akan dibahas lebih detail di materi pemrograman berorientasi objek.

## ***Input dari Keyboard***

Class `Scanner` dapat digunakan untuk menerima input dari keyboard. Contoh berikut memperlihatkan penggunaan class ini.

```

public static void main(String[] args) {
    Scanner in = new Scanner(System.in);

    System.out.println("Siapa nama anda?");
    String nama = in.nextLine();

    System.out.println("Berapa umur anda?");
    int umur = in.nextInt();

    System.out.println("Berapa gaji anda?");
    double gaji = in.nextDouble();

    System.out.println("Nama:"+nama);
    System.out.println("Umur:"+umur);
    System.out.println("Gaji:"+gaji);

}

```

Jika muncul warna merah, artinya ada kesalahan pada program. Class `Scanner` tidak dikenali karena berada di class terpisah. Kita harus import class ini. Arahkan kursor pada kata yang berwarna merah, tunggu maka akan keluar hint berwarna biru seperti gambar dibawah. Tekan `Alt-Enter`

A screenshot of an IDE showing a code completion popup. The popup is titled "? java.util.Scanner? (multiple choices...) Alt+Enter". Below the title, the text "public static void main(String[] args) {" is visible, followed by "Scanner in = new Scanner(System.in);".

Kadang muncul beberapa alternatif (gambar bawah) karena nama kelas dapat saja sama. Pilihlah Scanner (java.util).

A screenshot of an IDE showing a code completion popup. The popup is titled "Class to Import". Below the title, there are two options: "Scanner (java.util)" and "Scanner (com.sun.java\_cup.internal.runtime)". The first option is selected.

Lihat bagian paling atas dari program, IDE otomatis menambahkan import yang diperlukan.

```
import java.util.Scanner;
```

#### Tugas 4

Modifikasi tugas menghitung luas dan keliling segi empat, tetapi panjang dan lebarnya diinput dari keyboard.

### Formating Output

Cobalah contoh berikut: (perhatikan penggunaan .0 dibelakang 10 dan 3 untuk menandakan angka ini adalah angka float)

```
public static void main(String[] args) {
    double x = 10.0 / 3.0;
    System.out.print(x);
}
```

Hasil dari program tersebut sulit dibaca, terutama untuk orang awam. Jika kita ingin mem-format keluaran, maka dapat digunakan method printf yang serupa dengan printf pada bahasa C. Berikut adalah contoh penggunaan printf (import java.util.Date):

```

public static void main(String[] args) {
    double x = 10.0 / 3.0;
    System.out.printf("%.2f \n",x); // 8 tempat, 2 angka belakang koma
    System.out.printf("Selamat pagi %s, umur anda %d dan tinggi badan
    anda %.2f \n","Budi",20,175.4);

    Date tgl = new Date(); //tgl hari ini
    System.out.printf("Tgl hari ini: %td - %tB - %ty \n",tgl,tgl,tgl);

    //daripada tgl ditulis tiga kali, lebih praktis dengan <
    System.out.printf("Tgl hari ini: %td - %tB - %ty \n",tgl);
}

```

Silahkan melihat dokumentasi method printf untuk rincian formatting yang dapat dilakukan.

## **Conditional Statement (IF-THEN-ELSE)**

Format dari conditional statement di Java adalah

```

if (kondisi) {
    // aksi untuk kondisi = true
} else {
    // aksi untuk kondisi = false
}

```

Contoh:

```

public static void main(String[] args) {
    int umur = 20;
    if (umur>17) {
        System.out.println("Sudah cukup umur");
    } else {
        System.out.println("Belum cukup umur");
    }
}

```

Format dari nested if adalah:

```

if (kondisi) {
    // aksi untuk kondisi=true
} else if (kondisi2) {
    // aksi untuk kondisi = false dan kondisi2 = true
} else if (kondisi3) {
    // aksi untuk semua false kecuali kondisi3
}

```



Contoh programnya adalah sebagai berikut (import java.util.Scanner)

```
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.println("nilai anda?");
    int nilai = in.nextInt();
    if (nilai<20) {
        System.out.println("C");
    } else if (nilai<50) {
        System.out.println("B");
    } else {
        System.out.println("A");
    }
}
```

### **Tugas 5**

Buatlah program yang menerima input data penghasilan per bulan dan outputnya adalah pajak yang harus dibayar. Jika penghasilan lebih kecil dari satu juta, maka tidak kena pajak. Jika penghasilan antara 1 sd <2 juta, pajaknya 5%. Jika penghasilan antara 2 sd <5 juta, pajaknya 10%. Diatas 5 juta pajaknya 20%.

### **LOOP (WHILE-DO)**

Loop while-do melakukan pengecekan kondisi loop diawal. Format loop while-do adalah:

```
while (kondisi) {
    //aksi selama kondisi = true
}
```

Contoh:

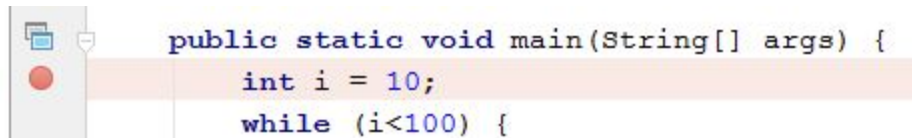
```
public static void main(String[] args) {
    int i = 10;
    while (i<100) {
        System.out.println(i);
        i = i + 20;
    }
    System.out.println("Selesai");
}
```

## CATATAN: DEBUGGING

Sebelum melanjutkan ke materi berikutnya, perlu dipelajari kemampuan yang penting dimiliki dalam dunia pemrograman yaitu debugging. Debugging adalah kegiatan menelusuri eksekusi program untuk mencari kesalahan atau error (bug).

Kita akan mempelajari cara men-debug program di IntelliJ.

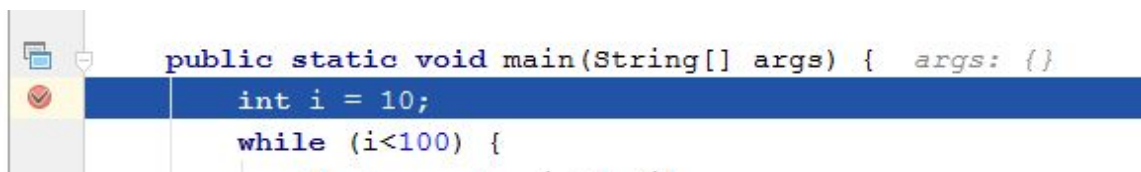
Pertama pada baris “int = 10” di program sebelumnya (while-do) click pada pita di kiri (gambar bawah) sampai titik merah muncul. Titik ini disebut breakpoint



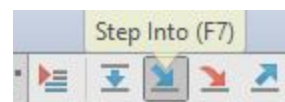
Jalankan debug melalui icon “serangga” atau Shift-F9. Program akan berhenti tepat di breakpoint.



atau Shift-F9. Program akan berhenti

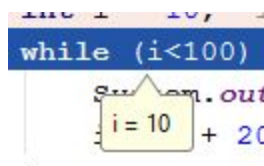


Untuk menjalankan baris demi baris, gunakan **F7** atau



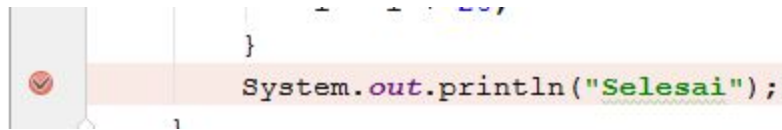
silahkan coba.

Fasilitas yang lain adalah melihat isi variabel. Setelah menekan F7, dan program berada di baris “while (i<100)”, arahkan mouse ke variabel i dan IDE akan menunjukkan isi dari variabel tersebut.

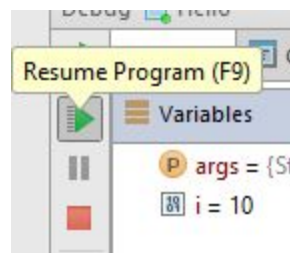


Jalankan F7 secara terus menerus, perhatikan pergerakan kursor dan nilai variabel i

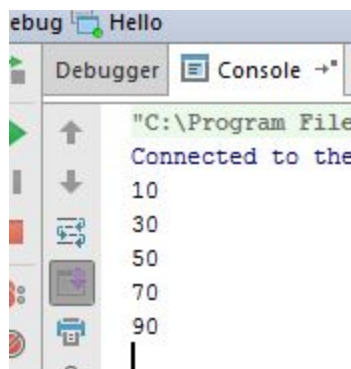
Untuk mempercepat (misalnya kita ingin loncat ke luar loop) dapat ditambahkan breakpoint diluar loop




lalu tekan F9 atau tombol play di kiri bawah (gambar bawah)



Ada dua tab di bagian bawah “Debugger” dan “Console”. Klik “console” untuk melihat hasil



Untuk menghentikan program di tengah-tengah eksekusi tekan ctrl-F2 atau  tapi jika program telah mencapai ujung, maka proses debugging akan berhenti secara otomatis.

## Tugas 6

Menggunakan while-do, buatlah program untuk mencari posisi huruf A pada sebuah kalimat.

Misalnya input “BUDI MARTAMI”, maka outputnya adalah: 7

Petunjuk:

- Gunakan method `String.length()` untuk mendapatkan panjang kalimat.
- Gunakan method `String.charAt(i)` untuk mendapatkan karakter pada posisi ke i (posisi dimulai dari 0)

Contoh penggunaan `length` dan `charAt`

```
public static void main(String[] args) {  
    String s = "halo dunia";  
    System.out.println("Panjang s="+s.length());  
    System.out.println("Karakter ke-0 =" +s.charAt(0));  
    System.out.println("Karakter ke-4 =" +s.charAt(3));  
}
```

## LOOP (DO-WHILE )

Do-While mirip dengan while-do hanya saja pengecekan dilakukan di bagian akhir sehingga bagian loop akan dieksekusi minimal satu kali.

```
do {  
    //aksi selama kondisi = true  
} while (kondisi);
```

Contoh:

```
public static void main(String[] args) {  
    int i = 10;  
    do {  
        System.out.println(i);  
        i = i + 20;  
    } while (i<100);  
    System.out.println("Selesai");  
}
```

Loop didalam Do-While minimal dieksekusi sebanyak satu kali, sedangkan loop didalam while-do minimal dieksekusi sebanyak 0 kali.

### **Tugas 7**

Menggunakan do-while, buatlah menjumlahkan. Selama pengguna belum mengisikan “T”, maka akan terus dilakukan penjumlahan.

Misalnya:

Angka? 10

Lagi? Y

Angka? 20

Lagi? Y

Angka? 30

Lagi? T

Total = 60

### **Jawaban**

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    int angka;  
    int total=0;  
    String lagi;  
    do {  
        System.out.println("Nilai:");  
        angka = sc.nextInt();  
        total = total + angka;  
        System.out.println("Menghitung lagi?");  
        lagi = sc.next();  
    } while (!lagi.equals("t"));  
    System.out.println("Total="+total);  
}
```

## Tugas 8

Menggunakan do-while, buatlah program tebak angka. Asumsikan sebuah angka, lalu buatlah user menebak angka tersebut.

Misalnya:

Tebakan? 10

Terlalu besar.

Tebakan? 6

Terlalu kecil

Tebakan? 8

Benar, anda menebak dalam 3 langkah

Petunjuk: untuk menghasilkan angka random, gunakan

```
int angkaDiTebak = (int) (Math.random()*10);
```

Kode diatas membuat bilangan random dari 0 sd 10

## LOOP (FOR)

Loop ini adalah loop yang paling sering digunakan. Gunakan loop ini jika jumlah loop sudah diketahui

Format loop ini adalah

```
for (inialisasi; kondisi ; increment/decrement) {  
  
}
```

Contoh:

```
public static void main(String[] args) {  
    for (int i=0;i<=10;i++) {  
        System.out.println("Nilai i="+i);  
    }  
    System.out.println("Hitung mundur");  
    for (int i=10;i>0;i--) {  
        System.out.println("Nilai i="+i);  
    }  
    System.out.println("Nested Loop");  
    for (int i=0;i<10;i++) {  
        for (int j=0;j<10;j++) {  
            System.out.println("Nilai i="+i+";nilai j="+j);  
        }  
    }  
}
```

Perhatikan deklarasi variabel `i`, dengan dideklarasikan di bagian `for`, maka variabel `i` hanya valid di dalam loop, setelah keluar dari loop, maka variabel `i` tidak dapat digunakan lagi.

### Tugas 9

Menggunakan loop `for`, buatlah segitiga seperti berikut:

```
1
123
12345
1234567
```

Petunjuk:

Program diatas terdiri atas tiga loop, loop pertama untuk menghasilkan 4 baris angka, loop kedua untuk mencetak spasi di depan angka dan loop ketiga untuk mencetak deretan angka.

### ***Keluar Paksa dari Loop: Break***

Dalam beberapa kondisi, terkadang kita membutuhkan mekanisme untuk keluar dari loop. Java menyediakan statement **`break`** untuk keluar dari loop (`for`, `while`). Catatan: gunakan debug untuk melihat dengan lebih baik.

Contoh:

```
public static void main(String[] args) {
    int tebakkan = 8; //bisa diganti dengan random
    int inputUser;
    boolean tertebak = false;
    Scanner sc = new Scanner(System.in);
    //hanya boleh menebak sampai 5 kali
    for (int i=0;i<5;i++) {
        System.out.println("Tebakan Anda?");
        inputUser = sc.nextInt();
        if (tebakkan == inputUser) {
            tertebak = true;
            break; //keluar paksa dari loop
        }
    }
    if (tertebak) {
        System.out.println("tertebak");
    } else {
        System.out.println("habis waktu");
    }
}
```

## Tugas 10

Contoh diatas dapat diimplementasikan dengan loop while-do.

```
while (( i < 5) && (!tertebak)) {  
    ...  
}
```

Mana yang lebih baik? mengapa?

## Melewati Loop: Continue

Continue membuat aliran loop meloncat kembali ke awal.

Contoh:

```
public static void main(String[] args) {  
    for (int i=0;i<5;i++) {  
        if (i==3) {  
            continue; //kembali ke atas (for), angka 3 tdk diprint  
        }  
        System.out.println("Nilai i:"+i);  
    }  
}
```

Contoh diatas dapat dibuat dengan if biasa

```
for (int i=0;i<5;i++) {  
    if (i!=3) {  
        System.out.println("Nilai i: "+i);  
    }  
}
```

Mana yang lebih baik? mengapa?



Pada program diatas, saat program mencapai continue, maka aliran program akan kembali ke bagian for. Gunakan debugging untuk melihat proses detilnya.

## **Array**

Array adalah kumpulan variabel bertipe sama yang dapat diakses melalui indeks. Elemen array diakses menggunakan simbol kurung siku []. Setelah dideklarasikan, array harus diinisialisasi terlebih dulu.

Contoh:

```
int[] tinggi;  
tinggi = new int[100]; //ada 100 elemen, mulai dari indeks 0 sd 99
```

atau dapat digabungkan:

```
int[] tinggi = new int[100];
```

Contoh (perhatikan penggunaan `tinggi.length` untuk mendapat ukuran array):

```
public static void main(String[] args) {  
    int[] tinggi; //deklarasi  
    tinggi = new int[3]; //inisiasi  
    tinggi[0] = 3;  
    tinggi[1] = 6;  
    tinggi[2] = 3;  
  
    for(int i=0;i<tinggi.length;i++) {  
        System.out.println("Tinggi indeks ke-"+i+" adalah "+tinggi[i]);  
    }  
}
```

Ada cara yang lebih ringkas untuk inisiasi, yaitu dengan menggunakan kurung kurawal

```
public static void main(String[] args) {  
    int[] tinggi; //deklarasi  
    tinggi = new int[] {3,6,3}; //lebih ringkas  
  
    for(int i=0;i<tinggi.length;i++) {  
        System.out.println("Tinggi indeks ke-"+i+" adalah "+tinggi[i]);  
    }  
}
```

## Tugas 12

Modifikasi program diatas untuk fasilitas pencarian. Contoh:

Angka yang akan dicari? 3

Angka tersebut ada di indeks: 0, 2

## Array: For-Each

Dimulai dari Java SE 5.0, programmer dapat melakukan loop untuk setiap elemen array dengan mudah. Bentuk dari for-each adalah sebagai berikut:

```
for (variable : array) {  
    ...  
}
```

Contoh array pada halaman sebelumnya dapat ditulis ulang sebagai berikut

```
public static void main(String[] args) {  
    int[] tinggi;  
    tinggi = new int[3];  
    tinggi[0] = 3;  
    tinggi[1] = 6;  
    tinggi[2] = 3;  
  
    for(int n:tinggi) {  
        System.out.println("Tinggi adalah:"+n);  
    }  
}
```

## Tugas 13

Modifikasi tugas 12, ganti loop-nya dengan for each

## Array: Sorting

Sorting pada array dapat dilakukan dengan menggunakan method Arrays.sort

```
import java.util.Arrays;
public class Main {
    public static void main(String[] args) {
        int[] tinggi;
        tinggi = new int[3];
        tinggi[0] = 5; //assignment
        tinggi[1] = 6;
        tinggi[2] = 3;
        Arrays.sort(tinggi);
        for(int n:tinggi) {
            System.out.println("Tinggi adalah"+n);
        }
    } //main
}
```

## Array: Dua Dimensi

Contoh deklarasi dan inisiasi array dua dimensi adalah sebagai berikut:

```
double[][] matriks = new double[1][1];
```

Contoh:

```
public static void main(String[] args) {
    double[][] matriks = new double[2][2]; // 2 x 2

    matriks[0][0] = 0;
    matriks[0][1] = 1;
    matriks[1][0] = 2;
    matriks[1][1] = 3;

    //cara biasa
    for(int i=0;i<matriks.length;i++) {
        for (int j=0;j<matriks[i].length;j++) {
            System.out.printf("Nilai [%d,%d] = %4.2f \n",i,j,matriks[i][j]);
        }
    }
}
```

```
//cara foreach
for(double[] baris:matriks) {
    for (double val:baris) {
        System.out.println("Nilai = "+val);
    }
}
}
```

Cara inisiasi yang lebih ringkas:

```
public static void main(String[] args) {
    double[][] matriks = new double[][]
    {
        {0,1},
        {2,3}
    };

    // .... sama dengan atas
}
```

### **Tugas 14**

Buatlah program penjumlahan dua matriks. Pertama user ditanya dimensi matriks yang diinginkan, kemudian user memasukan data kedua matriks. Program menampilkan matriks pertama, kedua dan hasil penjumlahan kedua matriks.

Topik Lanjut:

***Generic***

TBD

**Collection**

TbD