

**Konsep Dasar  
Pemrograman  
Berorientasi Obyek I**

**JAVA**

Anugrah Kusuma  
Seno Adi Putra

***Konsep Dasar Pemrograman Berorientasi Obyek I  
JAVA***

@2007 Java Competency Center - Institut Teknologi Bandung

**Penulis :**

Anugrah Kusuma, MT

Seno Adi Putra, MT

**Penanggung Jawab :**

Dr. Bambang Riyanto Trilaksono

**Tim Editor :**

Java Competency Center - Institut Teknologi Bandung

Labtek VIII - Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Jalan Ganesha No. 10 Bandung 40132

web. [www.jcc-itb.com](http://www.jcc-itb.com)

e-mail. [info@jcc-itb.com](mailto:info@jcc-itb.com)

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah Yang Maha Esa karena atas berkat rahmat dan karunia-Nya penulis dapat menyelesaikan buku ini.

Buku ini membahas tentang konsep dasar pemrograman berorientasi obyek berbasis teknologi Java. Pemrograman berorientasi obyek merupakan konsep pemrograman yang relatif baru, di mana pemrograman diarahkan ke paradigma pembentukan obyek-obyek yang saling berinteraksi. Selain konsepnya lebih mudah dicerna oleh pemrogram, baik yang awam sekalipun, konsep pemrograman berorientasi obyek mempermudah *maintenance* software sehingga software menjadi lebih fleksibel apabila akan direvisi atau dikembangkan. Java merupakan bahasa pemrograman yang murni berbasiskan pemrograman berorientasi obyek menawarkan fitur menarik bagi para programmer yang ingin memperdalam konsep pemrograman berorientasi obyek.

Buku ini ditujukan bagi pembaca yang belum memahami konsep pemrograman berorientasi obyek atau belum memahami bahasa pemrograman apa pun. Buku ini juga cocok bagi pembaca yang ingin memperdalam lebih lanjut tentang konsep pemrograman berorientasi obyek dan teknologi Java.

Penulis sadari bahwa buku ini masih jauh dari sempurna. Untuk itu segala saran dan kritik yang membangun dalam penyusunan buku berikutnya sangat penulis harapkan. Namun demikian, mudah-mudahan buku ini bermanfaat bagi semua pihak.

Penulis  
Anugrah Kusuma  
Seno Adi Putra



# DAFTAR ISI

<b>KATA PENGANTAR</b>	i
<b>DAFTAR ISI</b>	ii
<b>PEMROGRAMAN DAN SIKLUS HIDUP PERANGKAT LUNAK</b>	
I.1 Pendahuluan	1
I.2 Pemrograman	1
I.2.1 Pengertian	1
I.2.2 Bahasa Pemrograman	2
I.2.3 Level Bahasa Pemrograman	3
I.3 Siklus Hidup Perangkat Lunak	3
I.3.1 Analisis ( Analysis )	4
I.3.2 Desain ( Design )	5
I.3.3 Pengembangan (Development)	5
I.3.4 Pengujian ( Testing )	6
I.3.5 Implementasi ( Implementation )	7
I.3.6 Pemeliharaan ( Maintenance )	8
I.3.7 Akhir Siklus ( End-Of-Life )	8
<b>ANALISIS MASALAH MENGGUNAKAN KONSEP OBJECT-ORIENTED ANALYSIS</b>	
II.1 Pendahuluan	9
II.2 Contoh Masalah	9
II.2 Identifikasi Domain Masalah	10
II.3 Identifikasi Obyek	11
II.4 Seleksi Obyek	12
II.5 Solusi	15
<b>MEMBUAT DAN MENGUJI PROGRAM JAVA</b>	
III.1. Identifikasi Komponen-komponen dari Sebuah Class	17
III.1.1 Strukturisasi Class	17

III.1.2 Deklarasi Class	17
III.1.3 Deklarasi Variabel dan Penugasan	18
III.1.4 Pendefinisian Method	19
III.1.5 Pemberian Komentar	20
III.2. Membuat dan Menguji Program Java	20
III.2.1 Konfigurasi yang Dibutuhkan	20
III.2.2 Membuat dan Menggunakan Class	27
III.3. Mengkompilasi dan Mengeksekusi Program	29

### **DEKLARASI, INISIALISASI, DAN PENGGUNAAN VARIABEL**

IV.1 Identifikasi Penggunaan Variabel dan Syntax	31
IV.1.1 Penggunaan Variabel	32
IV.1.2 Deklarasi dan Inisialisasi Variabel	33
IV.2 Mendeskripsikan Tipe Data Primitif	36
IV.2.1 Tipe Data Integral	37
IV.2.2 Tipe Data Floating Point	38
IV.2.3 Tipe Data Textual	39
IV.2.4 Tipe Data Logika	39
IV.2.5 Memilih Tipe Data	39
IV.3 Mendeklarasikan Variabel dan Melewatkan Nilai ke Variabel	40
IV.3.1 Penamaan Sebuah Variabel	40
IV.3.2 Melewatkan Sebuah Nilai ke Sebuah Variabel	41
IV.3.3 Konstanta	42
IV.3.4 Menyimpan Variabel Primitif dan Konstanta pada Memori	42

### **MENGGUNAKAN OPERATOR ARITMATIKA DAN BITWISE**

V.1 Menggunakan Operator Aritmatika untuk Memodifikasi Nilai	43
V.2 Menggunakan Operator Bitwise untuk Memodifikasi Nilai	45
V.3 Prioritas Operator	53
V.4 Menggunakan Promosi dan <i>Type Casting</i>	54
V.4.1 Promosi	54

V.4.2	Type Casting	54
V.4.3	Beberapa Catatan pada Promosi dan <i>Type Casting</i>	57
V.4.3.1	Operasi Aritmatika Menghasilkan Nilai di Luar Batas Tipe Data	57
V.4.3.2	Asumsi Dasar Compiler	58

## **MEMBUAT DAN MENGGUNAKAN OBYEK**

VI.1	Mendeklarasikan Referensi Obyek, Instantiasi Obyek, dan Inisialisasi Referensi Obyek	63
VI.1.1	Mendeklarasikan Variabel Referensi Obyek	64
VI.1.2	Inisialisasi Variabel Referensi Obyek	65
VI.1.3	Menggunakan Variabel Referensi Obyek untuk Memanipulasi Data	67
VI.1.4	Memindahkan Sebuah Referensi dari Satu Obyek ke Obyek yang Lain	69

## **MENGGUNAKAN CLASS STRING DAN PUSTAKA JAVA**

VII.1	Menggunakan Class <i>String</i>	71
VII.1.1	Menginstantiasi Obyek <i>String</i> dengan Kata Kunci <i>new</i>	72
VII.1.2	Menginstantiasi Obyek <i>String</i> tanpa Kata Kunci <i>new</i>	73
VII.1.3	Penggunaan Operator ' <i>==</i> ' dan Method <i>equals ( )</i> untuk Membandingkan Dua Buah <i>String</i>	74
VII.1.4	Menggunakan Variabel Referensi untuk Obyek <i>String</i>	74
VII.2	Investigasi Pustaka Class Java	75
VII.2.1	Spesifikasi Pustaka Class-class Java	75
VII.2.2	Menggunakan Spesifikasi Pustaka Class-class Java untuk Mempelajari Method	75

## **MENGGUNAKAN OPERATOR DAN KONSTRUKSI PENGAMBILAN KEPUTUSAN**

VIII.1 Menggunakan Operator Relasional dan Kondisional	77
VIII.1.1 Operator Relasional	77
VIII.1.2 Operator Kondisional	78
VIII.2 Konstruksi Pengambilan Keputusan	79
VIII.2.1 Konstruksi <i>if</i>	79
VIII.2.2 Konstruksi <i>if / else</i>	81
VIII.2.3 Konstruksi <i>Switch</i>	85

## **MENGGUNAKAN KONSTRUKSI LOOP**

IX.1 Konstruksi Loop	89
IX.2 Membuat Loop Menggunakan <i>While</i>	89
IX.3 Membuat Loop Menggunakan <i>For</i>	92
IX.4 Membuat Loop Menggunakan <i>Do/While</i>	95
IX.5 Loop Bersarang ( <i>Nested Loop</i> )	96
IX.6 Perbandingan Konstruksi Loop	97
IX.7 Pernyataan <i>continue</i>	97

## **MENGGUNAKAN METHOD**

X.1 Pengertian <i>Method</i>	99
X.2 Membuat dan Memanggil ( <i>Invoke</i> ) Method	99
X.2.1 Bentuk Dasar Method	100
X.2.2 Memanggil Method dari Class yang Berbeda	104
X.2.3 Memanggil Method dari Class yang Sama	105
X.3 Melewatkan Argumen dan Mengembalikan Nilai	106
X.3.1 Mendeklarasikan Method ber-Argumen	108
X.3.2 Memanggil Method ber-Argumen	109
X.3.3 Mendeklarasikan Method yang Memiliki Nilai Pengembalian	109
X.3.4 Menerima Nilai Pengembalian	110



X.3.5 Keuntungan Menggunakan Method	111
X.4 Menggunakan Overloading pada Method	113
X.4.1 Overloading Method pada Java API	116

## **VARIABEL DAN METHOD STATIK**

XI.1 Membuat Method dan Variabel <i>static</i>	117
XI.1.1 Mendeklarasikan Method <i>static</i>	119
XI.1.2 Memanggil Method <i>static</i>	120
XI.1.3 Mendeklarasikan Variabel <i>static</i>	120
XI.1.4 Method <i>static</i> dan Variabel <i>static</i> pada Java API	121
XI.1.5 Method <i>main ( )</i>	122
XI.1.6 Kapan Menggunakan Method atau Variabel <i>static</i> ?	122

## **ENKAPSULASI DAN KONSTRUKTOR**

XII.1 Menggunakan Enkapsulasi	123
XII.1.1 Visibility Modifier	123
XII.1.2 Modifier <i>public</i>	123
XII.1.3 Potensi Masalah dengan Atribut <i>public</i>	123
XII.1.4 Modifier <i>private</i>	124
XII.1.5 Interface dan Impementasinya	126
XII.2 Mendeskripsikan <i>Variable Scope</i>	130
XII.2.1 Penempatan <i>Instance Variable</i> dan <i>Local Variable</i> pada Memori	131
XII.3. Kontruktor	132
XII.3.1 Mendefinisikan Kontruktor	132
XII.3.2 Kontruktor Default	134
XII.3.3 Kontruktor Overloading	134