IMPLEMENTASI SISTEM INFORMASI

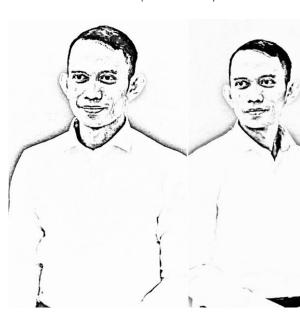
PRANATA KOMPUTER TINGKAT AHLI ANGKATAN II TAHUN 2019

FASILITATOR

وَمَا رَمَيْتَ إِذْ رَمَيْتَ وَلَٰكِنَّ اللَّهَ رَمَىٰ...

when you threw, but it was Allah who threw (QS:Al-Anfal :17)

Nama	Budi Subandriyo
NIP	19780720 2002121 007
Jabatan	Widyaiswara Ahli Madya
Pangkat	Pembina (IV/a)
Unit Kerja	Pusdiklat BPS





Tujuan Pembelajaran

Setelah mengikuti pembelajaran ini, peserta dapat menerapkan pemrogaman berbasis objek dan pemrograman berbasis web dengan baik dan benar

Tujuan Diklat

- Pelaksanaan tugas pokok dan fungsi BPS sebagai Instansi Pembina Jabatan Fungsional Pranata Komputer;
- Pelaksanaan tugas pokok dan fungsi Pusdiklat sebagai Penyelenggara Diklat Teknis dan Fungsional;
- Membentuk standar kompetensi yang perlu dimiliki oleh Pejabat Fungsional Pranata Komputer.

Sasaran Diklat

- Meningkatkan pengetahuan dan keterampilan yang diperlukan untuk melaksanakan tugas dalam Jabatan Fungsional Pranata Komputer secara profesional;
- Memenuhi salah satu persyaratan administrasi untuk pengangkatan dalam Jabatan Fungsional Pranata Komputer;
- Mampu mengikuti perkembangan teknologi informasi terkini;
- Mampu memberi saran-saran untuk meningkatkan pengelolaan Jabatan Fungsional Pranata Komputer.

Materi

- Sistem Penilaian dan Admistrasi Pranata Komputer
- Pengenalan Teknologi Informasi
- Rekayasa Sistem Informasi
- Analisis Sistem Informasi
- Perancangan Sistem Informasi
- Implementasi Sistem
- Administrasi & Pengelolaan Database
- Administrasi & Pengelolaan Jaringan Komputer
- Etika dan Pengembangan Profesi Pranata Komputer
- Dokumentasi dan Pelaporan



Pengenalan Java

Java Application Development 1



Why Java?

- Simple. Mengadopsi C/C++
- Secure. Java Applet memungkinkan executable program dieksekusi secara aman (komputer client dilindungi oleh JVM)
- Portable. Multiplatform.
- Object-Oriented. Metode pengembangan software terkini.
- Robust (powerfull). Pengecekan compilasi, penngecekan runtime, penanganan eksepsi.
- Multithreaded. Pemrosesan paralel.
- Architecture-Neutral. Write once; run anywhere, any time, forever.
- Interpreted and High Performance. Compiler terus mengalami penegmbangan.
- Distributed. Web service, RMI.
- Dynamic. Memungkinkan bagian program diload secara dinamis pada saat runtime.

Kompilasi



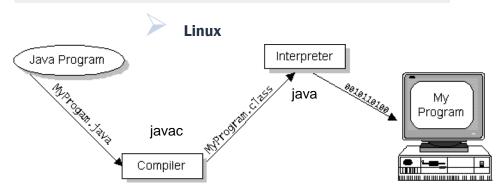
Setting Environment

Select Start -> Computer -> System Properties -> Advanced system settings -> Environment Variables -> System variables -> PATH.

[In Vista, select Start -> My Computer -> Properties -> Advanced -> Environment Variables -> System variables -> PATH.]

[In Windows XP, Select Start -> Control Panel -> System -> Advanced -> Environment Variables -> System variables -> PATH.]

please note 'PATH' is CASE sensitivity and must be in UPPERCASE
PATH=\$PATH:/path/to/dir1; export PATH



- File_name.java menunjukkan bahwa file tersebut berisi Java source code
- Compile: menggunakan command javac
- Setelah program di-compile, maka compiler akan menghasilkan *.class file yang berisi program terkompilasi
- Java compiler akan mengubah JSC ke bytecodes yang akan dieksekui oleh JVM

Contoh Program Sederhana

```
/*
   This is a simple Java program.
   Call this file "Example.java".
*/
class Example {
   // Your program begins with a call to main().
   public static void main(String args[]) {
      System.out.println("This is a simple Java program.");
   }
}
```

Komentar dan White Character

Komentar

- Mendokumentasikan program dan mempermudah pembacaan baris program
- Diawali dengan // (end of line) berakhir pada akhir baris tersebut
- Bisa dipecah menjadi beberapa baris (multiple line) / * */
- Diabaikan oleh compiler

White character

- Baris kosong, karakter spasi dan karakter tab
- Mempermudah pembacaan program
- Diabaikan oleh compiler

Class / Kelas



- Setiap program minimal terdiri dari satu class yang didefinisikan oleh programmer
- Ada beberapa keyword yang sudah dipakai oleh Java (reserved words)
- Class diikuti nama class merupakan deklarasi class
- Konvensi: Semua nama class diawali dengan huruf besar, contoh: NamaClass
- Nama class adalah identifier, terdiri dari huruf, angka, underscore, \$, tidak dimulai dengan angka, dan tidak mengandung spasi. Biasanya identifier yang tidak dimulai dengan huruf capital adalah bukan nama class.
- Nama class adalah case sensitive
- Setiap class diawali dan diakhiri dengan {}

Java's Reserved Words

abstract	assert	boolean	break	byte	case
catch	char	class	const	continue	default
do	double	else	enum	extends	final
finally	float	for	goto	iA	implements
import	instanceof	int	interface	long	native
new	package	private	protected	public	return
short	static	strictfp	super	switch	synchronized
this	throw	throws	transient	try	void
volatile	while				

Method / Metode / Prosedur / Fungsi

Method

- Method main adalah permulaan dari setiap aplikasi Java, dan harus dimulai dengan:
- public static void main(String args[])
- Dapat untuk melakukan task dan mengembalikan informasi
- Void mengindikasikan bahwa bahwa method melakukan tugas tetapi tidak mengembalikan informasi.

Statemet / Instruksi

Statement

- Memberi instruksi kepada komputer untuk melakukan "aksi"
- Gabungan karakter dalam " " adalah string
- System.out , menampilkan karakter ke command window
- System.out.println, menampilkan argumen ke command window diikuti oleh baris baru
- Harus diakhiri dengan semicolon (;)

Tipe Data, Variabel, dan Operator

Java Application Development 1

Туре	Contains	Default	Size	Range
boolean	true Of false	false	1 bit	NA
char	Unicode character	\u0000	16 bits	\u0000 to \uFFFF
byte	Signed integer	0	8 bits	-128 to 127
short	Signed integer	0	16 bits	-32768 to 32767
int	Signed integer	0	32 bits	-2147483648 to 2147483647
long	Signed integer	0	64 bits	-9223372036854775808 to 9223372036854775807
float	IEEE 754 floating point	0.0	32 bits	±1.4E-45 to ±3.4028235E+38
double	IEEE 754 floating point	0.0	64 bits	±4.9E-324 to ±1.7976931348623157E+308

- Boolean: bernilai true atau false.
- String: deretan karakter yang membentuk kata/kalimat.
- Dua buah String dapat disambung (concat) dengan operator '+'
- Karakter spesial:

Escape Sequence	Description
\ddd	Octal character (ddd)
\uxxxx	Hexadecimal UNICODE character (xxxx)
٧′	Single quote
\"	Double quote
11	Backslash
\r	Carriage return
\n	New line (also known as line feed)
\f	Form feed
\t	Tab
\b	Backspace

- Nama lokasi memory yang bisa diberi nilai.
- Setiap variabel harus dideklarasikan terlebih dahulu.
- Dideklarasikan sebagai:
 - type var-name;
 - type var=value;
- Type adalah tipe data, var-name adalah nama variabel.
- Nilai variabel dapat diubah pada saat eksekusi, sehingga nilainya tidak fix (tetap)

int a, b=8, c=19, d;

Contoh deklarasi: int count=10;

- Java memiliki fasilitas yang memungkinkan variabel untuk diinisialisasi secara dinamis, menggunakan sembarang ekspresi yang valid pada saat variabel tersebut dideklarasikan
- Contoh: program untuk menghitung volume silinder

- Biasanya variabel dideklarasikan pada awal main() method
- Java memperbolehkan variabel-variabel untuk dideklarasikan di dalam block {}, block inilah sebagai scope-nya
- Scope akan menentukan sampai sejauh mana variabel tersebut dikenali di dalam program
- Bagi Java, pengkategorian scope adalah yang didefiniskan oleh class dan method
- Variabel yang dideklarasikan di dalam scope tidak dapat diakses oleh bagian luar scope
- Dapat dibuat scope bersarang, yang berarti bahwa variabel scope terluar dapat dikenali oleh scope di dalamnya tapi tidak sebaliknya.

```
// Demonstrate block scope.
class ScopeDemo {
  public static void main(String args[]) {
          int x; // known to all code within
main
          x = 10;
          if (x == 10) { // start new scope
          int y = 20; // known only to this
block
          // x and y both known here.
          System.out.println("x and y: " + x
+ " " + y);
          x = y * 2;
   // y = 100; // Error! y not known here
   // x is still known here.
   System.out.println("x is " + x);
```

- Operator adalah simbol yang memberikan perintah kepada compiler untuk melakukan manipulasi matematika atau logika tertentu.
- Ada empat macam operator:
 - Arithmetic
 - **Bitmwise**
 - Relational
 - logical

- x = x+1; adalah sama dengan x++;
- x = x-1; adalah sama dengan x--
- Increment maupun decrement dapat berupa prefix atau postfix
- Contoh: ++x; //prefix
- x++;//postfix
- Jika operator mengikuti operand nya maka Java akan mengambil nilai operand sebelum proses inc/dec

```
x=10
```

y=++x; //y diset 11

$$x=10;$$

y=x++;//y diset 10

Hasil dari relasional dan logika operator adalah nilai boolean

- Semua objek dapat dibandingkan persamaan maupun pertidaksamaan dengan == dan !=
- <,>,<=,>= hanya dapat digunakan pada tipe data yang mendukung proses pengurutan
- Operator relasional dapat digunakan pada data tipe int dan char
- Tipe boolean hanya dapat dibandingkan dengan == atau !=
 - Untuk operator logika, operand nya harus dalam bentuk boolean, dan hasilnya dalam bentuk boolean

Operator	Meaning
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

The logical operators are shown next:

Operator	Meaning
&	AND
I	OR
۸	XOR (exclusive OR)
П	Short-circuit OR
&&	Short-circuit AND
!	NOT

highest			
()	[]		
++		~	į.
*	/	%	
+	-		
>>	>>>	<<	
>	>=	<	<=
==	!=		
&			
^			
I			
& &			
?:			
=	op=		
lowest			

Statement Kontrol dan Perulangan



Java Application Development 1



- Ada tiga tipe pernyataan seleksi dalam Java:
- Single selection statement (if statement)
- Hanya memilih atau mengabaikan satu aksi saja
- Double selection statement (if...else statement)
- Memilih diantara dua aksi yang berbeda
- Multiple selection statement (switch statement)
- Memilih diantara banyak aksi yang berbeda (group of action)

- Ada tiga tipe pernyataan seleksi dalam Java:
- Single selection statement (if statement)
- Hanya memilih atau mengabaikan satu aksi saja
- Double selection statement (if...else statement)
- Memilih diantara dua aksi yang berbeda
- Multiple selection statement (switch statement)
- Memilih diantara banyak aksi yang berbeda (group of action)







- Bahasa
- Pemrograman
- Java

- Java adalah bahasa pemrograman yang diperkenalkan oleh James Gosling dengan proyek bernama Oak
- Kemudian diformalkan oleh Sun Microsystem pada tahun 1995 dengan nama Java
- Java dikenal dapat dijalankan pada berbagai macam platform hardware komputasi, seperti computer, IoT, dan barang dengan mikrokontroler
- Memiliki prinsip "Write Once, Run Everywhere" karena dapat dijalankan pada beragam sistem operasi.

- Bahasa
- Pemrograman
- Java

- Java di setiap versinya memiliki 2 komponen utama untuk dapat dijalankan dalam sebuat system:
 - Java Application Programming Interface
 (API) = library pada Java yang dapat
 digunakan untuk kemudahan membuat
 program karena dengan Java API tidak
 perlu membuat program dasar dari awal
 (scratch)
 - Java Runtime Environment (JRE) = lingkungan yang memungkinkan aplikasi Java untuk dijalankan, komponen pentingnya adalah Java Virtual Machine
- Java Virtual Machine (JVM) adalah mesin virtual berupa aplikasi yang digunakan untuk menjalankan kode Java (bytecode).

- Bahasa
- Pemrograman
- Java

- Keunggulan Java dibandingkan Bahasa pemrograman lain adalah :
 - Relatif mudah dipelajari = Java adalah bahasa pemrograman berorientasi objek (OOP) murni yang lebih mudah dipelajari jika dibandingkan dengan C, C++ atau C#.
 - Peningkatan dari C++ dalam hal kemudahan penggunaan = dihilangkannya pointer pada tipe data biasa, mengimplementasikan multiple inheritance serta adanya garbage collection untuk kemudahan manajemen memory.
 - Mudah dikembangkan = karena sifat OOP-nya pengembangan aplikasi dengan Java dapat dipecah menjadi beberapa class, terdapat Java API dan dukungan framework, add-on, module dan API lain dari pihak ketiga.
 - Platform Independent = apapun platform yang memiliki JVM, maka program Java apapun dapat dijalankan pada platform tersebut.

- Bahasa
- Pemrograman
- Java

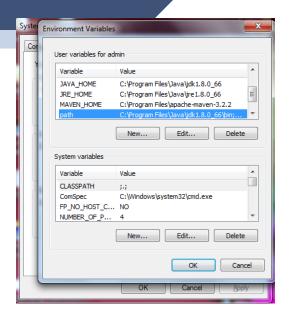
- Pemrogaman dengan Java memiliki 3 tahap :
 - Menuliskan kode program Java =
 programmer menuliskan kode program
 Java dengan text editor (Notepad,
 Notepad++), code editor (VSCode, Atom)
 atau Integrated Development
 Environment (IntellijIDEA, Netbeans,
 Eclipse) dan menyimpannya dengan
 ekstensi .java
 - Kompilasi progam Java = file .java yang telah dibuat diterjemahkan menjadi file binary dan menghasilkan file .class.
 - Menjalankan program dengan command "java nama_program"

- Bahasa
- Pemrograman
- Java

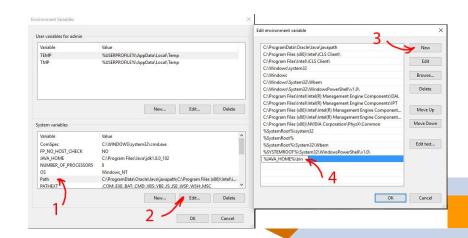
- Untuk mengembangkan program dengan Java, dibutuhkan beberapa tools sebagai berikut :
 - Java Development Kit (JDK)
 - Text editor/code editor/IDE
- JDK atau awalnya Java Software Development Kit (Java SDK) adalah seperangkat aplikasi yang digunakan untuk mengembangkan program dengan Bahasa Java.
- JDK memiliki 3 komponen yaitu :
 - Java API
 - Java Virtual Machine
 - Java Compiler

- Bahasa
- Pemrograman
- Java

- Instalasi JDK dapat dilakukan dengan mengunduh paket JDK yang disediakan oleh beberapa vendor misalnya Oracle JDK, RedHat JDK atau OpenJDK.
- Versi terbaru adalah Oracle JDK 12 yang berisi Java versi 12
- Ketika paket JDK telah terunduh, install JDK dan dilanjutkan dengan setting path (pada OS Windows) agar program Java dapat dikompilasi dan dijalankan disemua folder. Tanpa dilakukan setting path ini maka program Java hanya dapat decompile dan di jalankan di folder tempat JDK terinstall.



- Langkah konfigasi path adalah :
 - Cari folder instalasi JDK, biasanya ada di C:\Program Files\Java\jdkxxxxx\bin, copy alamat folder tersebut
 - Masuk ke Control Panel -> System ->
 Advance System Setting ->
 Environment Variable -> Path dan
 tekan edit. Sisipkan alamat folder bin
 JDK tersebut ke path.



Terima Kasih