

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/264422101>

Buku Pertama Belajar Pemrograman Java untuk Pemula

Book · January 2014

DOI: 10.13140/2.1.3686.2084

CITATIONS

6

READS

299,058

1 author:



[Abdul Kadir](#)

56 PUBLICATIONS 1,147 CITATIONS

SEE PROFILE

Abdul Kadir



100%
Mudah
Dipelajari

A red thumbs-up icon is positioned to the right of the text "100% Mudah Dipelajari".

Buku Pertama Belajar
Pemrograman
Java
untuk Pemula

Plus CD!!!

Berisi software Java SE7 dan
Eclipse serta contoh-contoh yang
digunakan di buku ini



BUKU PERTAMA BELAJAR PEMROGRAMAN JAVA UNTUK PEMULA

Oleh: Abdul Kadir

© all rights reserved

Hak Cipta Dilindungi Undang-Undang

Penyunting: Ucak

Desain Sampul: Fellauji Duri

Tata Letak: @LeoMangW

Diterbitkan Oleh:

Mediakom

Jl. Cempaka Putih No. 8 Deresan

CT X, Gejayan, Yogyakarta 55283

Telp. (0274) 556043/555939,

Faks. (0274) 546020

Email: penerbitmediakom@gmail.com

Website: www.mediakom-penerbit.com

Katalog Dalam Terbitan (KDT)

Abdul Kadir

Buku Pertama Belajar Pemrograman

Java untuk Pemula/Ucak—Cet. 1

Yogyakarta: Penerbit MediaKom, 2014,
444 hlm; 18 x 23 cm

ISBN (10) 979-877-368-3

ISBN (13) 987-979-877-368-6

I. Judul

II. Komputer 600

Distributor Tunggal:

PT. BUKU SERU

Jl. Kelapa Hijau No. 22 RT 006/03

Kelurahan Jagakarsa, Kecamatan

Jagakarsa, Jakarta 12620

Telp. (021) 7888-1850

Faks. (021) 7888-1860

e-mail: marketingbukuseru@gmail.com

Website: www.bukuseru.com

Cetakan Pertama, 2014

Undang-Undang RI Nomor 19 Tahun 2002 tentang Hak Cipta

Ketentuan Pidana

Pasal 72:

2. Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran Hak Cipta atau Hak Terkait sebagaimana dimaksud pada ayat (1), dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).



PENGANTAR

Salah bahasa pemrograman yang populer saat ini yang digunakan untuk membuat aplikasi adalah Java. Salah satu penyebabnya adalah karena Java dapat dijalankan di berbagai *platform*. Java pun menjadi lebih populer semenjak kehadiran ponsel cerdas yang berbasis Android, karena Java-lah yang menjadi tulang punggung untuk membuat aplikasi-aplikasi di piranti tersebut.

Buku ini memberikan landasan bagi siapa saja yang ingin mempelajari bahasa pemrograman Java. Materi disusun sedemikian rupa sehingga bagi yang belum pernah memprogram pun tidak akan mengalami kesulitan untuk mempraktikannya.

Untuk mempermudah dalam mempraktikkan materi, *software* yang terkait disediakan di CD yang disertakan bersama buku ini. Selain itu, contoh-contoh yang digunakan di buku ini juga disertakan di CD. Namun, contoh tersebut sebaiknya hanya digunakan ketika yang Anda coba tidak sesuai dengan harapan. Dengan perkataan lain, contoh dalam bentuk digital hanya digunakan sebagai pembanding. Mencoba untuk mengetik sendiri untuk setiap kode tetap dianjurkan agar Anda bisa menghayati proses untuk menyusun program, termasuk dalam mengatasi kesalahan-kesalahan. Akhirnya, selamat berselancar untuk menjelajahi dunia pemrograman, khususnya dengan Java. Jumlah “jam terbang” Anda akan menentukan seberapa cepat Anda akan menguasainya.

Yogyakarta, Juli 2013

Salam hangat,

Abdul Kadir

DAFTAR ISI

PENGANTAR 3

DAFTAR ISI 4

BAGIAN 1 PENGENALAN JAVA

Apa sebenarnya Java itu?	15
Apa bedanya program dan aplikasi?	16
Saya awam dengan pemrograman. Apakah saya bakal bisa membuat program dengan Java?	16
Apa betul untuk menguasai pemrograman harus belajar algoritma terlebih dulu?	16
Apa hal dasar yang harus saya siapkan untuk memulai pemrograman Java?	19
Bagaimana cara menginstal Java?	19
Bisakah saya dibantu untuk melakukan pemasangan Eclipse di komputer saya?	22
Saya melihat keberadaan folder Eclipse di komputer saya.	
Sekarang apa yang harus saya lakukan?	22
Lalu, Bagaimana cara menuliskan program?	23
Saya telah menuliskan Kode. Bagaimana cara memperoleh hasil program tersebut?	25
Bisakah dijelaskan mengenai kode yang baru saja saya praktikkan?	25
Kapan sebaiknya komentar diberikan?	27
Apa yang terjadi kalau sebuah pernyataan tidak diakhiri dengan titik koma?	28
Apakah sebuah pernyataan bisa ditulis lebih dari satu baris?	28
Apakah sebuah baris dapat dipakai untuk lebih dari sebuah pernyataan?	28
Sebenarnya, bagaimana mekanisme di dalam pembuatan program?	28
Bagaimana menghindari kesalahan di dalam penulisan program?	29
Apakah letak kesalahan karena kesalahan sintaks ditunjukkan oleh Eclipse?	31
Mengapa jendela Console tidak terlihat?	32
Secara tidak sengaja saya menutup kode sumber? Bagaimana membukanya kembali?	32
Bagaimana cara mengakhiri Eclipse?	33
Bagaimana cara menjalankan program Java tanpa melalui Eclipse?	33
Pada contoh yang diberikan, tanda { diletakkan satu baris dengan main.	
Namun, saya seringkali melihat tanda { diletakkan di bawah main (pada baris yang berbeda). Apa perbedaan kedua penulisan tersebut?	35
Saya sering mendengar bahwa bahasa Java adalah bahasa pemrograman berorientasi objek. Apa maksudnya?	35

BAGIAN 2 VARIABEL, KONSTANTA, DAN LITERAL	37
Saya sering mendengar istilah variabel. Apa sebenarnya variabel itu?	37
Dapatkah diberikan contoh variabel dalam bentuk program?	38
Tipe data apa saja yang disediakan oleh Java?	40
Untuk bilangan bulat, Apakah Java hanya menyediakan tipe data int?	40
Apa yang terjadi kalau misalnya saya memaksakan untuk memasukkan nilai 32768 ke tipe short?	41
Apakah Java tidak mendukung unsigned seperti pada C++?	42
Kalau saya mau memproses bilangan real, tipe data apa yang perlu saya gunakan?	42
Apa kegunaan tipe data boolean? Bagaimana gambarannya?	44
Tipe karakter itu hanya untuk menyimpan sebuah karakter?	45
Bagaimana membuat variabel yang bertipe string?	46
Deklarasi beberapa variabel bisa diletakkan di dalam sebuah pernyataan?	47
Apa deklarasi variabel dapat sekaligus digunakan untuk memberikan nilai kepada variabel?	47
Apakah ada aturan untuk memberikan nama variabel?	58
Apakah yang dimaksud kata kunci?	49
Apakah konstanta itu?	49
Kalau tulisan seperti 56 di dalam kode sumber dinamakan apa?	51
Bagaimana aturan penulisan literal bilangan?	51
Saya pernah melihat kode seperti 0xffff tanpa petik. Apa maksudnya?	52
Apa penulisan konstanta 030 dan 30 itu memang berbeda?	53
Bagaimana halnya kalau mau menyatakan literal bilangan dengan bentuk biner?	54
Literal seperti <code>\"</code> tergolong sebagai apa?	55
BAGIAN 3 EKSPRESI, OPERATOR, & OPERAND	57
Apa yang dimaksud dengan ekspresi?	57
Lalu, operator atau <i>operand</i> itu apa?	58
Apakah antara <i>operand</i> dan operator harus diberi sebuah spasi?	59
Apa maksud operator <i>unary</i> dan <i>binary</i> ?	59
Operator apa saja yang disediakan Java untuk kepentingan penghitungan bilangan?	59
Apa operator untuk menangani perpangkatan tidak ada?	61
Bagaimana kalau misalnya saya ingin menghitung 3^{10} .	61
Apa harus menulis a sebanyak 10 kali? Apa itu praktis?	61
Bagaimana jika saya ingin menghitung akar 5?	62
Saya menuliskan ekspresi seperti berikut: $2 + 3 * 5$. Hasilnya kenapa bukan 25?	63
Bagaimana caranya agar $2 + 3$ pada contoh di depan dikerjakan dulu dan kemudian baru dikalikan dengan 5?	64

Saya sering melihat ekspresi seperti <code>x++</code> atau bahkan <code>++x</code> . Apa maksudnya?	65
Saya pernah menjumpai ekspresi <code>x = x + 1</code> pada Java.	
Secara logika, ekspresi itu apa tidak salah?	66
Apa perbedaan antara <code>x = x + 1</code> dengan <code>x++</code> ?	67
Apa maksud bentuk ekspresi seperti <code>x + = 2</code> ?	67
Apakah suatu <i>operand</i> yang berbeda tipe dapat dikenakan dalam suatu operator <i>binary</i> ?	67
<i>Typecasting</i> itu apa?	69
Apakah Java tidak mendukung operasi seperti logaritma dan sinus?	69
Bisakah diberi contoh berbagai notasi matematika dan perwujudannya dalam ekspresi Java?	70

BAGIAN 4 INPUT/OUTPUT 73

Bagaimana caranya kalau saya ingin memasukkan data dari <i>keyboard</i> ketika program dijalankan?	73
Pada program di depan, saya memasukkan data seperti berikut: 30 15 diikuti dengan Enter. Kok program tidak lagi meminta data lebar?	76
Saya membuat program yang membaca data <i>int</i> dan kemudian data <i>string</i> . Kok selalu ada masalah. Kenapa ya?	77
Kapan <i>import</i> diperlukan?	78
Contoh di depan digunakan untuk membaca data bertipe <i>double</i> . Bagaimana kalau saya ingin membaca data bertipe <i>string</i> ?	78
Misalnya, saya ingin membaca data bertipe <i>int</i> , tetapi pemakai memasukkan <i>string</i> . Apakah kesalahan seperti itu bisa dideteksi oleh program?	79
Apa benar <i>Scanner</i> memiliki fasilitas pemformat yang berguna untuk menampilkan keluaran?	81

BAGIAN 5 KEPUTUSAN DENGAN IF DAN SWITCH 85

Bisakah diberikan gambaran tentang manfaat keputusan dengan <i>if</i> ?	85
Bisakah dijelaskan mengenai penggunaan <i>if</i> di dalam program?	86
Lalu, bentuk <i>if</i> yang tidak sederhana itu seperti apa?	88
Apakah kondisi dalam <i>if</i> harus ditulis dalam tanda kurung. Kalau tidak, apa yang terjadi?	89
Dapatkah pernyataan yang dijalankan oleh <i>if</i> lebih dari satu pernyataan?	90
Beberapa contoh <i>if</i> yang telah dibahas menggunakan operator seperti <code><</code> dan <code>==</code> . Apa sebutan untuk operator seperti itu?	93
Apakah kondisi seperti <code>a == b</code> boleh disederhanakan menjadi <code>a = b</code> ?	94
Jadi, kondisi dalam <i>if</i> tidak harus selalu melibatkan operator relasional?	94
Apakah operator <code>==</code> berlaku untuk <i>string</i> ?	95

BAB 7 PENULISAN SALAH PROGRAMIRAN JIKA BUTUH PERUBA	
Bagaimana caranya kalau saya ingin menggunakan operasi "dan" pada suatu kondisi di dalam if?	96
Apakah kondisi & memang harus ditulis dua kali?	97
Apakah operasi "atau" juga berbeda?	97
Apakah kondisi & juga harus ditulis dua kali?	98
Dalam beberapa bahasa pemrograman terdapat operasi "NOT". Apa padamnya pada java?	98
Apakah kondisi &¬ juga identik dengan (a==b && not c)?	99
Saya pernah mendengar istilah "short-circuit" pada operasi "dan" dan "atau". Apa maksudnya?	100
Apakah pernyataan if bisa diletakkan di dalam pernyataan if?	100
Operator ? : itu identik dengan if?	102
Apakah label pernyataan switch bisa menggunakan if?	103
Apakah break memang harus selalu diletakkan di bagian akhir setiap case dalam pernyataan switch?	105
Apakah pernyataan switch harus mempunyai default?	107
Apakah switch bisa menggunakan literal string?	107
BAGIAN 6 PENGULANGAN DENGAN WHILE DAN DO-WHILE	109
Apakah yang dimaksud dengan pengulangan?	109
Bagaimana bentuk pengulangan dengan while?	109
Bisakah diberikan gambaran penggunaan while?	110
Bagaimana kalau saya ingin menampilkan bilangan seperti pada contoh di depan, yaitu: 1, 2, 3, ..., 10?	111
Saya ingin menampilkan bilangan 1 sampai dengan N, tetapi khusus untuk bilangan yang habis dibagi dengan 5 tidak perlu ditampilkan. Bagaimana caranya?	116
Bagaimana kalau saya menggunakan keluaran $\sum_{i=1}^n i$?	121
Apakah while bisa digunakan untuk menghitung $\sum_{i=1}^n i$?	122
Bagaimana menghitung deret $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \frac{1}{7} - \frac{1}{8} + \frac{1}{9} - \frac{1}{10}$?	124
Saya ingin menghitung n! (n faktorial). Bagaimana caranya?	125
Kadang saya mendapati pernyataan continue di dalam while. Apa fungsinya?	127
Bagaimana bentuk pengulangan dengan do-while?	129

BAB 8 PENULISAN SALAH PROGRAMIRAN JIKA BUTUH PERUBA	
BAGIAN 7 PENGULANGAN DENGAN FOR	131
Bagaimana bentuk pengulangan dengan for?	131
Bentuk for lebih sederhana daripada while, ya?	133
Bagaimana caranya kalau saya ingin menampilkan bilangan dari 10 sampai ke 27?	133
Jadi, bagian penulisan bisa dibuat naik atau turun dengan selang tidak harus bernilai satu?	134
Di dalam bagian inisialisasi for kadang terdapat kode semacam berikut: int i = 0. Apa maksud int di situ?	135
Bagian penulisan pada for bisa melibatkan lebih dari satu variabel?	135
Saya ingin membuat tabel yang mencantumkan nilai suhu dalam derajat Celsius dan Fahrenheit dimulai dari 0°C hingga 100°C dengan selang sebesar 10°C. Bagaimana kodenya?	137
Apakah for bisa dipakai untuk menambah kuantitas seperti "taman" menjadi "taman2"?	138
Bagaimana cara menghitung $\sum_{i=1}^n i$ menggunakan for?	139
Saya ingin menghitung n! (n faktorial) dengan for. Bagaimana caranya?	139
Sebuah bilangan dianggap prima jika, jumlah tanda bintang diberikan oleh nilai yang dimasukkan dari keyboard. Bagaimana implementasinya kalau menggunakan for?	140
Bagaimana kalau ada persoalan seperti berikut?	142
Dalam hal ini, jumlah hari diberikan melalui keyboard sewaktu program dijalankan? Bagaimana kalau bentuk segitrimya seperti berikut?	144
Bagaimana caranya membuat kotak semacam berikut, dengan tinggi dan lebar kotak bisa ditentukan sewaktu program dijalankan?	146
Menurut saya, persoalan membuat kotak bisa dipecahkan tanpa menggunakan for. Bagaimana, Benek, kamu?	147
Bagaimana cara membuat deret bilangan Fibonacci seperti berikut: 1, 1, 2, 3, 5, 8, 13, 21, ...? Apakah bisa dan continue yang diterapkan pada while juga berlaku untuk for?	150
BAGIAN 8 PEMBUATAN METODE	153
Bisakah dijelaskan dahulu mengenai metode dan bagian metode diperlukan?	153
Nilai yang diletakkan dalam tanda kurung seperti argumen? Bisa dibuat apa?	154
Apakah setiap metode pasti memiliki return?	154
Apakah perbedaan rekursif dengan return?	155
Dapatkah diberikan contoh pembuatan metode?	155
Apakah metode boleh tidak mengimpor argumen?	156
Metode seperti shell pada Math dapat melibatkan argumen bertipe double atau int. Hal yang menarik, bagaimana tipe nilai bulatnya bisa memasukkan dengan tipe argumen?	157
Saya pernah melihat sebuah metode yang sama kadang dipanggil dengan sebuah argumen. Apa maksudnya?	158

BAB 9 PENULISAN SALAH PROGRAMIRAN JIKA BUTUH PERUBA	
Variabel yang dideklarasikan di dalam suatu metode apa bisa diisali oleh pemanggilnya?	159
Apakah dimungkinkan untuk membuat variabel global?	161
Apakah semesta variabel pada deklarasi seperti static string harusnya adalah ada public static void main pada contoh di depan?	162
Apakah metode bisa memanggil dirinya sendiri?	162
Adakah contoh lain penggunaan metode rekursif yang sangat sulit kalau tidak menggunakan metode rekursif?	163
Bisakah argumen yang digunakan saat pemanggilan metode diubah oleh metode?	166
Suatu variabel di kelas kadang dipanggil secara langsung tanpa kelas. Bagaimana aturannya?	168
Di dalam suatu metode sering kadang diawali dengan kelas. Bagaimana aturannya?	168
Bagaimana caranya kalau saya ingin membuat kelas yang terpisahkan dengan main?	168
BAGIAN 9 PEMBUATAN KELAS DAN OBJEK	169
Apakah semesta main?	169
Bisakah diberikan gambaran mengenai pembuatan kelas?	169
Kelalu itu suatu tipe data?	171
Lalu, bagaimana cara menggunakan tipe tersebut untuk memberikan suatu variabel?	173
Pada contoh di depan, saya mendapati this yang diletakkan di depan nama kelas. Kenapa nama tidak dimulai oleh this. Apa maksud this di situ?	176
Bisakah nilai untuk objek diberikan ketika objek dipanggil?	177
Bagaimana cara memberikan nilai kegunaan pada argumen konstruktor sehingga argumen tersebut bisa tidak dibutuhkan?	180
Apakah semesta public pada field maupun metode?	182
Apakah kelas public tidak dibutuhkan pada metode atau field ber tipe public?	184
Bagaimana caranya agar file yang berisi definisi kelas terpisah dengan main?	186
Apakah objek bisa diletakkan sebagai argumen metode?	188
Nilai hasil fungsi yang berupa objek apa dimungkinkannya?	189
Saya pernah mendengar istilah metode kelas dan variabel kelas. Apa maksudnya?	190
Bagaimana administrasi kelas pewarisan yang disebut sub class di depan?	194
Bagaimana konsep penggunaan operator + seperti yang digunakan pada penghitungan miring bisa dilaksanakan?	187
BAGIAN 10 ARRAY, DAFTAR DATA	199
Apakah array itu?	199
Index array memang dimulai dari nol? Apa bisa diatur agar index dimulai dari satu?	200
Bagaimana cara membuat array yang berisi 6 elemen?	200
Apakah nilai setiap elemen dalam sebuah array tidak boleh ada yang berkali?	201
Bagaimana cara mengambil sebuah elemen array?	201

Adakah perintah untuk mengetahui ukuran <i>array</i> ?	203
Apakah ukuran <i>array</i> bisa diubah-ubah ketika program dijalankan?	204
Bisakah deklarasi <i>array</i> langsung diikuti dengan pemberian nilai setiap elemennya?	204
Saya bermaksud menciptakan <i>array</i> dengan 7 elemen.	
Apakah semua elemen harus diisi?	205
Saya ingin mengenolkan semua elemen <i>array</i> . Bagaimana caranya?	205
Apa yang terjadi kalau saya mengakses elemen yang sebenarnya tidak terdapat pada <i>array</i> ?	207
Bagaimana cara yang efektif untuk menghitung nilai rata-rata semua elemen pada <i>array</i> ?	208
Apakah tipe elemen <i>array</i> bisa berupa <i>char</i> ?	209
Saya mempunyai tabel dengan baris menyatakan cabang dan kolom menyatakan jumlah penjualan per semester. Bisakah dinyatakan dengan <i>array</i> ?	210
Bagaimana cara inisialisasi <i>array</i> berdimensi dua ketika dideklarasikan?	212
Apakah <i>array</i> berdimensi tiga juga ada?	213
Bisakah <i>array</i> dijadikan sebagai argumen fungsi?	216
Bagaimana dengan argumen yang berupa <i>array</i> berdimensi dua?	217
Apakah suatu metode bisa memberikan nilai balik berupa <i>array</i> ?	218
Saya bermaksud mengurutkan data yang ada di <i>array</i> . Bagaimana caranya?	219
Berdasarkan contoh <code>Arrays.sort()</code> di depan, isi argumen berupa <i>array</i> bisa diubah oleh metode?	222
Apakah <code>Arrays.sort()</code> dapat dipakai untuk mengurutkan data string secara descending?	222
Saya mencoba menerapkan <code>Arrays.sort(data, Collections.reverseOrder())</code> ; untuk data bertipe <i>int</i> . Kok dianggap salah, ya?	224
Selain untuk pengurutan data, <code>Arrays</code> menyediakan fasilitas apa saja?	225
Di depan disinggung adanya <i>array</i> yang ukurannya bisa diubah sewaktu program dieksekusi. Mohon dijelaskan!	226
Bagaimana saya bisa mengetahui jumlah elemen di <code>ArrayList</code> ?	229
Betulkah ada bentuk <i>for</i> yang dikhususkan untuk menangani <i>array</i> ?	229

BAGIAN 11 DERETAN KARAKTER 231

Operator <code>+</code> memang bisa digunakan untuk menggabungkan string?	231
Apakah fungsi <code>charAt()</code> dapat digantikan dengan <code>[]</code> ?	232
Bagaimana cara menyalin bagian suatu string?	232
Bagaimana kalau saya ingin membandingkan dua string?	234
Apakah ada fungsi untuk mengubah semua huruf kecil dalam string menjadi huruf kapital?	239
Apakah terdapat metode yang bermanfaat untuk menghapus spasi yang terletak di bagian depan atau bagian akhir string?	240

Benarkah terdapat metode yang dapat dipakai untuk menentukan suatu string berakhiran dengan substring tertentu?	240
Saya ingin mengetahui posisi suatu substring di dalam string.	
Metode apa yang bisa saya gunakan?	242
Bagaimana cara mengetahui string kosong atau tidak?	244
Apakah pencocokan dengan ekspresi regular dimungkinkan?	244
Bisakah diberi gambaran mengenai simbol-simbol apa saja yang dapat digunakan di ekspresi regular?	246
Adakah metode yang dapat dipakai untuk mengganti suatu karakter di string dengan karakter lain?	247
Metode apa yang bisa saya pakai untuk memecah suatu string menurut tanda pemisah tertentu?	249
Ada nggak fasilitas untuk mengonversikan string menjadi array bertipe char?	251
Kalau begitu, apakah ada pula metode untuk mengubah array bertipe char ke string?	251
Bagaimana cara mengonversikan bilangan ke String?	252
Bagaimana cara mengonversikan string ke bilangan?	253

BAGIAN 12 PENANGANAN WAKTU 255

Apakah Java memiliki tipe data primitif yang berhubungan dengan tanggal atau jam?	255
Bagaimana cara mendapatkan tanggal sekarang?	255
Bagaimana cara mendapatkan hari sekarang?	256
Apakah Calendar menyediakan metode untuk menghitung tanggal sebelum atau sesudah suatu tanggal dengan jarak beberapa hari?	257
Misalnya, saya memiliki dua variabel yang masing-masing berisi tanggal.	
Bagaimana caranya saya membandingkan bahwa tanggal di satu variabel lebih dahulu daripada tanggal di variabel kedua?	259
Saya ingin mengetahui suatu tahun termasuk tahun kabisat atau tidak.	
Bagaimana caranya?	261
Apakah tanggal yang tercatat di file bisa dibaca?	262

BAGIAN 13 PENANGANAN EKSEPSI 263

Eksepsi itu artinya apa?	263
Lalu, penanganan eksepsinya seperti apa?	264
Benarkah bahwa try bisa mengandung lebih dari satu catch?	267
Apakah try mendukung blok finally?	269
Lho, try di dalam try dimungkinkan?	270

Berdasarkan contoh-contoh yang diberikan, saya bisa menyimpulkan bahwa tanpa penanganan eksepsi pun hal-hal yang dibahas tersebut bisa ditangani dengan if.	
Mohon tanggapan?	271
Bisakah dijelaskan mengenai eksepsi lebih lanjut sehingga saya bisa memahami berbagai eksepsi terutama di <i>catch</i> ?	273
Apakah dimungkinkan untuk membuat penanganan suatu eksepsi yang memungkinkan saya memberikan pesan kesalahan tetapi saya tetap membiarkan eksepsi tetap terjadi?	274
Apakah <i>throw</i> bisa digunakan di luar <i>try</i> untuk menimbulkan eksepsi?	275

BAGIAN 14 FILE: PENYIMPAN DATA SECARA PERMANEN 277

Apakah operasi untuk menyimpan data ke <i>file</i> mudah dilakukan?	277
Bagaimana memastikan <i>file</i> sudah terbentuk?	280
Bagaimana cara menampilkan isi <i>file</i> melalui program C++?	281
Apakah isi <i>file</i> bisa ditambahi dengan data baru?	283
Apakah dimungkinkan untuk memformat data yang ingin ditulis ke <i>file</i> ?	286
Apakah dimungkinkan untuk menyimpan data yang berbentuk objek?	288
Contoh-contoh yang diberikan tampaknya beroperasi pada file biner.	
Bagaimana kalau saya ingin menggunakan file teks?	292
Bagaimana cara membaca isi file teks?	294
Apakah dimungkinkan untuk mengakses data secara random?	295
Bagaimana kalau saya ingin mendapatkan atribut <i>file</i> seperti kapan tanggal diciptakan ataupun ukuran <i>file</i> ?	299
Perintah apa yang digunakan untuk menghapus <i>file</i> ?	301
Bagaimana cara mengganti nama <i>file</i> ?	303
Apa perintah untuk membuat folder tersedia?	305
Bagaimana cara membaca isi folder?	305

BAGIAN 15 PENANGANAN THREAD 307

Apa sebenarnya thread itu dan apa kegunaannya?	307
Ya, saya sudah mengerti konsep seperti itu.	
Lalu, apa yang terjadi kalau <i>thread</i> lebih dari satu?	309
Saya sering melihat contoh penanganan thread yang melibatkan <i>sleep()</i> .	
Apa bedanya dengan penggunaan <i>Thread.yield()</i> ?	312
Bagaimana cara mewujudkan thread dengan <i>Runnable</i> ?	315
Apakah dimungkinkan untuk memonitor suatu thread masih berjalan atau sudah selesai?	317
Apakah suatu thread bisa diprioritaskan daripada thread yang lain?	319

BAGIAN 16 PENGGUNAAN KOLEKSI 323

Apa yang dimaksud koleksi?	323
Bisakah dijelaskan mengenai List?	323
Set? Apa maksudnya?	330
Queue? Apa itu?	336

BAGIAN 17 DASAR SWING 339

Bisakah dijelaskan kegunaan Swing?	339
Bagaimana cara membuat jendela seperti contoh di atas?	340
Apakah dimungkinkan untuk membuat jendela yang terkesan membesar ketika dijalankan?	342
Apa yang bisa ditambahkan di jendela?	344
Komponen apa saja yang disediakan di Swing?	344
Bisakah diberikan contoh penggunaan kontainer dan komponen non-kontainer?	345
Bagaimana caranya meletakkan jendela di tengah layar?	348
Bagaimana cara supaya letak kontainer atau tombol bisa diatur?	348
Apakah tombol bisa dilengkapi dengan ikon?	348
Mungkinkah tulisan di tombol diletakkan di kiri ikon?	350
Kalau saya ingin memasang tooltip di tombol, apakah dimungkinkan?	353
Bagaimana cara menangani klik pada tombol?	354
Bagaimana kalau tombol yang akan ditangani lebih dari satu? Apakah penanganan klik ada dua?	356
Apakah judul di tombol bisa diganti sewaktu program dieksekusi?	360
Bagaimana cara menampilkan teks yang biasa diletakkan di kiri atau di atas suatu komponen?	363
Komponen apa yang dapat dipakai pemakai untuk memasukkan data dari keyboard ?	365
Bagaimana caranya agar data <i>password</i> tidak ditampilkan apa adanya di <code>JTextField</code> ?	370
Program: BarisTeks.Java	372
Bagaimana caranya untuk menerima masukan beberapa baris dari pemakai?	372
Apakah text area memang tidak memiliki batang penggulung?	373
Bagaimana caranya kalau saya mau menambahkan combo box?	375
Apakah Swing mendukung List Box?	378
Bagaimana cara untuk menyertakan tombol radio dan kotak cek?	381
Apakah judul seperti Message dalam kotak dialog yang diperoleh melalui <code>JOptionPane</code> bisa diubah?	387
Bagaimana cara membuat kotak dialog yang berisi konfirmasi?	390
Bisakah judul tombol seperti Yes atau No diganti dengan Ya atau Tidak?	393
Apakah ada kotak dialog yang digunakan untuk memasukkan suatu nilai dari keyboard?	395
Apakah <code>JSlider</code> itu untuk membuat slider?	397
Progress bar juga ada, ya?	401
Apakah <code>JToolBar</code> berguna untuk membuat toolbar?	403



BAGIAN 1

PENGENALAN JAVA

Bab ini menjelaskan dasar bahasa Java, yang mencakup:

- pengertian program, flowchart, dan algoritma;
- instalasi Java;
- pemasangan Eclipse;
- penulisan program;
- penggunaan komentar;
- penulisan pernyataan;
- penanganan kesalahan;
- pengenalan pemrograman berorientasi objek.

Apa sebenarnya Java Itu?

Java adalah nama sebuah bahasa pemrograman yang sangat terkenal. Sebagai bahasa pemrograman, Java dapat digunakan untuk menulis program. Sebagaimana diketahui, program adalah kumpulan instruksi yang ditujukan untuk komputer. Melalui program, komputer dapat diatur agar melaksanakan tugas tertentu sesuai yang ditentukan oleh pemrogram (orang yang membuat program). Sebagai contoh, Anda mungkin terbiasa menggunakan pengolah kata seperti Word. Nah, Word adalah contoh aplikasi yang dibuat dengan menggunakan bahasa pemrograman.

Jadi, komputer itu terlihat pandai karena ada program yang dibuat oleh pemrogram?

Ya, memang begitu. Makanya, menjadi pemrogram itu menantang. Bisa mengendalikan komputer. Asyik, kan?



Bahasa Java dikembangkan di Sun Microsystems dan mulai diperkenalkan kepada publik pada tahun 1995. Seperti halnya C++, Java juga merupakan bahasa yang berorientasi objek. Dengan demikian, Java juga memudahkan dalam pembuatan aplikasi yang berskala besar.

Sebagai bahasa yang beraras tinggi, yang menggunakan perintah-perintah yang mudah dimengerti oleh orang, Java mempunyai keunggulan yakni bersifat universal. Sebagai bahasa yang universal, Java bisa dijumpai di berbagai platform (Linux, UNIX, Windows, Mac, dan lain-lain). Artinya, jika Anda menguasai Java di platform PC, sangat mudah untuk berpindah di Linux ataupun sistem operasi yang lain. Hal yang menarik lagi, hasil kompilasi Java yang dinamakan *bytecode* dapat dijalankan di berbagai platform sepanjang di sistem target memiliki *Java Runtime Environment* (JRE).

Apa bedanya program dan aplikasi?

Aplikasi itu sebenarnya adalah program, tetapi berbeda dari titik pandang. Program adalah istilah yang biasa dipakai oleh pembuat program, sedang aplikasi adalah istilah dari sisi pemakai (*user*). Program adalah yang ditulis oleh pemrogram, sedangkan aplikasi adalah hasil terjemahan program, berupa kode yang dipahami oleh mesin. Pada sistem Windows, hasil terjemahan program disimpan di dalam *file* berekstensi *.exe*. *File* inilah yang bisa dijalankan oleh pemakai. Pada Java, hasil terjemahan berupa *bytecode*. Kode tersebut diproses oleh JRE dan selanjutnya dapat digunakan untuk mengontrol mesin agar melaksanakan tindakan yang telah diprogram oleh pemrogram.

Selain kode yang dipahami oleh mesin (biasa disebut **kode mesin**) ataupun yang berbentuk *bytecode*, terdapat istilah kode sumber (*source code*). Kode sumber adalah kode asli yang ditulis oleh pemrogram. Kode dalam bahasa Java yang disimpan dengan ekstensi *.java* adalah contoh kode sumber.

Saya awam dengan pemrograman. Apakah saya bakal bisa membuat program dengan Java?

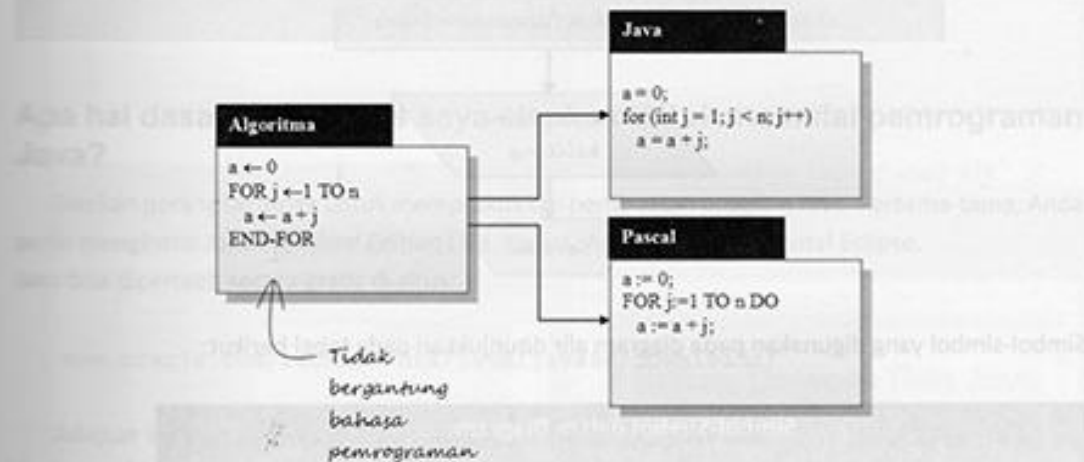
Jangan khawatir. Semua para pemrogram profesional pada awalnya tidak dapat membuat program. Namun, semangat yang tinggi membuat mereka akhirnya melewati tahap awal seperti Anda dan kemudian dengan cepat dapat menguasai elemen-elemen dasar pemrograman. Sejatinya, Java mudah dipelajari. Anda akan merasakan bahwa Java adalah bahasa yang menyenangkan dan gampang untuk dipahami.

Apa betul untuk menguasai pemrograman harus belajar algoritma terlebih dulu?

Perlu dijelaskan dulu mengenai algoritma. Algoritma sebenarnya adalah urutan langkah yang ditujukan untuk menyelesaikan masalah dengan menggunakan komputer. Kalau di depan telah

dijelaskan tentang makna program, algoritma mirip dengan program. Namun, ada perbedaan penting yang perlu diketahui. Algoritma independen terhadap bahasa pemrograman, sedangkan program sangat bergantung pada bahasa pemrograman. Jadi, algoritma lebih bersifat umum.

Hal penting yang perlu digarisbawahi di sini, ketika menulis program, tentu saja Anda perlu memikirkan urutan langkah yang diperlukan agar komputer menuruti perintah Anda. Dengan perkataan lain, algoritma dan program bisa dipelajari secara bersama-sama. Sekadar sebagai ilustrasi, perbedaan algoritma dan program Java dan Pascal ditunjukkan pada gambar berikut.



Contoh di atas memperlihatkan bahwa sebuah algoritma bisa dituangkan ke dalam dua bahasa pemrograman. Tentu saja, kode untuk bahasa pemrograman Java dan Pascal berbeda walaupun ada kemiripan. Suatu algoritma juga bisa berbentuk instruksi dalam bahasa manusia seperti berikut:

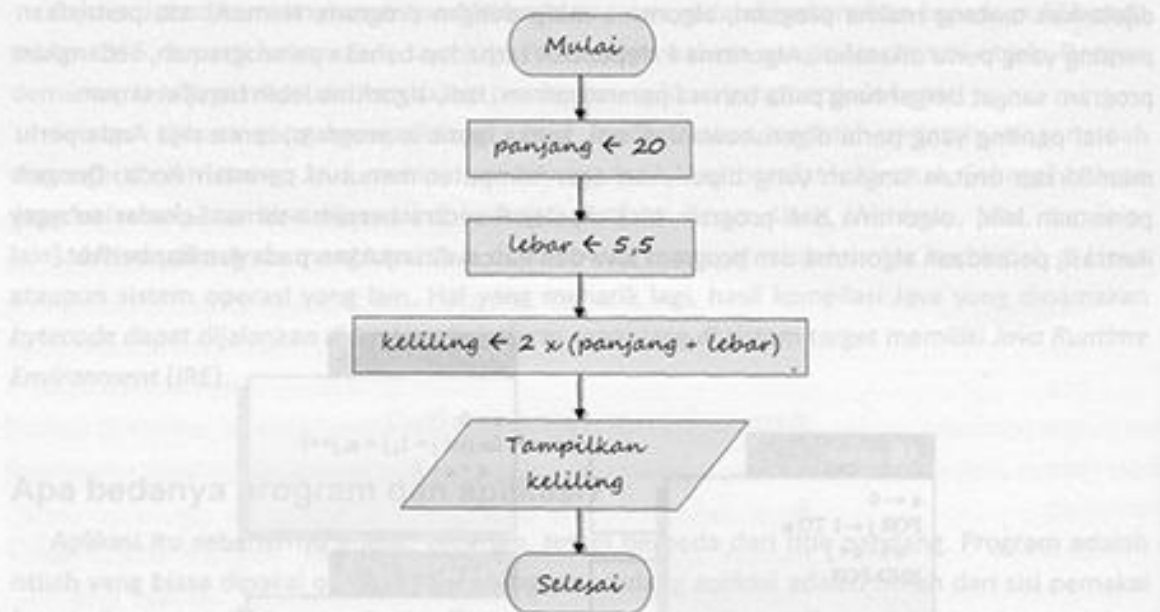
- tampilkan tulisan "Jumlah buah: ";
- tampilkan nilai yang terkandung dalam variabel jumlah.

Jika diterjemahkan ke dalam bahasa Java, kodenya seperti berikut:





```



System.out.println("Jumlah buah: ");
System.out.println(jumlah);
  
```

Algoritma terkadang disajikan dalam bentuk diagram alir. Dengan menggunakan pendekatan yang menyajikan penyelesaian suatu persoalan secara visual, terkadang pemecahan ke dalam program menjadi lebih mudah. Contoh algoritma ditunjukkan pada gambar berikut.



Simbol-simbol yang digunakan pada diagram alir ditunjukkan pada tabel berikut:

Simbol-Simbol dalam Diagram:	
Simbol	Keterangan
	Tanda yang menyatakan awal atau akhir diagram alir. Tulisan yang diletakkan di dalamnya bisa berupa "Mulai" atau "Selesai". Kadangkala, tanda ini tidak disertakan dalam diagram alir
	Tanda yang menyatakan suatu proses atau perhitungan. Umum digunakan untuk menyatakan pemberian nilai ke suatu variabel
	Tanda yang menyatakan operasi masukan (membaca dari <i>keyboard</i>) atau keluaran (menampilkan informasi ke layar)
	Tanda yang digunakan sebagai pengambilan keputusan. Di dalamnya berisi suatu kondisi yang bisa bernilai ya atau tidak

Simbol-Simbol dalam Diagram:	
Simbol	Keterangan
	Menyatakan arah aliran pemrosesan
	Berfungsi sebagai tanda penghubung yang menyatakan arah aliran pemrosesan. Pada contoh di samping, dua aliran menuju ke satu aliran. Namun, seringkali gambar lingkaran tidak disertakan

Apa hal dasar yang harus saya siapkan untuk memulai pemrograman Java?

Siapkan perangkat lunak untuk mempraktikkan pembuatan program Java. Pertama-tama, Anda perlu menginstal *Java Standard Edition (SE)*. Kedua, Anda perlu menginstal Eclipse. Java bisa diperoleh secara gratis di situs:

www.oracle.com/technetwork/java/javase/downloads/

Adapun Eclipse adalah *software* tersebut tergolong sebagai *Open Source*. Dengan demikian, Anda juga bisa menggunakan Eclipse secara gratis. Anda bisa mengunduh Eclipse for Java Developer di www.eclipse.org. Namun, untuk kemudahan Anda, *software* tersebut maupun Java disediakan pada CD yang disertakan bersama buku ini.

Bagaimana cara menginstal Java?

Instalasi Java dapat dilakukan dengan mudah. Ikuti saja petunjuk berikut.

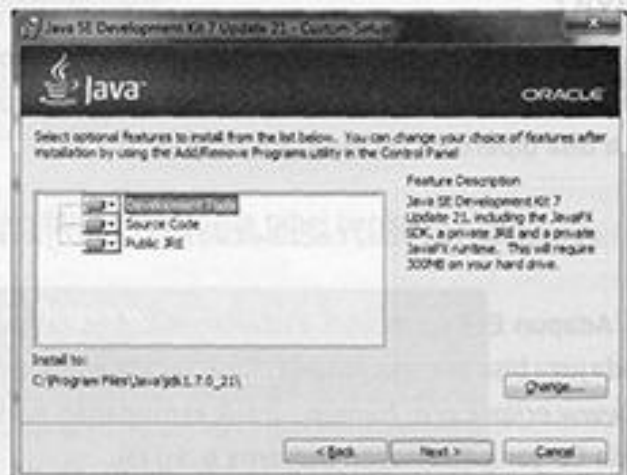
1. Klik ganda pada file `jdk-7u21-windows.exe` untuk menjalankannya. Langkah ini memunculkan tampilan seperti berikut.



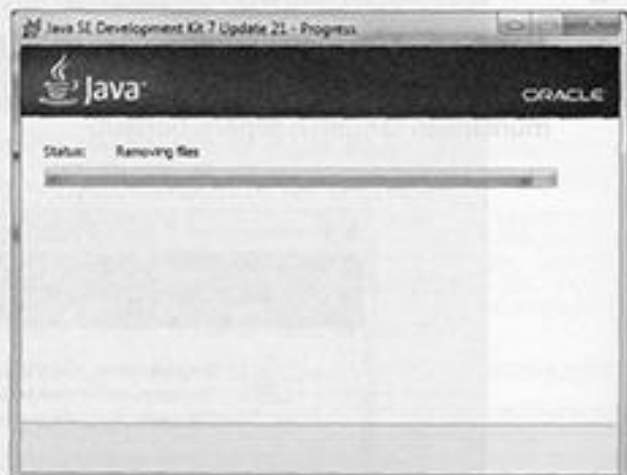
2. Klik pada tombol **Yes**. Langkah ini memunculkan tampilan sebagai berikut:



3. Klik pada tombol **Accept >**. Langkah ini memunculkan tampilan sebagai berikut:



4. Klik pada tombol **Next >**. Tunggu beberapa saat setelah menjumpai tampilan berikut:



5. Sesaat kemudian akan muncul tampilan sebagai berikut.



6. Klik pada tombol **Next >**. Langkah ini memunculkan tampilan sebagai berikut:



7. Tunggu sampai dijumpai tampilan sebagai berikut:



8. Klik pada tombol **Close**.

Bisakah saya dibantu untuk melakukan pemasangan Eclipse di komputer saya?


Instalasi Eclipse dapat dilakukan dengan mudah. Ikuti saja petunjuk berikut.

1. Buatlah *folder* bernama C:\LatJava.
2. Salinlah *file* eclipse-java-juno-SR2-win32 yang terdapat pada folder Software pada CD yang disertakan bersama buku ini ke folder C:\LatJava.
3. Dekompresilah *file* hasil salinan tersebut sehingga terbentuk subfolder eclipse di dalam folder C:\LatJava.

Setelah itu, Anda sudah bisa memulai untuk membuat program Java.

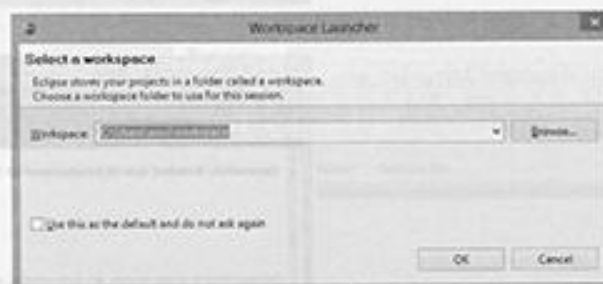
Saya melihat keberadaan folder Eclipse di komputer saya. Sekarang apa yang harus saya lakukan?

Anda bisa memulai untuk belajar pemrograman. Pertama-tama, lakukan langkah seperti berikut untuk memanggil Eclipse.

1. Klik ganda pada folder eclipse.
2. Klik ganda pada  eclipse. Langkah ini membuat tampilan berikut segera muncul:

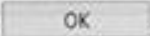



Beberapa saat kemudian Anda akan menjumpai tampilan seperti berikut:



3. Ketikkan C:\LatJava pada kotak di kanan judul **Workspace**.
4. Centang pada kotak cek sehingga menjadi seperti berikut:

☒ Use this as the default and do not ask again



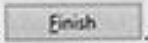
5. Klik pada tombol .

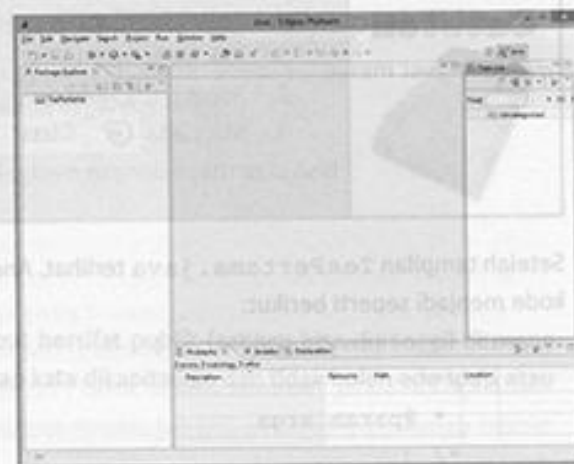
6. Klik pada posisi berikut  untuk menutup jendela Welcome. Hasilnya, akan muncul tampilan seperti berikut:



Lalu, bagaimana cara menuliskan program?


Pertama-tama, Anda perlu membuat proyek baru. Di proyek itulah Anda membuat kelas yang ditujukan untuk menuliskan program yang Anda kehendaki. Berikut adalah cara membuat proyek baru dengan nama TesPertama.

1. Klik pada menu **File**.
2. Sorot pada **New**.
3. Klik pada  **Java Project**. Langkah ini memunculkan kotak dialog seperti berikut:
4. Ketikkan **TesPertama** pada kotak di kanan **Project name**.
5. Klik pada tombol radio,  untuk memilih JRE bawaan (dalam hal ini adalah Jre7).
6. Klik pada tombol .



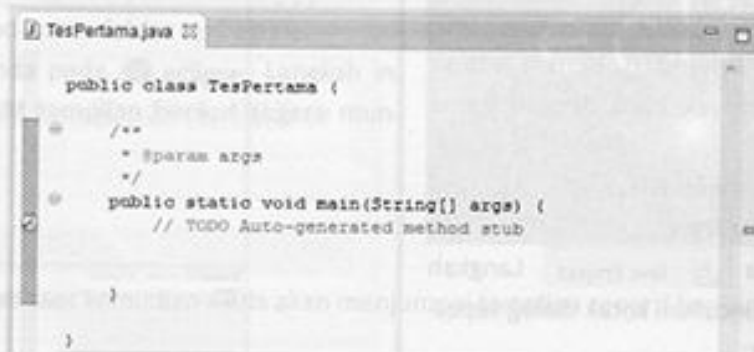
Hasil akhir langkah di atas diperlihatkan di gambar di samping:

Selanjutnya, lakukan pembuatan kelas seperti berikut.

1. Klik pada menu **File**.
2. Sorot pada **New**.
3. Klik pada  **Class** . Langkah ini membuat kotak dialog seperti gambar di samping:
4. Ketikkan **TesPertama** pada kotak di kanan **Name**.
5. Centanglah pada kotak cek.
☐ `public static void main(String[] args)` untuk menambahkan metode `main()`.
6. Klik pada tombol **Finish**.




Hasilnya, muncul jendela berjudul **TesPertama.java** seperti berikut:




Catatan



Ketiga langkah berikut

1. Klik pada menu **File**.
2. Sorot pada **New**.
3. Klik pada  **Class** .

bisa digantikan dengan mengklik **New Java Class** () pada *main toolbar*.

Setelah tampilan **TesPertama.java** terlihat, Anda bisa menuliskan kode sehingga keseluruhan kode menjadi seperti berikut:

```
public class TesPertama {
    /**
     * @param args
     */
}
```



```

public static void main(String[] args) {
    // TODO Auto-generated method stub
    System.out.println("Selamat belajar Java!");
}

```

Tekan **Ctrl+S** untuk menyimpan kode yang baru saja Anda tambahkan.

Catatan



Java adalah bahasa yang sensitif terhadap huruf kapital dan huruf kecil atau dikenal dengan sebutan *case sensitive*. Artinya, Java membedakan huruf kapital dan huruf kecil di dalam kode. Sebagai contoh, huruf S kapital harus digunakan pada `System.out.println()`. Jika Anda menggunakan huruf s kecil, perintah tersebut menjadi tidak dikenal oleh kompiler Java.

Saya telah menuliskan Kode. Bagaimana cara memperoleh hasil program tersebut?

Untuk menjalankan program, pilihlah menu **Run** terlebih dahulu. Kemudian, klik pada **Run**. Cara lain adalah dengan menekan tombol **Ctrl+F11**. Jika kode yang Anda tulis tidak ada yang salah, akan diperoleh hasilnya pada tab Console, sebagaimana terlihat di gambar berikut:



Bisakah dijelaskan mengenai kode yang baru saja saya praktikkan?

Kode sumber di depan (`TesPertama.java`) adalah contoh sebuah program Java. Program Java ditulis dalam bentuk kelas. Bentuk kelas adalah seperti berikut:

```

public class NamaKelas {

```

Kata `public` menyatakan bahwa kelas tersebut bersifat publik (artinya bisa dipanggil di mana saja). Nama kelas biasa ditulis dengan awal setiap kata dikapitalkan dan tidak boleh ada spasi atau simbol-simbol tertentu di nama kelas.

Kenapa harus Java?

Karena Java yang menjadi tulang punggung untuk membuat aplikasi-aplikasi di peranti ponsel berbasis Android.

Buku ini didedikasikan untuk siapa saja yang ingin mempelajari pemrograman berbasis *desktop* yang menggunakan Java. Berbagai aspek dasar yang perlu dikuasai dalam bidang pemrograman dikupas di buku ini, dimulai dengan hal yang paling mudah dan mendasar. Berbagai masalah yang biasa dihadapi oleh pemula disertakan dengan tujuan agar mempelajari pemrograman terasa mudah dan menyenangkan.

Untuk membantu mempermudah dalam menguasai materi yang dibahas dalam buku ini, program **Java SE7**, **Eclipse** dan juga *file-file* gambar yang digunakan untuk latihan disertakan bersama buku ini.

- ▶ Pengenalan Java
- ▶ Variabel, Konstanta, dan Literal
- ▶ Ekspresi, Operator, dan Operand
- ▶ Input/Output
- ▶ Keputusan dengan if dan switch
- ▶ Pengulangan dengan while dan do..while
- ▶ Pengulangan dengan for
- ▶ Pembuatan Metode
- ▶ Pembuatan Kelas dan Objek
- ▶ Array: Daftar Data
- ▶ String: Deretan Karakter
- ▶ Penanganan Waktu
- ▶ Penanganan Eksepsi
- ▶ File: Penyimpan Data Secara Permanen
- ▶ Penanganan Thread
- ▶ Penggunaan Koleksi
- ▶ Dasar Swing
- ▶ Penggunaan Layout Manajer



Abdul Kadir, telah menulis puluhan buku di bidang Teknologi Informasi. Selain mengajar dan mengisi kegiatan pelatihan, ia juga ikut aktif dalam pengembangan berbagai sistem informasi.

Medi@Kom

Penerbit Mediakom

Jl. Cempaka Putih No. 8

Derasan CT X, Gejayan,

Yogyakarta 55283

Telp. 0274-556043, 0274-555939

Faks. 0274-546020

Email: penerbitmediakom@gmail.com

Website: www.mediakom-penerbit.com

ISBN (13) 978-979-877-368-6
ISBN 979-877-368-3

