

Mengenal **Java** dan **Database** dengan **NetBeans**

ustaka-indo.blogspot.com

Jubilee Enterprise

pustaka-indo.blogspot.com

pustaJa-indo.blogspot.com

MENGENAL JAVA DAN DATABASE DENGAN NETBEANS

Sanksi Pelanggaran Pasal 113
Undang-Undang Nomor 28 Tahun 2014
tentang Hak Cipta

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
3. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
4. Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah).

MENGENAL JAVA DAN DATABASE DENGAN NETBEANS

Jubilee Enterprise

PENERBIT PT ELEX MEDIA KOMPUTINDO



KOMPAS GRAMEDIA

Mengenal Java dan Database dengan NetBeans

Jubilee Enterprise

©2015, PT Elex Media Komputindo, Jakarta

Hak cipta dilindungi undang-undang

Diterbitkan pertama kali oleh

Penerbit PT Elex Media Komputindo

Kelompok Gramedia, Anggota IKAPI, Jakarta 2015

elizabet@elexmedia.co.id

715051392

ISBN: 978-602-02-6891-0

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penerbit.

Dicetak oleh Percetakan PT Gramedia, Jakarta

Isi di luar tanggung jawab percetakan

Kata Pengantar

Java yang dipadukan dengan database adalah pemrograman masa depan. Google mencatat bahwa Java adalah bahasa pemrograman yang paling banyak dipelajari di seluruh dunia. Kalau sudah begini, maka dapat dipastikan jika aplikasi-aplikasi yang beredar di dunia dibuat menggunakan Java.

Oleh karena itu, penting kiranya jika sejak awal kita mempelajari Java, khusus yang dikaitkan dengan database. Di dalam buku ini, kita akan mempelajari pemrograman database menggunakan Java yang disokong oleh NetBeans.

Di akhir bab, Anda akan diajak untuk membuat aplikasi siap-pakai yang bisa dikembangkan sendiri sesuai kebutuhan. Semoga buku ini bisa membantu Anda mengenali Java dan database secara cepat dan praktis.

Yogyakarta, 23 Mei 2015

Gregorius Agung

Founder Jubilee Enterprise

"Devotion for Innovation"

Do you need top-notch IT Book? Just thinkjubilee.com

Daftar Isi

Kata Pengantar	v
Daftar Isi	vi

BAB 1 PENDAHULUAN 1

Java	1
Berbagai Kelebihan Java.....	1
Konsep PBO (Pemrograman Berbasis Objek)	3
Objek	3
Kelas	4
Abstraksi Data	5
Enkapsulasi Data.....	5
Pewarisan	5
Polimorfisme	6
Database.....	6
Terminologi RDBMS.....	7
Mengenal NetBeans IDE	8

BAB 2 INSTALASI JAVA DAN NETBEANS 11

Instalasi Java.....	11
Menginstal NetBeans	14
Windows	14
Mac OS	16
Linux	17

BAB 3 DASAR-DASAR PEMROGRAMAN JAVA19

Membuat Aplikasi 'Halo Dunia'	19
Sintaks Dasar Java	21
Identifier dalam Java	21
Modifier dalam Java	22
Variabel dalam Java	22
Array dalam Java	22
Enum dalam Java	22
Kata Kunci dalam Java	23
Menulis Komentar dalam Java	23
Penggunaan Spasi dalam Java	24
Pewarisan	24
Interface	24
Objek dalam Java	24
Kelas dalam Java	25
Konstruktor	25
Membuat Objek	26
Mengakses Variabel Instance dan Method	26
Package Java	27
Tipe Data dalam Java	28
Tipe Data Primitif	28
byte	28
short	28
int	29
long	29
float	29
double	30
boolean	30
char	30
Tipe Data Referensi	30
Literal dalam Java	31
Tipe Variabel dalam Java	31
Variabel Lokal	32

Variabel Instance.....	33
Variabel Kelas/Static.....	34
Modifier dalam Java	35
Modifier Akses	35
Modifier Akses Default - tanpa kata kunci.....	36
Modifier Akses Privat - private.....	36
Modifier Akses Publik - public	36
Modifier Akses Protected - protected	37
Kontrol Akses dan Pewarisan.....	37
Modifier Non Akses.....	38
Operator dalam Java.....	38
Operator Aritmetika.....	38
Operator Relasional.....	39
Operator Logika.....	40
Operator Assignment.....	41
Operator Kondisional (? :).....	42
Operator instanceof.....	42
Preseden Operator-Operator Java	43
Kontrol Perulangan dalam Java	44
Perulangan while.....	44
Perulangan do...while.....	45
Perulangan for	46
Perulangan for... Versi Baru dalam Java	47
Break	48
Continue	49
Pengambilan Keputusan dalam Java	49
Statemen if	50
Statemen if...else.....	50
Statemen if...else if...else	51
Statemen if...else Bersarang	52
Statemen Switch	53
BAB 4 MENYIAPKAN DATABASE.....	57
WAMP, MAMP, dan LAMP	57
Menginstal WAMP di Windows.....	57

Menginstal MAMP di Mac OS X	63
Konfigurasi MySQL.....	66
Menginstal LAMP di Linux.....	67
BAB 5 PENGENALAN SQL	69
Istilah-Istilah Penting dalam Database	70
Perintah-Perintah dalam SQL.....	70
Membuat Database	71
Membuat Pengguna.....	72
Membuat Tabel	73
Tipe Data dalam SQL.....	74
CHAR	75
BINARY.....	75
TEXT	75
BLOB	76
NUMERIK	76
Tanggal dan Waktu.....	78
Menambah Data ke dalam Tabel	78
Mengganti Nama Tabel	79
Mengganti Tipe Data Kolom	80
Menambah Kolom Baru	80
Mengganti Nama Kolom.....	80
Menghapus Kolom	81
Menghapus Tabel.....	81
Query dalam Database SQL.....	81
SELECT.....	82
DELETE	82
WHERE	83
LIMIT.....	84
UPDATE ... SET	85
ORDER BY	86
GROUP BY	86
Menggunakan Operator Logika	87

BAB 6 MEMBUAT APLIKASI BERBASIS DATABASE SEDERHANA89

Pembuatan Database.....	89
Menyiapkan Aplikasi	92

BAB 7 CONTOH PEMBUATAN APLIKASI JAVA BERBASIS DATABASE.....97

Menyiapkan Database.....	97
Membuat Tabel dalam dbpenggajian	99
Tabel Karyawan.....	99
Tabel Jabatan	102
Tabel Pengguna	104
Package	107
Package Entity	107
Package Koneksi	108
Package Utama.....	109
Kelas	111
Kelas dalam Package Entity.....	111
Kelas dalam Package Koneksi.....	117
Membuat Antarmuka (Graphical User Interface) Aplikasi	118
Form	119
Tabbed Pane	120
Panel Karyawan.....	123
Panel Jabatan	127
Panel Gaji	131
Menghubungkan Aplikasi dengan Database	132
Tabel Karyawan.....	132
Tabel Jabatan	134
Menambahkan Data dalam Database	137
Menambahkan Data Karyawan	137
Menambah Data Jabatan	140
Menampilkan Gaji	143
Tentang Penulis	149

BAB 01

Pendahuluan

Java

Java merupakan pemrograman yang sangat populer karena rentang aplikasi yang bisa dibuat menggunakan bahasa ini sangatlah luas, mulai dari komputer hingga smartphone. Bahasa pemrograman Java dikembangkan pertama kali oleh Sun Microsystem yang dimulai oleh James Gosling dan dirilis pada tahun 1995. Saat ini Sun Microsystem telah diakuisisi oleh Oracle Corporation. Apabila sudah terbiasa dengan bahasa C dan C++, Anda bisa mempelajari Java dengan cepat.

Java bersifat *Write Once, Run Anywhere* (program yang ditulis satu kali dan dapat berjalan pada banyak platform). Dengan demikian tidak mengherankan apabila aplikasi yang dibuat menggunakan Java bisa ditemukan di lingkungan komputer dan smartphone tanpa perbedaan yang berarti.

Sama seperti pemrograman pada umumnya, Java merupakan bahasa pemrograman yang mampu bekerja dengan sebuah database. Nah di dalam buku ini, kita akan mempelajari pemrograman database dengan memanfaatkan Java dan Netbeans sebagai tool untuk perancangan script aplikasi berbasis Java tersebut.

Berbagai Kelebihan Java

Jika Anda mengetik ke dalam textbox pencarian “Google” dengan kata kunci berbentuk pertanyaan: *what is the most popular programming language*, maka Google akan menjawab: Java.

Memang Java merupakan bahasa pemrograman paling populer dan secara resmi dipelajari di bangku perkuliahan. Popularitas tersebut terjadi karena Java memiliki berbagai kelebihan yang dirinci dalam setiap fiturnya.

Berikut ini fitur-fitur Java:

- **Berorientasi Objek:** Dalam Java, semua adalah Objek.
- **Bersifat Platform Independent:** Java di-compile dalam bit kode platform independen. Itu artinya, aplikasi yang ditulis menggunakan Java dapat dengan mudah dibuka menggunakan berbagai sistem komputer dan arsitektur komputer. Ini berbeda dengan aplikasi yang dibuat memakai Visual Basic, misalnya, yang secara umum hanya bisa dibuka menggunakan komputer bersistem operasi MS Windows. Oleh karena itu tidak heran jika aplikasi yang dibuat menggunakan Java umumnya bisa dibuka menggunakan komputer, smartphone, hingga perangkat IT lainnya.
- **Sederhana:** Java didesain untuk dapat dengan mudah dipelajari sehingga penetrasi pemrograman ini cukup tinggi di kalangan para pelajar.
- **Aman:** Dengan fitur keamanan Java, Anda dapat membuat sistem yang bebas virus dan powerful.
- **Bersifat Architectural-neutral:** Compiler Java membuat format file objek yang architectural-neutral, yang membuat kode yang di-compile dapat dieksekusi pada berbagai jenis prosesor yang memiliki sistem runtime Java.
- **Portabel:** Java bersifat portable karena adanya fitur platform independent dan architectural-neutral.
- **Kuat dan powerful:** Java mengeliminasi error dengan menjalankan pengecekan pada waktu compile dan runtime.
- **Multithreaded:** Dengan adanya fitur multithread, Anda dapat membuat program yang dapat mengerjakan banyak tugas sekaligus.
- **Terinterpretasi:** Kode bit Java ditranslasi secara langsung pada instruksi mesin dan tidak disimpan.
- **Performa tinggi:** Java memiliki performa yang tinggi karena menggunakan compiler langsung.

- **Terdistribusi:** Java didesain untuk lingkungan distribusi internet.
- **Dinamis:** Java lebih dinamis dari C dan C++ karena Java didesain untuk beradaptasi dengan lingkungan pengembangan.

Konsep PBO (Pemrograman Berbasis Objek)

Java merupakan bahasa pemrograman yang berbasis objek, atau biasa disebut dengan istilah OOP (*Object Oriented Programming*).

Berikut ini konsep-konsep dasar pemrograman berbasis objek:

- Objek
- Kelas
- Abstraksi Data
- Enkapsulasi Data
- Pewarisan
- Polimorfisme

Objek

Objek merupakan bagian penting dalam pemrograman berbasis objek. Secara umum, objek menyatukan data dan fungsi menjadi suatu unit dalam sebuah aplikasi. Perhatikan contoh berikut:

<u>Objek: SISWA</u>
<u>DATA</u>
Nama
Alamat
NIS
<u>METHOD</u>
Total()
RataRata()

Representasi dari objek SISWA

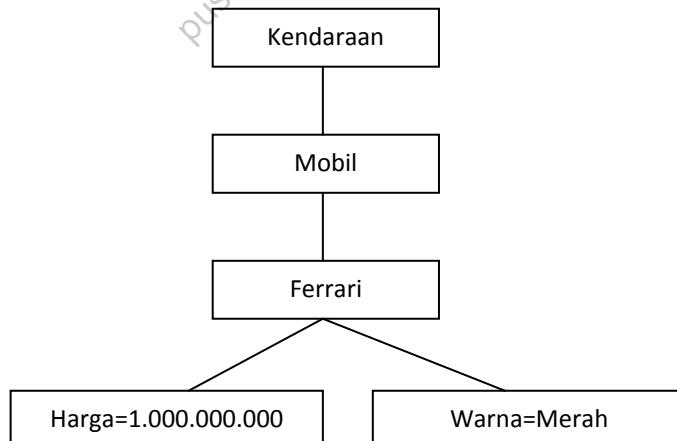
Setiap objek menampung data dan kode untuk mengelola data-data tersebut. Objek dapat berinteraksi tanpa harus mengetahui informasi rinci mengenai data atau kode dalam objek lain. Berikut ini contoh lain dari sebuah objek, yaitu MOBIL:

Objek: <u>MOBIL</u>
<u>DATA</u>
Warna
Harga
<u>METHOD</u>
Kunci()
Nyalakan()

Representasi dari objek MOBIL

Kelas

Kelas adalah sekumpulan objek dengan properti (atribut), perilaku (operasi), dan hubungan yang sama dengan antar objek. Sebuah objek merupakan variabel dengan tipe kelas. Kelas merupakan tipe data yang dibuat oleh pengguna. Setelah suatu kelas didefinisikan, Anda dapat membuat berapa pun jumlah objek dari kelas tersebut.



Contoh diagram kelas

Pada contoh di atas, objek Ferrari merupakan objek dari kelas Mobil. Setiap objek dari kelas Mobil memiliki Harga dan Warna masing-masing yang berbeda.

Abstraksi Data

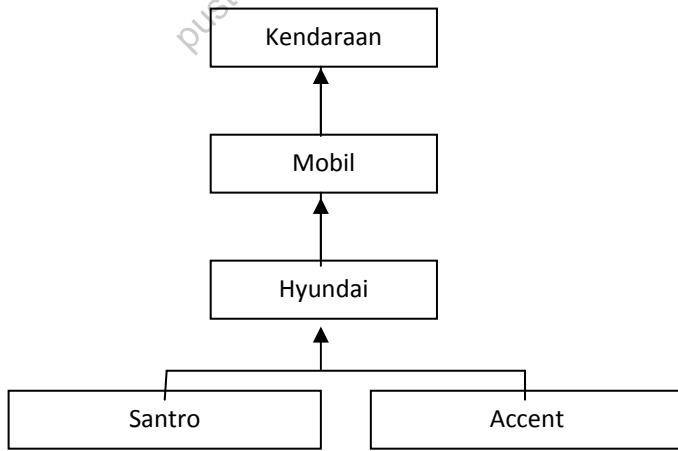
Abstraksi data dilakukan dengan menyediakan deskripsi penting tanpa menyertakan informasi detail atau penjelasan. Kelas-kelas menggunakan konsep abstraksi dan didefinisikan sebagai daftar atribut abstrak seperti ukuran, dan fungsi-fungsi yang mengelola atribut-atribut tersebut. Semua itu merangkum properti penting dari objek yang akan dibuat.

Enkapsulasi Data

Enkapsulasi data artinya membungkus data dan fungsi dalam sebuah unit tunggal. Dengan enkapsulasi, data tidak dapat diakses secara langsung dari luar. Untuk mengakses data, kita dapat memanfaatkan fungsi-fungsi yang ada dalam kelas tersebut.

Pewarisan

Pewarisan adalah proses yang memungkinkan objek dari suatu kelas dapat memperoleh properti dari objek dari kelas yang lain.

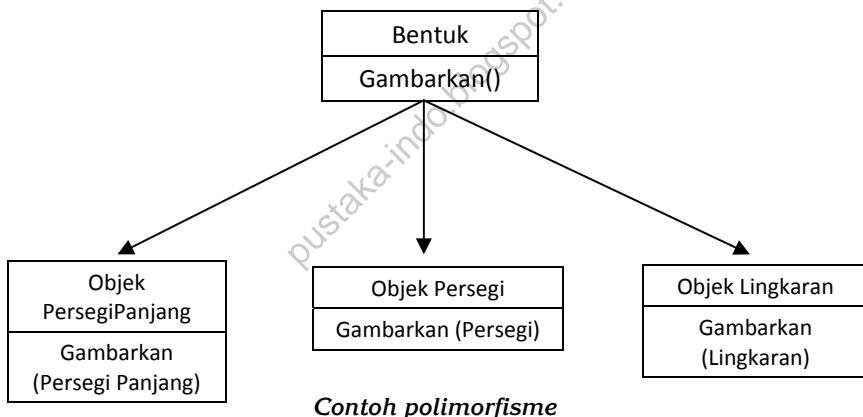


Pewarisan berarti suatu kelas mewarisi data dan perilaku dari kelas yang lain. Dengan pewarisan, Anda dapat menambahkan karakteristik pada suatu kelas tanpa harus memodifikasinya. Sebagai contoh dapat dijelaskan seperti ilustrasi pada gambar di atas.

Pada contoh tersebut, Santro merupakan bagian dari kelas Hyundai yang merupakan bagian dari kelas Mobil, dan Mobil merupakan bagian dari kelas Kendaraan. Dalam hal ini Kendaraan merupakan superkelas.

Polimorfisme

Polimorfisme merupakan kemampuan untuk memiliki lebih dari satu bentuk. Polimorfisme berarti bahwa operasi yang sama dapat melakukan hal yang berbeda untuk tiap-tiap kelas. Ilustrasinya sebagai berikut, kelas Singa apabila mendapatkan operasi “bergerak”, maka kelas ini akan menggerakkan kakinya. Namun dengan operasi “bergerak” yang sama untuk kelas Burung, maka kelas tersebut akan mengepakan sayapnya. Contoh lainnya sebagai berikut.



Database

Database adalah sebuah sistem yang berfungsi untuk menyimpan dan mengolah sekumpulan data. Setiap database mempunyai API tertentu untuk membuat, mengakses, mengatur, mencari, dan menyalin data yang ada di dalamnya sehingga bisa dimanfaatkan oleh aplikasi lainnya.

Untuk menampung dan mengatur data yang begitu banyak, Anda dapat menggunakan *Relational Database Management Systems* (RDBMS). Disebut relational database karena semua data disimpan dalam tabel-tabel yang berbeda dan dihubungkan berdasarkan relasinya satu dengan lainnya dengan menggunakan primary key dan foreign key.

Relational Database Management Systems (RDBMS) adalah aplikasi yang mampu:

- Memungkinkan Anda untuk mengimplementasikan sebuah database dengan tabel-tabel, kolom-kolom, dan indeks-indeks.
- Menjamin integritas referensi di antara baris-baris pada berbagai tabel.
- Meng-update indeks-indeks secara otomatis.
- Menginterpretasikan query SQL dan menggabungkan informasi dari berbagai tabel.

Terminologi RDBMS

Ketika bekerja dengan sebuah database, maka ada berbagai terminologi atau istilah-istilah yang mesti diperhatikan. Berikut ini istilah-istilah yang digunakan dalam dunia database:

- **Database:** merupakan kumpulan tabel-tabel yang berisi data-data yang saling berkaitan.
- **Tabel:** merupakan matriks berisi data. Tabel dalam database terlihat seperti spreadsheet sederhana.
- **Kolom (Column):** satu kolom (elemen data) mengandung data dengan satu jenis yang sama.
- **Baris (Rows):** sebuah baris (masukan atau rekaman data) merupakan sekumpulan data yang berhubungan.
- **Redundancy:** menyimpan data dua kali secara redundant untuk membuat sistem berjalan lebih cepat.
- **Primary Key:** key yang bersifat unik. Sebuah nilai key tidak dapat digunakan dua kali dalam satu tabel.
- **Foreign Key:** merupakan penghubung antara dua tabel.

- **Compound Key:** atau disebut juga composite key merupakan key yang terdiri dari beberapa kolom.
- **Indeks:** merupakan indeks dalam database yang menyerupai indeks pada buku.
- **Integritas referensial:** digunakan untuk memastikan nilai foreign selalu mengacu pada suatu baris yang ada.

Mengenal NetBeans IDE

NetBeans merupakan IDE (*Integrated Development Environment*) untuk membuat aplikasi dengan Java, PHP, C, C++, dan HTML5. Secara garis besar, NetBeans IDE bekerja menyerupai Microsoft Visual Studio maupun Dreamweaver dalam konteks sebagai aplikasi yang memiliki lingkungan kerja lengkap untuk membangun aplikasi lain.

NetBeans dimulai pada tahun 1996 sebagai Xelfi yang merupakan proyek IDE pada Universitas Charles di Praha. Pada tahun 1997, Roman Stanek mengomersialkan IDE NetBeans dan kemudian dibeli oleh Sun Microsystems pada tahun 1999. Pada tahun 2010, Sun (dan NetBeans) diakuisisi oleh Oracle.

NetBeans bukanlah satu-satunya IDE yang dapat digunakan untuk menulis bahasa pemrograman Java. Selain NetBeans, mungkin Anda pernah mendengar Eclipse. Namun di dalam buku ini, kita akan memanfaatkan NetBeans sebagai tool utama mempelajari Java dan database.

NetBeans merupakan IDE open-source yang dapat Anda gunakan secara gratis. Untuk men-download-nya, Anda dapat mengunjungi situs <https://netbeans.org/downloads/>.

NetBeans NetBeans IDE NetBeans Platform Plugins Docs & Support Community Partners Search 

HOME / Download

NetBeans IDE 8.0.1 Download

8.0 | 8.0.1 | Development | Archive

Email address (optional): IDE Language: Platform: Note: Greyed out technologies are not supported for this platform.

Subscribe to newsletters: Monthly Weekly NetBeans can contact me at this address

Supported technologies * NetBeans IDE Download Bundles

	Java SE	Java EE	C/C++	HTML5 & PHP	All
NetBeans Platform SDK	•	•			•
Java SE	•	•			•
Java FX	•	•			•
Java EE		•			•
Java ME				•	•
HTML5		•		•	•
Java Card™ 3 Connected			•		•
C/C++					•
Groovy					•
PHP				•	•
Bundled servers					
GlassFish Server Open Source Edition 4.1		•			•
Apache Tomcat 8.0.9	•				•

[Download](#) [Download](#) [Download](#) [Download](#) [Download](#)
Free, 90 MB Free, 185 MB Free, 63 MB Free, 63 MB Free, 204 MB

Halaman download NetBeans

###

pustaka-indo.blogspot.com

BAB 02

Instalasi Java dan NetBeans

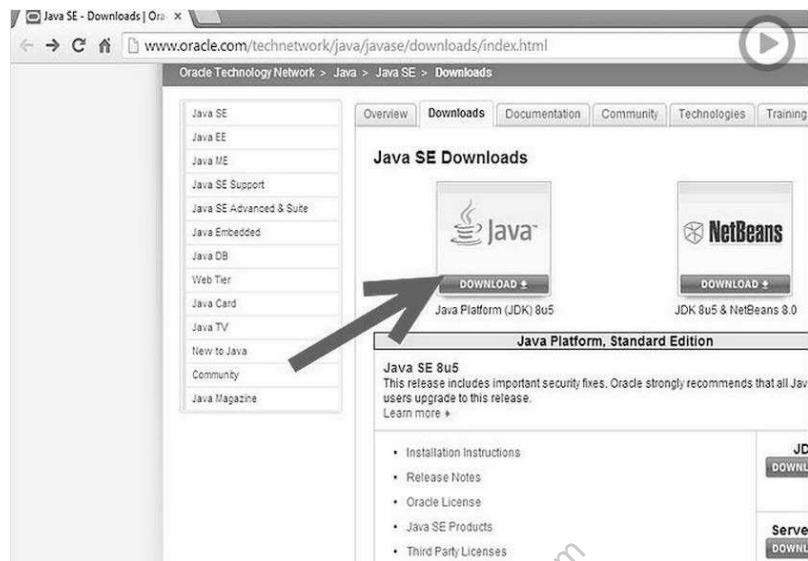
Salah satu alasan mengapa penetrasi Java dan NetBeans cukup luas di kalangan masyarakat, terutama kaum terpelajar dan mahasiswa, adalah karena kedua tool tersebut didistribusikan secara gratis. Jadi untuk memulai pemrograman Java, Anda tidak perlu mengeluarkan uang di muka. Cukup download Java dan NetBeans dari situs resmi yang menyediakannya, dan Anda sudah bisa membuat aplikasi berbasis Java dan database.

Di dalam bab ini, kita akan memulai pembahasan tentang bahasa pemrograman, database, dan NetBeans dengan proses instalasi.

Instalasi Java

Tool pertama yang perlu di-download dan diinstal adalah Java. Berikut ini langkah-langkah untuk menginstal Java.

1. Download JDK Java versi terbaru dari situs **oracle.com**.



Men-download JDK versi terbaru

2. Setelah proses download selesai, klik-ganda file installer untuk membuka jendela instalasi.
3. Klik tombol **Next** untuk mulai menginstal.



Mulai menginstal JDK

4. Klik **Next** sampai muncul jendela proses penginstalan.



Proses penginstalan sedang berlangsung

5. Setelah proses penginstalan selesai, klik **Close** untuk menutup jendela instalasi Java.



Klik Close untuk menutup jendela penginstalan

Menginstal NetBeans

Tool kedua yang akan diinstal adalah NetBeans. Pada bagian ini akan dijelaskan langkah-langkah instalasi NetBeans pada tiga platform yang berbeda, yaitu:

- Microsoft Windows
- Macintosh OS
- Linux Ubuntu

Windows

NetBeans didistribusikan secara gratis bagi pengguna MS Windows. Berikut langkah-langkah untuk menginstal NetBeans pada sistem operasi MS Windows.

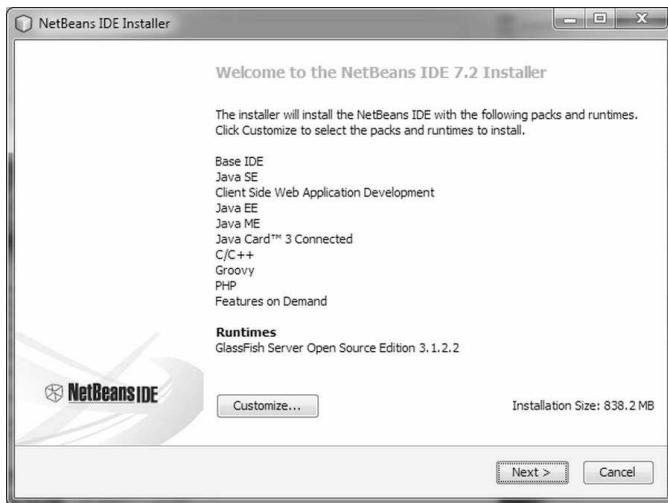
1. Download installer NetBeans pada situs <http://netbeans.org/downloads/>.

The screenshot shows the NetBeans IDE 8.0.1 Download page. At the top, there are fields for Email address (optional), IDE Language (English), Platform (Windows), and checkboxes for Monthly and Weekly newsletters, and a note about unsupported technologies. Below this is a table titled 'Supported technologies *' with columns for Java SE, Java EE, C/C++, HTML5 & PHP, and All. The table lists various technologies like NetBeans Platform SDK, Java SE, Java FX, Java EE, Java ME, HTML5, Java Card, C/C++, Groovy, and PHP, each with a dot indicating support. At the bottom, there are five 'Download' buttons corresponding to the technology categories.

Supported technologies *	Java SE	Java EE	C/C++	HTML5 & PHP	All
NetBeans Platform SDK	●	●			●
Java SE	●	●			●
Java FX	●	●			●
Java EE		●			●
Java ME		●			●
HTML5		●		●	●
Java Card™ 3 Connected					●
C/C++			●		●
Groovy					●
PHP				●	●
Bundled servers					
GlassFish Server Open Source Edition 4.1	●				●
Apache Tomcat 8.0.9	●				●

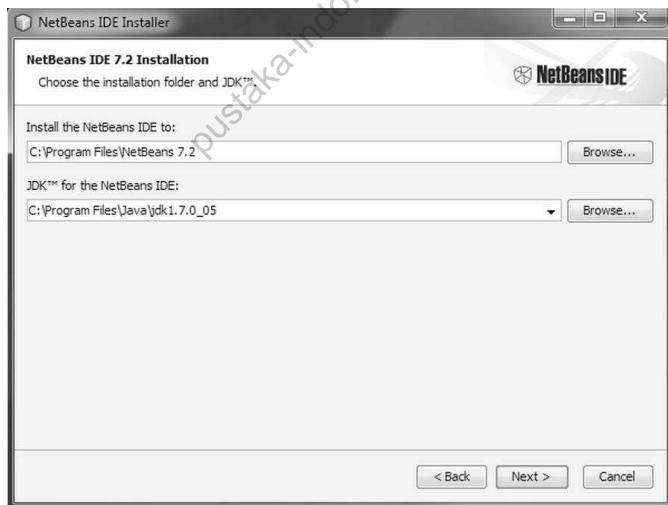
Halaman download NetBeans

2. Klik-ganda file installer untuk memulai proses instalasi.



Jendela penginstalan NetBeans

3. Klik Next dan pilih Accept the license agreement.
4. Kemudian tentukan direktori tepat instalasi NetBeans dilakukan.



Mengatur direktori penginstalan NetBeans

5. Beberapa saat kemudian akan terlihat jendela berisi rangkuman proses penginstalan yang ditentukan.



Rangkuman pengaturan penginstalan

6. Klik **Install** untuk mulai menginstal NetBeans. Tunggu sampai proses instalasi selesai dengan sempurna.
7. Setelah proses instalasi selesai, klik **Finish** untuk menutup jendela instalasi.

Mac OS

Apabila menggunakan Mac OS, maka Anda bisa mengunduh NetBeans untuk platform sistem operasi buatan Apple tersebut. Berikut ini langkah-langkah untuk menginstal NetBeans pada Mac OS.

1. Pertama-tama, download installer NetBeans pada situs <http://netbeans.org/downloads/>.
2. Lantas, jangan lupa untuk memilih platform **Mac OS**.

NetBeans IDE 8.0.1 Download

8.0 | 8.0.1 | Development | Archive

Email address (optional):

Subscribe to newsletters: Monthly Weekly NetBeans can contact me at this address

IDE Language: English Platform: Mac OS X

Note: Greyed out technologies: Windows, Linux (x86/x64), Mac OS X, OS Independent Zip

Supported technologies *	Java SE	Java EE	C/C++	HTML5 & PHP	All
NetBeans Platform SDK	•	•			•
Java SE	•	•			•
Java FX	•	•			•

NetBeans IDE Download Bundles

Men-download installer NetBeans untuk Mac OS

3. Klik-ganda file **Disk Image (.dmg)** yang telah di-download dari situs NetBeans di atas.
4. Klik-ganda “**NetBeans x.x.mpkg**” untuk menjalankan jendela instalasi.
5. Pada jendela penginstalan, atur pilihan-pilihan yang muncul sesuai dengan kebutuhan Anda.
6. Setelah proses instalasi selesai, maka akhiri dengan melakukan eject terhadap file .dmg.

Linux

Berikut ini langkah-langkah untuk menginstal NetBeans pada Linux Ubuntu:

1. Jika Anda merupakan pengguna Linux, maka download installer NetBeans pada situs <http://netbeans.org/downloads/>.
2. Pilih platform **Linux**.

NetBeans IDE 8.0.1 Download

8.0 | 8.0.1 | Development | Archive

Email address (optional):

Subscribe to newsletters: Monthly Weekly NetBeans can contact me at this address

IDE Language: English Platform: Linux (x86/x64)

Note: Greyed out technologies: Windows, Linux (x86/x64), Mac OS X, OS Independent Zip

Supported technologies *	Java SE	Java EE	C/C++	HTML5 & PHP	All
NetBeans Platform SDK	•	•			•
Java SE	•	•			•
Java FX	•	•			•
Java EE		•			•

NetBeans IDE Download Bundles

Men-download installer NetBeans untuk Linux Ubuntu

3. Atur file .sh yang telah di-download menjadi executable dan jalankan file tersebut.
4. Buka Terminal dan ketik penggalan script di bawah ini:

```
$ cd ~/Downloads  
$ chmod a+x netbeans-7.x-ml-javase-linux.sh  
    // Mengatur executable untuk semua (a+x)  
$ ./netbeans-7.x-ml-javase-linux.sh  
    // Menjalankan installer
```

5. Pada jendela instalasi, atur pilihan-pilihan yang muncul sesuai dengan kebutuhan Anda.
6. Untuk menjalankan NetBeans, jalankan script “netbeans” pada direktori bin NetBeans:

```
$ cd netbeans-bin-directory  
$ ./netbeans
```

BAB 03

Dasar-Dasar Pemrograman Java

Program Java dapat didefinisikan sebagai sebuah kumpulan objek-objek yang saling berkomunikasi dengan cara memanggil method-method yang dimiliki masing-masing objek.

Berikut ini penjelasan singkat mengenai kelas, objek, method, dan variabel instance yang ada di lingkungan Java:

- **Objek:** Objek memiliki state (keadaan) dan behavior (perilaku). Contoh: Seekor anjing memiliki keadaan -warna, nama, jenis, dan juga perilaku -menggonggong, makan, tidur. Sebuah objek adalah instance dari kelas.
- **Kelas:** Sebuah kelas dapat didefinisikan sebagai sebuah template/blue print yang mendeskripsikan perilaku/keadaan yang didukung oleh objek merupakan tipe kelas tersebut.
- **Method:** Method pada dasarnya adalah sebuah perilaku. Sebuah kelas dapat memiliki banyak method. Penulisan logika, manipulasi data dan eksekusi dari aksi lainnya semuanya dilakukan di dalam method.
- **Variabel Instance:** Setiap objek memiliki variabel uniknya sendiri. Keadaan sebuah objek dibuat berdasarkan nilai-nilai yang dimasukkan dalam variabel instance-nya.

Membuat Aplikasi ‘Halo Dunia’

Umumnya, aplikasi pertama yang dibuat oleh para pemula di bidang pemrograman adalah aplikasi ‘Halo Dunia’. Aplikasi tersebut sangatlah sederhana. Ketika dijalankan, aplikasi tersebut hanya menampilkan

tulisan ‘Halo Dunia’ di layar monitor. Dengan menggunakan Java, Anda bisa membuat aplikasi seperti itu dengan mudah.

Berikut ini contoh kode sederhana untuk menampilkan Halo Dunia.

```
public class ProgramJavaPertama
{
    /* Ini adalah program java pertama.
     * Ini akan menampilkan output 'Halo Dunia'
     */

    public static void main(String[] args)
    {
        System.out.println("Halo Dunia");
        // menampilkan Halo Dunia
    }
}
```

Untuk menyimpan, compile, dan menjalankan program tersebut, lakukan langkah-langkah berikut:

1. Buka editor teks. Untuk membuat script sederhana di atas, Anda bisa memanfaatkan **Notepad** (apabila menggunakan Windows).
2. Tuliskan kode program di atas menggunakan Notepad tersebut (kali ini belum memanfaatkan NetBeans).
3. Tekan tombol **Ctrl+S** dan simpan kode di atas ke dalam sebuah file dengan nama: **ProgramJavaPertama.java**.

TIP:

Di dalam textbox Filename Notepad, ketik “ProgramJavaPertama.java” (gunakan tanda kutip agar file tidak tersimpan dengan ekstensi .txt).

4. Buka jendela command prompt. Jika menggunakan MS Windows, klik tombol **Start > Run** dan ketik **cmd**.
5. Pergi ke direktori di mana Anda menyimpan kode di atas.
6. Ketik perintah berikut: ‘ javac ProgramJavaPertama.java ’ dan tekan **Enter** untuk melakukan compile.
7. Lanjutkan dengan mengetik ‘ java ProgramJavaPertama.java ’ untuk menjalankan program. Dalam waktu singkat, ‘ Halo Dunia ’ akan ditampilkan.

```
C :> javac ProgramJavaPertama.java
C :> java ProgramJavaPertama
HaloDunia
```

Sintaks Dasar Java

Sama seperti bahasa dan script pemrograman lainnya, Java memiliki beragam sintaks atau semacam “aturan main dalam menulis kode pemrograman”.

Berikut ini hal-hal yang harus diperhatikan dalam pemrograman Java:

- **Case Sensitive:** Java bersifat *case sensitive*, yang berarti identifier Halo dan halo memiliki arti yang berbeda.
- **Nama Kelas:** Untuk semua nama kelas, huruf pertama harus merupakan huruf kapital. Contoh: KelasJavaPertama.
- **Nama Method:** Semua nama method harus diawali dengan huruf kecil. Contoh: public void namaMethodSaya().
- **Nama File Program:** Nama file program harus sama dengan nama kelas.
- **public static void main(String args[]):** Pemrosesan program Java dimulai dari method main, yang merupakan bagian yang harus ada dalam setiap program Java.

Identifier dalam Java

Semua komponen Java memiliki nama. Nama-nama yang digunakan untuk kelas, variabel, dan method disebut *identifier*. Berikut ini hal-hal yang perlu diperhatikan mengenai identifier:

- Semua identifier harus diawali dengan huruf (A - Z atau a - z), karakter \$ atau karakter _ (underscore).
- Setelah karakter pertama, identifier dapat berisi kombinasi karakter apa saja.
- Sebuah kata kunci tidak dapat digunakan sebagai identifier.
- Identifier bersifat case sensitive.

Modifier dalam Java

Seperti pada bahasa pemrograman lainnya, Anda dapat memodifikasi kelas, method dan sebagainya dengan menggunakan modifier.

Ada dua kategori modifier dalam Java:

- Modifier Akses: default, public, protected, private
- Modifier Non-akses: final, abstract, strictfp

Variabel dalam Java

Ketika Anda bekerja menggunakan variabel, maka cermati bahwa terdapat beberapa jenis variabel dalam Java:

- Variabel lokal
- Variabel kelas (variabel static)
- Variabel instance (variabel non-static)

Array dalam Java

Array adalah objek yang menyimpan beberapa variabel dengan jenis yang sama. Array akan dibahas secara detail pada bab berikutnya.

Enum dalam Java

Enum membatasi sebuah variabel untuk memiliki satu nilai dari nilai-nilai yang telah ditentukan sebelumnya. Nilai-nilai ini kemudian disebut enum.

Berikut ini contoh penggunaan enum untuk membatasi ukuran gelas di tempat penjualan jus segar. Ukuran gelas dibatasi menjadi kecil, sedang, dan besar.

```
Class JusSegar
{
    enum UkuranJusSegar{ KECIL, SEDANG, BESAR }
    UkuranJusSegar ukuran;
}

public class TesJusSegar
{
```

```

public static void main(String args[])
{
    JusSegar jus = new JusSegar();
    jus.ukuran = JusSegar.UkuranJusSegar.SEDANG ;
}
}

```

Kata Kunci dalam Java

Ketika ingin membuat konstanta, variabel, atau nama identifier, cermatilah bahwa ada beberapa kata yang dilarang untuk digunakan. Umumnya, kata-kata yang dilarang untuk digunakan sebagai nama konstanta, variabel, dan identifier disebut dengan kata kunci.

Berikut ini daftar kata kunci dalam Java. Kata kunci ini tidak dapat digunakan sebagai konstanta, variabel, atau nama identifier.

abstract	assert	boolean	break
byte	case	catch	char
class	const	continue	default
do	double	else	enum
extends	final	finally	float
for	goto	if	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		

Daftar kata kunci dalam Java

Menulis Komentar dalam Java

Jika Anda sering lupa dengan maksud setiap script yang dibuat, maka cobalah untuk menulis komentar di sekitar script tersebut. Java mendukung penggunaan komentar single-line dan multi-line untuk membantu mendokumentasikan maksud dari script yang ada di bawah atau

atas komentar tersebut. Selama ditulis dengan sintaks yang tepat, maka semua karakter di dalam komentar akan diabaikan oleh compiler Java.

```
public class ProgramJavaPertama
{
    /* Ini adalah program java pertama.
     * Ini akan menampilkan output 'Halo Dunia'
     * Ini adalah contoh komentar multi-line
     */

    public static void main(String[]args)
    {
        // Ini adalah contoh komentar single-line
        /* Ini juga adalah contoh komentar single-line */

        System.out.println("Halo Dunia");
        // menampilkan Halo Dunia
    }
}
```

Penggunaan Spasi dalam Java

Suatu baris kode yang hanya berisi spasi (baris kosong) akan diabaikan dan tidak akan di-compile oleh Java.

Pewarisan

Kelas-kelas dalam Java dapat mewarisi kelas-kelas yang lain. Pada dasarnya, jika Anda ingin membuat kelas baru yang membutuhkan kode-kode yang ada pada kelas lain, Anda dapat mewarisi kelas lain itu.

Konsep ini memungkinkan Anda untuk menggunakan field dan method yang sudah ada tanpa harus menulis kembali kode tersebut pada kelas yang baru.

Interface

Pada Java, interface dapat didefinisikan sebagai sebuah kontrak antara objek-objek tentang bagaimana berkomunikasi satu sama lain.

Objek dalam Java

Objek dalam Java dapat diandaikan seperti objek yang biasa Anda temui sehari-hari seperti Mobil, Anjing, Manusia, dan sebagainya.

Semua objek yang ada memiliki keadaan dan perilakunya masing-masing.

Misalnya seekor anjing, memiliki keadaan -nama, jenis, warna, dan perilaku -menggonggong, tidur, makan, berlari.

Kelas dalam Java

Sebuah kelas merupakan blue print/cetakan untuk membuat tiap-tiap objek.

Berikut ini contoh kelas Anjing:

```
public class Anjing
{
    String jenis;
    int umur;
    String warna;

    void menggonggong()
    {

    }

    void lapar()
    {

    }
}
```

Konstruktor

Setiap kelas memiliki konstruktor. Jika Anda tidak membuat konstruktor untuk sebuah kelas, compiler Java akan membuat konstruktor default secara otomatis.

Setiap kali sebuah objek baru dibuat, satu atau lebih konstruktor akan dijalankan. Aturan utama dari sebuah konstruktor adalah memiliki nama yang sama dengan nama kelas tersebut. Sebuah kelas dapat memiliki lebih dari satu konstruktor.

Berikut ini contoh konstruktor:

```
public class Anjing
{
    public Anjing()
```

```
{  
}  
  
public Anjing(String nama)  
{  
    // Konstruktor ini memiliki satu parameter, yaitu nama.  
}  
}
```

Membuat Objek

Objek dibuat dari sebuah kelas. Pada Java, kata kunci new digunakan untuk membuat objek baru.

Berikut ini contoh untuk membuat objek baru:

```
public class Anjing  
{  
    public Anjing(String nama)  
    {  
        // Konstruktor ini memiliki satu parameter, yaitu nama.  
        System.out.println("Nama yang diberikan:"+ nama );  
    }  
  
    public static void main(String[]args)  
    {  
        // Statemen berikut akan membuat objek anjingKu  
        Anjing anjingKu = new Anjing("tommy");  
    }  
}
```

Mengakses Variabel Instance dan Method

Variabel instance dan method dapat diakses melalui objek-objek yang dibuat.

Berikut ini langkah-langkah untuk mengakses variabel instance:

```
/* Pertama-tama membuat objek */  
ReferensiObjek = new Konstruktor();  
  
/* Memanggil variabel */  
ReferensiObjek.namaVariabel;  
  
/* Sekarang Anda dapat memanggil method dari kelas */  
ReferensiObjek.NamaMethod();
```

Berikut ini contoh penggunaannya:

```

public class Anjing
{
    int umurAnjing;

    public Anjing(String nama)
    {
        // Konstruktor ini memiliki satu parameter, yaitu nama.
        System.out.println("Nama yang diberikan:" + nama );
    }

    public void aturUmur(int umur )
    {
        umurAnjing = umur;
    }

    public int ambilUmur()
    {
        System.out.println("Umur anjing adalah :" + umurAnjing );
        return umurAnjing;
    }

    public static void main(String[] args)
    {
        /* Membuat objek */
        Anjing anjingKu = new Anjing("tommy");

        /* Memanggil method kelas untuk mengatur umur anjing */
        anjingKu.aturUmur(2);

        /* Memanggil method kelas untuk mengambil umur anjing */
        anjingKu.ambilUmur();

        /* Anda juga dapat mengakses variabel instance
         * sebagai berikut
         */
        System.out.println("Nilai Variabel :" +
                           anjingKu.umurAnjing );
    }
}

```

Jika program tersebut dijalankan, akan menghasilkan output berikut:

```

Nama yang diberikan:tommy
Umur anjing adalah :2
Nilai Variabel :2

```

Package Java

Saat Anda membuat suatu aplikasi dengan Java, Anda dapat menggunakan package yang berisi kelas-kelas yang sudah berisi kode-kode yang sesuai dengan kebutuhan Anda.

Untuk menggunakannya Anda perlu menyertakan semua kelas yang ada pada suatu direktori. Berikut ini contoh penggunaannya:

```
import java.io.*;
```

Tipe Data dalam Java

Variabel adalah lokasi-lokasi dalam memori untuk menampung nilai-nilai. Ini berarti saat Anda membuat variabel, Anda menyiapkan ruang penampungan dalam memori.

Berdasarkan tipe data dari sebuah variabel, sistem operasi mengalokasikan memori dan menentukan tipe nilai apa yang dapat ditampung dalam memori tersebut.

Tipe Data Primitif

Berikut ini tipe data primitif dalam Java.

byte

- Tipe data byte merupakan integer 8-bit yang bersifat signed.
- Nilai minimum adalah -128.
- Nilai maksimum adalah 127.
- Nilai default adalah 0.
- Tipe data byte digunakan untuk menghemat ruang pada array yang besar.
- Contoh: byte a = 100, byte b = -50.

short

- Tipe data short merupakan integer 16-bit yang bersifat signed.
- Nilai minimum adalah -32,768.
- Nilai maksimum adalah 32,767.
- Nilai default adalah 0.

- Tipe data short juga dapat menghemat ruang seperti pada tipe data byte.
- Contoh: short s = 10000, short r = -20000.

int

- Tipe data int merupakan integer 32-bit yang bersifat signed.
- Nilai minimum adalah - 2,147,483,648.
- Nilai maksimum adalah 2,147,483,647.
- Nilai default adalah 0.
- Tipe data int digunakan secara umum untuk menampung nilai-nilai integral.
- Contoh: int a = 100000, int b = -200000.

long

- Tipe data long merupakan integer 64-bit yang bersifat signed.
- Nilai minimum adalah - 9,223,372,036,854,775,808.
- Nilai maksimum adalah 9,223,372,036,854,775,807.
- Nilai default adalah 0L.
- Tipe data long digunakan untuk menampung nilai-nilai dengan jangkauan nilai yang lebih dari tipe data int.
- Contoh: int a = 100000L, int b = -200000L.

float

- Tipe data float merupakan nilai desimal 32-bit presisi tunggal.
- Float biasanya digunakan untuk menghemat ruang pada array besar yang berisi nilai-nilai desimal.
- Nilai default adalah 0.0f.
- Tipe data float tidak pernah digunakan untuk nilai-nilai seperti nilai mata uang.
- Contoh: float f1 = 234.5f

double

- Tipe data double merupakan nilai desimal 64-bit presisi ganda.
- Double biasanya digunakan sebagai tipe data default untuk nilai desimal.
- Nilai default adalah 0.0d.
- Tipe data double tidak pernah digunakan untuk nilai-nilai presisi seperti nilai mata uang.
- Contoh: double d1 = 123.4

boolean

- Tipe data Boolean merepresentasikan satu bit informasi.
- Hanya ada dua nilai yang mungkin: true (benar) dan false (salah).
- Tipe data ini digunakan sebagai penanda kondisi benar/salah.
- Nilai default adalah false.
- Contoh: boolean satu = true;

char

- Tipe data char merupakan karakter Unicode 16-bit tunggal.
- Nilai minimum adalah '\u0000'.
- Nilai maksimum adalah '\uffff'.
- Tipe data char digunakan untuk menampung karakter apa saja.
- Contoh: char hurufA ='A'.

Tipe Data Referensi

- Variabel referensi dibuat menggunakan konstruktor yang didefinisikan dalam kelas. Variabel ini digunakan untuk mengakses objek-objek. Variabel ini dideklarasikan sebagai tipe tertentu yang tidak dapat diubah. Sebagai contoh Karyawan, Anjing, dan sebagainya.

- Objek kelas dan berbagai tipe variabel array merupakan tipe data referensi.
- Nilai default adalah null.
- Variabel referensi dapat digunakan untuk mengacu pada objek dengan tipe yang sama atau yang kompatibel.
- Contoh: Hewan hewan = new Hewan(“jerapah”);

Literal dalam Java

Bahasa pemrograman Java mendukung beberapa escape sequence untuk literal String dan char.

Notasi	Karakter yang Direpresentasikan
\n	Baris baru
\r	Awal baris
\f	Halaman baru
\b	Backspace
\s	Spasi
\t	Tab
\"	Tanda petik dua
'	Tanda petik
\\"	Garis miring
\ddd	Karakter octal
\uxxxx	Karakter UNICODE hexadesimal

Daftar escape sequence

Tipe Variabel dalam Java

Setiap variabel dalam Java memiliki tipe spesifik, yang menentukan ukuran dan layout memori; jangkauan nilai yang dapat disimpan; dan operasi-operasi yang dapat dijalankan terhadap variabel.

Anda harus mendeklarasikan semua variabel sebelum digunakan. Berikut ini bentuk dasar deklarasi variabel:

```
tipe data variabel [ = nilai][, variabel [= nilai] ...] ;
```

Berikut ini adalah contoh deklarasi dan inisialisasi variabel dalam Java:

```
int a, b, c; // Mendeklarasikan tiga int, a, b, dan c.  
int a = 10, b = 10; // Contoh inisialisasi.  
byte B = 22; // Inisialisasi variabel B dengan tipe byte.  
double pi = 3.14159; // Deklarasi dan inisialisasi nilai pi.  
char a = 'a'; // variabel char a diinisialisasi dengan  
// nilai 'a'.
```

Variabel Lokal

- Variabel lokal dideklarasikan di dalam method, konstruktor, atau blok.
- Variabel lokal dibuat saat method, konstruktor atau blok mulai dijalankan dan akan dihapus saat selesai dijalankan.
- Modifier akses tidak dapat digunakan untuk variabel lokal.
- Variabel lokal hanya dapat digunakan di dalam method, konstruktor, atau blok tempat pendeklarasiannya.
- Tidak ada nilai default untuk variabel lokal sehingga variabel lokal harus dideklarasikan dan diinisialisasi sebelum digunakan.

Berikut ini contoh variabel lokal umur yang didefinisikan di dalam method umurAnj() dan lingkup penggunaannya hanya di dalam method tersebut.

```
public class Tes  
{  
    public void umurAnj()  
    {  
        int umur = 0;  
        umur = umur + 7;  
        System.out.println("Umur anjing adalah : " + umur);  
    }  
  
    public static void main(String args[])  
    {  
        Tes tes = new Tes(); tes.umurAnj();  
    }  
}
```

Variabel Instance

- Variabel instance dideklarasikan di dalam kelas, tetapi di luar method, konstruktor, atau blok.
- Saat sebuah objek dibuat, tempat untuk setiap variabel instance disiapkan dalam memori.
- Variabel instance dibuat saat sebuah objek dibuat dengan menggunakan kata kunci ‘new’ dan dihapus saat objek tersebut dihapus.
- Variabel instance dapat dideklarasikan dalam kelas sebelum atau sesudah penggunaan.
- Modifier akses dapat digunakan pada variabel instance.
- Variabel instance dapat digunakan oleh semua method, konstruktor, dan blok di dalam kelas.
- Variabel instance mempunyai nilai default: untuk tipe numerik o, untuk Boolean false dan untuk referensi objek adalah null.

Berikut ini contoh penggunaannya:

```
import java.io.*;  
  
public class Karyawan  
{  
    // variabel instance ini dapat digunakan oleh semua subkelas  
    // (kelas anak).  
    public String nama;  
  
    // variabel gaji hanya dapat digunakan dalam kelas Karyawan.  
    private double gaji;  
  
    // Variabel nama diinisialisasi dalam konstruktor.  
    public Karyawan (String namaKar)  
    {  
        nama = namaKar;  
    }  
  
    // Memasukkan nilai dalam variabel gaji.  
    public void aturGaji(double gajiKar)  
    {  
        gaji = gajiKar;  
    }  
  
    // Method ini menampilkan informasi karyawan.  
    public void tampilKar()  
    {  
        System.out.println("nama : " + nama );  
    }  
}
```

```

        System.out.println("gaji :" + gaji);
    }

    public static void main(String args[])
    {
        Karyawan karSatu = new Karyawan("Ransika");
        karSatu.aturGaji(1000);
        karSatu.tampilKar();
    }
}

```

Variabel Kelas/Static

- Variabel kelas/static dideklarasikan dengan kata kunci static di dalam kelas, tetapi di luar method, konstruktor, atau blok.
- Hanya ada satu salinan dari variabel kelas per kelas.
- Variabel static biasanya dideklarasikan sebagai konstanta. Variabel konstanta nilainya tetap dan tidak berubah.
- Variabel static dibuat ketika program dimulai dan dihapus saat program berhenti.
- Nilai defaultnya sama dengan variabel instance.
- Variabel static dapat diakses dengan memanggil nama kelas .NamaKelas.NamaVariabel.
- Saat mendeklarasi variabel kelas dengan public static final, maka nama variabel (konstanta) dituliskan semua dalam huruf kapital. Jika variabel static tidak dideklarasikan dengan public final, sintaks penamaannya sama dengan variabel instance dan lokal.

Berikut ini contoh penggunaan variabel kelas:

```

import java.io.*;

public class Karyawan
{
    // variabel gaji adalah variabel private static
    private static double gaji;

    // DEPARTMENT adalah konstanta
    public static final String DEPARTMENT = "Pengembangan ";

    public static void main(String args[])
    {
        gaji = 1000;
    }
}

```

```
        System.out.println(DEPARTMENT+"gaji rata-ratanya:"+gaji);  
    }  
}
```

Modifier dalam Java

Modifier adalah kata kunci yang ditambahkan pada definisi variabel untuk menentukan artinya. Berikut ini daftar modifier dalam bahas pemrograman Java.

Modifier Akses

Java menyediakan beberapa modifier akses untuk mengatur level akses untuk kelas, variabel, method dan konstruktor.

Berikut ini empat level akses:

- Dapat diakses oleh package, yang merupakan level akses default. Tidak memerlukan modifier.
- Dapat diakses hanya oleh kelas (private).
- Dapat diakses oleh semua (public).
- Dapat diakses oleh package dan semua subkelas (protected).

Untuk menggunakan modifier, Anda menambahkan kata kunci pada definisi kelas, method, atau variabel. Modifier berada di bagian awal statemen.

Berikut ini contohnya:

```
public class namaKelas  
{  
    // ...  
}  
  
private boolean penanda;  
static final double minggu = 9.5;  
protected static final int LEBARKOTAK = 42;  
  
public static void main(String[] arguments)  
{  
    // isi method  
}
```

Modifier Akses Default - tanpa kata kunci

Variabel atau method yang dideklarasikan tanpa kata kunci dapat diakses oleh semua kelas pada package yang sama.

Berikut ini contohnya:

```
String version ="1.5.1";  
  
boolean processOrder()  
{  
    return true;  
}
```

Modifier Akses Privat - private

Method, variabel dan konstruktor yang dideklarasikan dengan kata kunci private hanya dapat diakses di dalam kelas tempat pendeklarasiannya.

Berikut ini contohnya:

```
public class Logger  
{  
    private String format;  
  
    public String getFormat()  
    {  
        return this.format;  
    }  
  
    public void setFormat(String format)  
    {  
        this.format = format;  
    }  
}
```

Pada contoh di atas, variabel format dalam kelas Logger adalah privat, jadi kelas lain tidak dapat mengakses atau memodifikasi nilainya secara langsung.

Modifier Akses Publik - public

Sebuah kelas, method, konstruktor, interface, dan lain-lain yang dideklarasikan public dapat diakses oleh kelas-kelas lainnya.

Berikut ini contohnya:

```
public static void main(String[] arguments)
{
    // ...
}
```

Modifier Akses Protected - protected

Variabel, method dan konstruktor yang dideklarasikan protected di dalam superclass dapat diakses hanya oleh subclass pada package yang berbeda atau kelas yang ada dalam package yang sama.

Berikut ini contoh kelas yang menggunakan protected untuk memungkinkan kelas turunannya melakukan override method openSpeaker():

```
class AudioPlayer
{
protected boolean openSpeaker(Speaker sp)
{
    // implementasi
}
}

class StreamingAudioPlayer
{
    boolean openSpeaker(Speaker sp)
    {
        // implementasi
    }
}
```

Pada contoh di atas, jika Anda mendefinisikan method openSpeaker() sebagai private, maka method itu tidak dapat diakses oleh kelas lain selain AudioPlayer.

Kontrol Akses dan Pewarisan

Berikut ini aturan untuk method yang diwariskan:

- Method yang dideklarasikan public pada superclass juga harus public dalam semua subclass.
- Method yang dideklarasikan protected pada superclass harus protected atau public dalam subclass; tidak boleh private.

- Method yang dideklarasikan tanpa kontrol akses (tanpa modifier) dapat dideklarasikan lebih private dalam subclass.
- Method yang dideklarasikan private tidak diwariskan. Sehingga tidak ada aturannya.

Modifier Non Akses

Java menyediakan beberapa modifier non-akses.

- Modifier *static* untuk membuat kelas, method dan variabel.
- Modifier *final* untuk menetapkan implementasi dari kelas, method, dan variabel.
- Modifier *abstract* untuk membuat kelas dan method abstract.
- Modifier *synchronized* dan *volatile*, yang digunakan untuk thread.

Operator dalam Java

Berikut ini daftar operator yang dapat digunakan dalam bahasa pemrograman Java.

Operator Aritmetika

Operator aritmetika digunakan pada ekspresi matematik seperti pada operasi aljabar. Berikut ini daftar operator aritmetika:

Diasumsikan bahwa variabel A bernilai 10 dan B bernilai 20, maka:

Operator	Deskripsi	Contoh
+	Penjumlahan: Menambahkan nilai-nilai yang ada di kedua sisi operator.	$A + B$ hasilnya 30
-	Pengurangan: Mengurangkan nilai operand di sebelah kiri dengan nilai operand di sebelah kanan.	$A - B$ hasilnya -10

*	Perkalian: Mengalikan nilai-nilai yang ada di kedua sisi operator.	A * B hasilnya 200
/	Pembagian: Membagi nilai operand di sebelah kiri dengan nilai operand di sebelah kanan.	B / A hasilnya 2
%	Modulus: Melakukan pembagian nilai operand di sebelah kiri dengan nilai operand di sebelah kanan dan mengambilkan sisa nilainya.	B % A hasilnya 0
++	Peningkatan: Menambahkan 1 pada nilai operand.	B++ hasilnya 21
--	Penurunan: Mengurangkan 1 dari nilai operand.	B-- hasilnya 19

Operator aritmetika dalam Java

Operator Relasional

Berikut ini operator relasional yang didukung dalam Java:

Diasumsikan bahwa variabel A bernilai 10 dan B bernilai 20, maka:

Operator	Deskripsi	Contoh
==	Memeriksa apakah nilai kedua operand sama atau tidak, jika sama maka kondisi bernilai benar.	(A == B) adalah tidak benar
!=	Memeriksa apakah nilai kedua operand sama atau tidak, jika tidak sama maka kondisi bernilai benar.	(A != B) adalah benar
>	Memeriksa apakah nilai operand di sebelah kiri lebih dari nilai operand di sebelah kanan, jika ya maka kondisi bernilai benar.	(A > B) adalah tidak benar
<	Memeriksa apakah nilai operand di sebelah kiri kurang dari nilai	(A < B) adalah benar

	operand di sebelah kanan, jika ya maka kondisi bernilai benar.	
\geq	Memeriksa apakah nilai operand di sebelah kiri lebih dari atau sama dengan nilai operand di sebelah kanan, jika ya maka kondisi bernilai benar.	(A \geq B) adalah tidak benar
\leq	Memeriksa apakah nilai operand di sebelah kiri kurang dari atau sama dengan nilai operand di sebelah kanan, jika ya maka kondisi bernilai benar.	(A \leq B) adalah benar

Operator relasional dalam Java

Operator Logika

Berikut ini daftar operator logika. Diasumsikan bahwa nilai Boolean variabel A adalah true dan B adalah false, maka:

Operator	Deskripsi	Contoh
$\&\&$	Operator logika AND. Jika kedua operand bukan nol, maka kondisi bernilai benar.	(A $\&\&$ B) adalah tidak benar
$\ $	Operator logika OR. Jika ada operand yang bukan nol, maka kondisi bernilai benar.	(A $\ $ B) adalah benar
!	Operator logika NOT. Digunakan untuk membalik keadaan logika dari operan. Jika kondisi bernilai benar maka operator NOT akan membuatnya menjadi tidak benar.	!(A $\&\&$ B) adalah benar

Operator logika dalam Java

Operator Assignment

Berikut ini daftar operator assignment dalam Java:

Operator	Deskripsi	Contoh
=	Memasukkan nilai pada operand di sebelah kanan ke dalam operand di sebelah kiri.	$C = A + B$ akan memasukkan nilai $A + B$ ke dalam C
+=	Menjumlahkan nilai operand di sebelah kiri dengan nilai operand di sebelah kanan dan memasukkan hasilnya ke dalam operand di sebelah kiri.	$C += A$ hasilnya sama dengan $C = C + A$
-=	Mengurangkan nilai operand di sebelah kiri dengan nilai operand di sebelah kanan dan memasukkan hasilnya ke dalam operand di sebelah kiri.	$C -= A$ hasilnya sama dengan $C = C - A$
*=	Mengalikan nilai operand di sebelah kiri dengan nilai operand di sebelah kanan dan memasukkan hasilnya ke dalam operand di sebelah kiri.	$C *= A$ hasilnya sama dengan $C = C * A$
/=	Membagi nilai operand di sebelah kiri dengan nilai operand di sebelah kanan dan memasukkan hasilnya ke dalam operand di sebelah kiri.	$C /= A$ hasilnya sama dengan $C = C / A$
%=	Menggunakan operasi modulus terhadap kedua operand dan memasukkan hasilnya ke dalam operand di sebelah kiri.	$C %= A$ hasilnya sama dengan $C = C \% A$

Operator assignment dalam Java

Operator Kondisional (?:)

Operator kondisional juga disebut operator *ternary*. Operator ini terdiri atas tiga operand dan digunakan untuk mengevaluasi ekspresi Boolean. Tujuan dari operator ini adalah untuk menentukan nilai mana yang akan dimasukkan ke dalam variabel.

Berikut ini syntax penulisan operator ternary:

```
variabel x =(ekspresi)? nilai jikabenar: nilai jikalah
```

Berikut ini contoh penggunaannya:

```
public class Tes
{
    public static void main(String args[])
    {
        int a , b;

        a =10;
        b =(a ==1)?20:30;
        System.out.println("Nilai b adalah : "+ b );

        b =(a ==10)?20:30;
        System.out.println("Nilai b adalah : " + b );
    }
}
```

Kode tersebut akan menghasilkan output berikut:

```
Nilai b adalah : 30
Nilai b adalah : 20
```

Operator instanceof

Operator ini digunakan hanya untuk variabel referensi objek. Operator memeriksa apakah objek merupakan tipe tertentu.

```
(Objek variabel referensi) instanceof (tipe kelas/interface)
```

Jika objek yang direferensikan oleh variabel di sebelah kiri operator merupakan tipe kelas/interface di sebelah kanan, maka hasilnya adalah benar. Berikut ini contohnya:

```
String nama = "James";
boolean hasil = nama instanceof String;
// Ini akan bernilai benar karena nama bertipe String
```

Pada contoh berikut tetap bernilai benar karena objek yang dibandingkan kompatibel dengan tipe di sebelah kanan:

```
class Kendaraan{}  
  
public class Mobil extends Kendaraan  
{  
    public static void main(String args[])  
    {  
        Kendaraan a = new Mobil();  
        boolean hasil = a instanceof Mobil;  
        System.out.println(hasil);  
    }  
}
```

Kode tersebut akan menghasilkan output berikut:

```
true
```

Preseden Operator-Operator Java

Berikut ini urutan preseden operator-operator Java:

Kategori	Operator	Asosiasi
Postfix	() [] . (operator titik)	Kiri ke kanan
Unary	++ -- ! ~	Kanan ke kiri
Multiplicative	* / %	Kiri ke kanan
Additive	+ -	Kiri ke kanan
Shift	>>>><<	Kiri ke kanan
Relasional	>>= <<=	Kiri ke kanan
Persamaan	== !=	Kiri ke kanan
Bitwise AND	&	Kiri ke kanan
Bitwise XOR	^	Kiri ke kanan
Bitwise OR		Kiri ke kanan
Logical AND	&&	Kiri ke kanan

Logical OR		Kiri ke kanan
Kondisional	?:	Kanan ke kiri
Assignment	= += -= *= /= %= >>= <<= &= ^= =	Kanan ke kiri
Koma	,	Kiri ke kanan

Urutan preseden operator dalam Java

Kontrol Perulangan dalam Java

Ada beberapa metode perulangan yang bisa digunakan dalam bahasa pemrograman Java.

Perulangan while

Perulangan while adalah struktur kontrol yang memungkinkan Anda mengulangi suatu proses dengan jumlah perulangan tertentu.

Berikut ini sintaks dari perulangan while:

```
while(ekspresi_Boolean)
{
    // Statement
}
```

Selama ekspresi Boolean bernilai benar, maka statemen dalam tubuh while akan terus dieksekusi.

Berikut ini contoh penggunaannya:

```
public class Test
{
    public static void main(String args[])
    {
        int x =10;

        while( x <20)
        {
            System.out.print("nilai x : "+ x );
            x++;

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

```
    }  
}
```

Kode tersebut akan menghasilkan output berikut:

```
nilai x :10  
nilai x :11  
nilai x :12  
nilai x :13  
nilai x :14  
nilai x :15  
nilai x :16  
nilai x :17  
nilai x :18  
nilai x :19
```

Perulangan do...while

Perulangan do...while sama seperti pengulangan while, tetapi perulangan do...while pasti akan dieksekusi minimal satu kali.

Berikut sintaksnya:

```
do  
{  
    // Statement  
}  
while(ekspresi_boolean);
```

Berikut ini contoh penggunaannya:

```
public class Test  
{  
    public static void main(String args[])  
    {  
        int x =10;  
  
        do  
        {  
            System.out.print("nilai x : "+ x );  
            x++;  
  
            System.out.print("\n");  
        }  
        while( x <20);  
    }  
}
```

Kode tersebut akan menghasilkan output berikut:

```
nilai x :10
nilai x :11
nilai x :12
nilai x :13
nilai x :14
nilai x :15
nilai x :16
nilai x :17
nilai x :18
nilai x :19
```

Perulangan for

Perulangan for adalah struktur kontrol repetitif yang memungkinkan Anda untuk menjalankan proses dengan jumlah perulangan tertentu (jumlah perulangan sudah diketahui sebelumnya).

Berikut ini sintaksnya:

```
for(inisialisasi;ekspresi_boolean; update)
{
    // Statement
}
```

Berikut ini aliran proses dalam perulangan for:

- Bagian inisialisasi dieksekusi pertama kali dan hanya sekali. Bagian ini memungkinkan Anda untuk mendeklarasikan dan menginisialisasi variabel kontrol perulangan.
- Setelah itu, ekspresi Boolean dievaluasi. Jika bernilai benar, statemen dalam tubuh for dieksekusi. Jika tidak, statemen tidak dieksekusi dan proses berlanjut pada bagian setelah perulangan for.
- Setelah statemen dalam for dieksekusi, aliran proses kembali pada bagian update. Statemen update ini memungkinkan Anda untuk meng-update variabel kontrol dalam for.
- Ekspresi Boolean kemudian dievaluasi lagi. Jika benar, statemen dalam perulangan for kembali dieksekusi, dan dilanjutkan ke bagian update. Jika tidak, perulangan for selesai dan proses berlanjut pada bagian setelah perulangan for.

Berikut ini contoh penggunaannya:

```
public class Tes
{
    public static void main(String args[])
    {
        for(int x =10; x <20; x = x+1)
        {
            System.out.print("nilai x : " + x );
            System.out.print("\n");
        }
    }
}
```

Berikut ini hasil outputnya:

```
nilai x :10
nilai x :11
nilai x :12
nilai x :13
nilai x :14
nilai x :15
nilai x :16
nilai x :17
nilai x :18
nilai x :19
```

Perulangan for... Versi Baru dalam Java

Perulangan for... versi baru disediakan pada Java versi 5 dan biasanya digunakan untuk array.

Berikut ini sintaksnya:

```
for(deklarasi : ekspresi)
{
    // Statement
}
```

- Deklarasi: berisi variabel blok yang baru dideklarasi, yang kompatibel dengan jenis elemen array yang diakses.
- Ekspresi: ekspresi ini dievaluasi berdasarkan array yang diakses.

Berikut ini contoh penggunaannya:

```
public class Tes
{
    public static void main(String args[])
    }
```

```

{
    int[] angka ={10,20,30,40,50};

    for(int x : angka )
    {
        System.out.print(x);
        System.out.print(",");
    }

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

    String[] namanama ={"James", "Larry", "Tom", "Lacy"};

    for(String nama : namanama )
    {
        System.out.print( nama );
        System.out.print(",");
    }
}

```

Berikut ini hasil output dari kode di atas:

10,20,30,40,50,
James,Larry,Tom,Lacy,

Break

Kata kunci break digunakan untuk menghentikan eksekusi perulangan.

Berikut ini contoh penggunaannya:

```

public class Tes
{
    public static void main(String args[])
    {
        int[] angka ={10,20,30,40,50};

        for(int x : angka)
        {
            if(x ==30)
            {
                break;
            }

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

```

Berikut ini output dari kode tersebut:

10
20

Continue

Kata kunci continue digunakan untuk melompat pada iterasi selanjutnya pada perulangan.

- Pada perulangan for, kata kunci menyebabkan aliran proses melompat langsung pada bagian update.
- Pada perulangan while, aliran proses melompat langsung pada bagian ekspresi Boolean.

Berikut ini contoh penggunaannya:

```
public class Tes
{
    public static void main(String args[])
    {
        int[] angka ={10,20,30,40,50};

        for(int x : angka)
        {
            if( x ==30)
            {
                continue;
            }

            System.out.print( x );
            System.out.print( "\n");
        }
    }
}
```

Kode tersebut akan menghasilkan output berikut:

10
20
40
50

Pengambilan Keputusan dalam Java

Ada beberapa metode pengambilan keputusan dalam bahasa pemrograman Java.

Statemen if

Stemen if terdiri dari sebuah ekspresi Boolean yang diikuti dengan satu statemen atau lebih.

Berikut ini sintaks untuk statemen if:

```
if(ekspresi_boolean)
{
    // Statemen akan dieksekusi jika ekspresi Boolean
    // bernilai benar
}
```

Berikut ini contoh penggunaannya:

```
public class Tes
{
    public static void main(String args[])
    {
        int x =10;

        if( x <20)
        {
            System.out.print("Ini adalah statemen if");
        }
    }
}
```

Kode tersebut akan menghasilkan output berikut:

```
Ini adalah statemen if
```

Statemen if...else

Statemen if dapat diikuti dengan statemen else opsional, yang di-eksekusi jika ekspresi Boolean bernilai salah.

Berikut ini sintaksnya:

```
if(ekspresi_boolean)
{
    // Dieksekusi jika ekspresi Boolean bernilai benar
}
else
{
    // Dieksekusi jika ekspresi Boolean bernilai salah
}
```

Berikut ini contoh penggunaannya:

```
public class Tes
{
    public static void main(String args[])
    {
        int x =30;
        if(x <20)
        {
            System.out.print("Ini adalah statemen if");
        }
        else
        {
            System.out.print("Ini adalah statemen else");
        }
    }
}
```

Kode tersebut akan menghasilkan output berikut:

```
Ini adalah statemen else
```

Statemen if...else if...else

Statemen if dapat diikuti dengan statemen else if...else opsional.

Saat menggunakan statemen if, else if, else ada beberapa hal yang harus Anda perhatikan:

- Sebuah statemen if dapat memiliki nol atau satu statemen else dan harus digunakan setelah statemen else if.
- Sebuah statemen if dapat memiliki nol atau banyak statemen else if dan harus digunakan sebelum statemen else.
- Setelah sebuah statemen else if dieksekusi, statemen else if atau else yang lain tidak akan diperiksa (akan langsung melompat pada kode setelah else terakhir).

Berikut ini sintaksnya:

```
if(ekspresi_boolean1)
{
    // Dieksekusi jika ekspresi boolean 1 bernilai benar
}
elseif(ekspresi_boolean2)
{
    // Dieksekusi jika ekspresi boolean 2 bernilai benar
}
```

```
elseif(ekspresi_boolean3)
{
    // Dieksekusi jika ekspresi boolean 3 bernilai benar
}
else
{
    // Dieksekusi jika tidak ada kondisi di atas yang
    // bernilai benar.
}
```

Berikut ini contoh penggunaannya:

```
public class Tes
{
    public static void main(String args[])
    {
        int x =30;

        if( x ==10)
        {
            System.out.print("Nilai X adalah 10");
        }
        elseif( x ==20)
        {
            System.out.print("Nilai X adalah 20");
        }
        elseif( x ==30)
        {
            System.out.print("Nilai X adalah 30");
        }
        else
        {
            System.out.print("Ini adalah statemen else");
        }
    }
}
```

Kode tersebut akan menghasilkan output berikut:

```
Nilai X adalah 30
```

Statemen if...else Bersarang

Anda dapat memasukkan statemen if else ke dalam statemen if else lainnya.

Berikut ini sintaks dari statemen if else bersarang:

```
if(ekspresi_boolean1)
{
    // Dieksekusi jika ekspresi boolean 1 bernilai benar
```

```
if(ekspresi_boolean2)
{
    // Dieksekusi jika ekspresi boolean 2 bernilai benar
}
```

Berikut ini contoh penggunaannya:

```
public class Tes
{
    public static void main(String args[])
    {
        int x =30;
        int y =10;

        if( x ==30)
        {
            if( y ==10)
            {
                System.out.print("X = 30 dan Y = 10");
            }
        }
    }
}
```

Hasil dari kode tersebut sebagai berikut:

```
X =30 dan Y =10
```

Statemen Switch

Statemen switch memungkinkan sebuah variabel diperiksa kesamaannya dengan daftar nilai-nilai yang ada. Setiap nilai dalam daftar disebut *case*.

Berikut ini sintaks statemen switch:

```
switch(expression)
{
    case nilai :
        // Statemen
        break; // opsional
    case nilai :
        // Statemen
        break; // opsional
    // Anda dapat menggunakan beberapa statemen case.
    default: // optional
        // Statements
}
```

Berikut ini aturan penggunaan statemen switch:

- Variabel yang digunakan dalam switch hanya boleh berupa byte, short, int, atau char.
- Anda dapat menggunakan beberapa case di dalam switch. Setiap case diikuti dengan nilai yang akan dibandingkan dan tanda titik dua (:).
- Nilai case harus sama tipe datanya dengan variabel dalam switch dan harus berupa konstanta atau literal.
- Ketika variabel yang dibandingkan sama dengan dengan case, statemen pada case tersebut akan dieksekusi sampai pada kata kunci break.
- Ketika sampai pada kata kunci break, switch berhenti dieksekusi, dan aliran proses melompat pada kode sesudah statemen switch.
- Tidak semua case memerlukan break. Jika tidak ada break, aliran proses akan terus dilanjutkan pada case selanjutnya sampai pada break.
- Statemen switch dapat memiliki case default, yang diletakkan pada akhir statemen. Default dijalankan jika tidak ada case yang dijalankan.

Berikut ini contoh penggunaannya:

```
public class Tes
{
    public static void main(String args[])
    {
        char nilai = args[0].charAt(0);

        switch(nilai)
        {
            case 'A':
                System.out.println("Sempurna!");
                break;
            case 'B':
            case 'C':
                System.out.println("Bagus");
                break;
            case 'D':
                System.out.println("Lumayan");
            case 'F':
                System.out.println("Coba lagi");
                break;
        }
    }
}
```

```
    default:  
        System.out.println("Nilai tidak valid");  
    }  
  
    System.out.println("Nilai Anda "+ nilai);  
}
```

Kode tersebut menghasilkan output sebagai berikut:

```
$ java Tes a  
Nilai tidak valid  
Nilai Anda a  
  
$ java Tes A  
Sempurna!  
Nilai Anda A  
  
$ java Tes C  
Bagus  
Nilai Anda C
```

pustaka-indo.blogspot.com

BAB 04

Menyiapkan Database

Sekarang, kita akan memanfaatkan database dalam pemrograman Java. Database merupakan sistem yang membantu kita menyimpan data-data user untuk banyak kepentingan.

Ada beberapa system database yang dapat Anda gunakan, seperti MySQL, SQL Server, Oracle, dan lain-lain. Salah satu database yang sering digunakan adalah database MySQL yang dapat Anda gunakan secara gratis. Berikut ini cara untuk menginstal database MySQL dalam bentuk bundle program.

WAMP, MAMP, dan LAMP

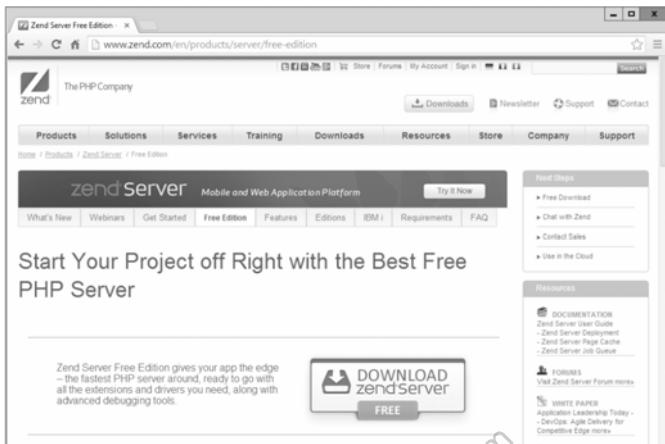
WAMP, MAMP, dan LAMP adalah singkatan dari “Windows, Apache, MySQL, dan PHP”, “Mac, Apache, MySQL, dan PHP”, dan “Linux, Apache, MySQL, dan PHP”. Ketiga server development ini menyediakan pengaturan lengkap yang diperlukan untuk membangun sebuah halaman web yang dinamis.

Dengan server development tersebut, Anda dapat dengan mudah menginstal Apache, MySQL, dan PHP sekaligus dalam satu paket. Anda hanya perlu men-download dan menginstal satu kali saja untuk mendapatkan fitur lengkap Apache, MySQL, dan PHP.

Menginstal WAMP di Windows

Ada beberapa server WAMP yang tersedia untuk Anda gunakan, dan dalam buku ini Anda akan menggunakan edisi gratis dari Zend Server.

Anda dapat men-download-nya melalui situs <http://tinyurl.com/zendfree> seperti terlihat pada gambar berikut.



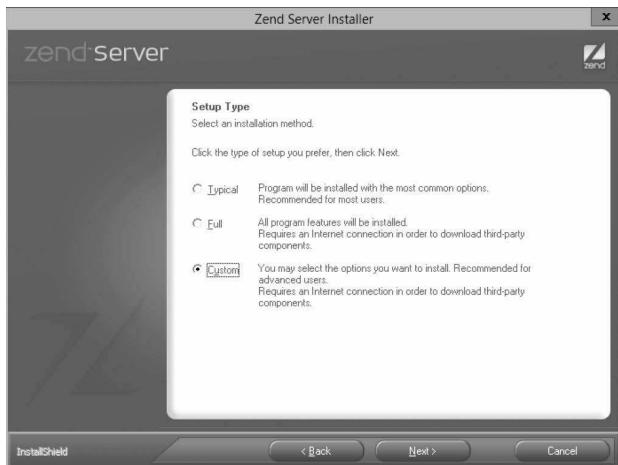
Situs download Zend Server Free Edition

Setelah Anda men-download Zend Server Free Edition, jalankan file installer untuk mulai menginstal.



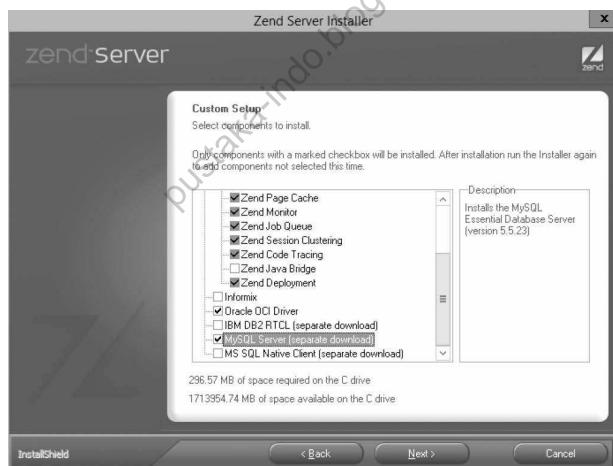
Jendela utama installer Zend Server

Klik Next dan pilih Accept the license agreement untuk melanjutkan ke jendela Setup Type. Pilih Custom sehingga server MySQL dapat diinstall juga.



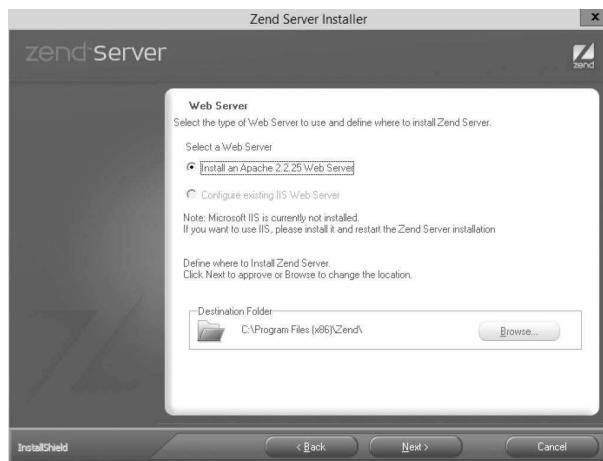
Memilih pilihan Custom

Pada jendela Custom Setup, gulir pilihan menu ke bawah dan pastikan bahwa MySQL Server sudah dicentang. Setelah itu klik Next.



Centang pilihan MySQL Server

Pilih Install an Apache Web Server, kemudian klik Next.



Menginstal web server Apache

Atur nilai default Web Server Port menjadi 80, dan Zend Server Interface Port menjadi 10081. Kemudian klik Next.



Mengatur port server

Setelah itu klik Install pada jendela Installation Settings.



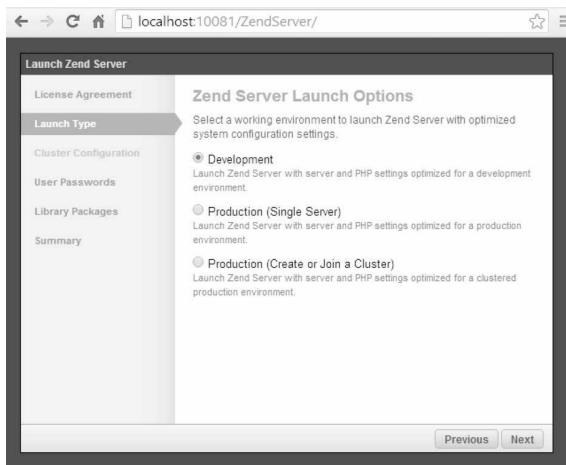
Jendela Installation Settings

Setelah instalasi selesai, Anda dapat langsung menggunakan software dengan mengklik Finish. Browser default Anda akan terbuka. Pilih kotak I have read and agree untuk melanjutkan.



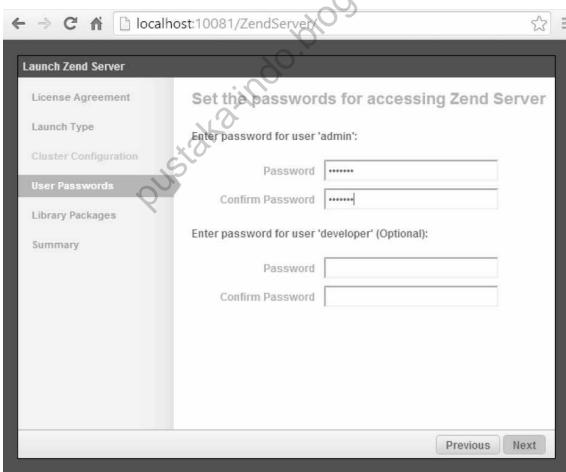
Memilih I have read....

Pada tab Launch Type pilih Development kemudian klik Next.



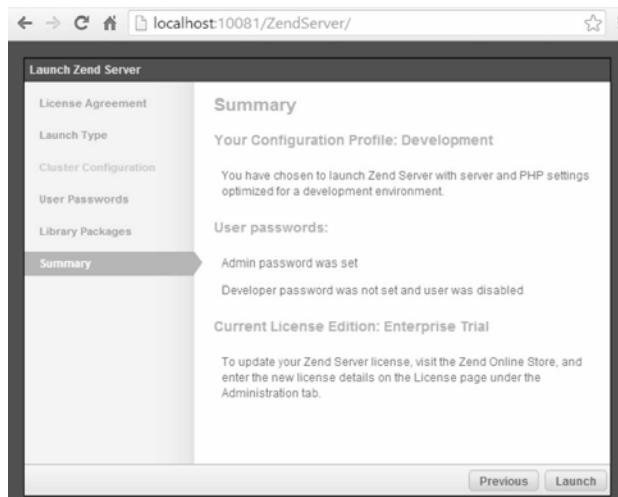
Memilih pilihan Development

Atur password untuk admin pengguna. Anda tidak perlu mengatur password untuk developer pengguna.



Masukkan password untuk admin dua kali

Klik Next pada jendela Library Packages kemudian klik Launch untuk menyelesaikan proses instalasi.



Klik Submit untuk menyelesaikan pengaturan

Setelah itu browser akan menampilkan Dashboard server.



Dashboard Zend Server

Menginstal MAMP di Mac OS X

Zend Server Free Edition juga dapat digunakan pada OS X. Anda dapat men-download-nya melalui situs <http://tinyurl.com/zendfree>.

The screenshot shows the Zend Server download page. The sidebar on the left has 'Products' selected. The main content area displays the Zend Server logo and a 'Downloads' section. It lists three versions of Zend Server for Mac OS X:

Product	Version	Format/Size	Notes	MD5 Checksum
Zend Server (PHP 5.5)	6.3.0	[DMG] 287.11 MB	Release Notes	1753261813246a2cd216abbff688f
Zend Server (PHP 5.4)	6.3.0	[DMG] 285.97 MB	Release Notes	cd9c72093e0fdead25819ef01d8517
Zend Server (PHP 5.3)	6.3.0	[DMG] 283.17 MB	Release Notes	56a0908a13611b8d9ab0328e1ac542a

Men-download Zend Server untuk Mac OS X

Setelah installer selesai di-download, klik-ganda file .dmg.



Klik-ganda Zend Server untuk menginstal

Jendela instalasi akan muncul. Klik Continue, akan muncul instruksi penginstalan. Klik Continue lagi untuk melanjutkan.



Jendela instalasi

Pilih lokasi penginstalan, klik Install dan masukkan password.



Mulai menginstal Zend Server

Setelah selesai menginstal, carilah program ZendServer di folder Applications dan klik-ganda untuk menyelesaikan pengaturan. Browser akan terbuka dan Anda dapat mengikuti langkah-langkah yang sama dengan instalasi WAMP pada Windows.

Konfigurasi MySQL

Installer MAMP tidak dilengkapi dengan command untuk memulai dan menutup MySQL server, sehingga Anda harus melakukannya secara manual dengan membuka Terminal dan menuliskan command berikut:

```
sudo nano /usr/local/zend/bin/zendctl.sh
```

Setelah memasukkan password, editor teks Nano akan terbuka. Pin-dahkan kursor ke bawah sampai Anda menemukan baris `MySQL_EN="false"`. Ubah nilainya menjadi `true`.

Setelah itu terus gulir ke bawah sampai Anda menemukan baris berikut:

```
case $1 in
    "start")
        ...
        $0 start-apache %
```

Tepat di bawah baris tersebut, tambahkan baris berikut:

```
$0 start-MySQL %
```

Kemudian terus gulir ke bawah sampai Anda menemukan bagian berikut:

```
"stop")
...
$0 stop-apache %
```

Tepat di bawah baris tersebut, tambahkan baris berikut:

```
$0 stop-MySQL %
```

Setelah itu tekan Ctrl-X untuk keluar dari mode edit, tekan Y, dan kemudian tekan Return untuk menyimpan file.

Menginstal LAMP di Linux

Anda dapat men-download Zend Server Free Edition melalui situs <http://tinyurl.com/zendfree>. Setelah itu Anda dapat menginstalnya dengan langkah-langkah yang sama dengan cara penginstalan WAMP dan MAMP pada bagian sebelumnya.

pustaka-indo.blogspot.com

###

pustaka-indo.blogspot.com

BAB 05

Pengenalan SQL

SQL merupakan singkatan dari *Structured Query Language*. Bahasa ini didasarkan pada bahasa Inggris dan digunakan dalam database seperti MySQL, Oracle, dan Microsoft SQL Server. Bahasa ini dirancang untuk memungkinkan permintaan sederhana dari database melalui perintah seperti:

```
SELECT judul FROM publikasi WHERE penulis = 'Rendra';
```

Sebuah database berisi satu tabel atau lebih, yang masing-masing berisi catatan atau baris. Dalam baris-baris ini terdapat berbagai kolom atau field yang berisi data. Tabel berikut menunjukkan contoh database yang berisi lima publikasi yang dirinci berdasarkan penulis, judul, jenis, dan tahun publikasi.

Author	Title	Type	Year
Mark Twain	The Adventures of Tom Sawyer	Fiction	1876
Jane Austen	Pride and Prejudice	Fiction	1811
Charles Darwin	The Origin of Species	Non-Fiction	1856
Charles Dickens	The Old Curiosity Shop	Fiction	1841
William Shakespeare	Romeo and Juliet	Play	1594

Contoh database sederhana

Istilah-Istilah Penting dalam Database

Database

Semua kontainer untuk menampung koleksi data.

Tabel

Sebuah subkontainer di dalam database yang menampung data aktual.

Baris

Sebuah rekaman data tunggal di dalam tabel, yang dapat berisi beberapa field.

Kolom

Nama field dalam baris.

Perintah-Perintah dalam SQL

Pada bagian ini akan dibahas mengenai perintah-perintah (query) SQL. Pertama-tama Anda harus memperhatikan poin-poin berikut:

- Perintah SQL dan kata-kata kunci adalah case-insensitive. CREATE, create, dan CrEaTe mempunyai arti yang sama. Namun, untuk kejelasan, gaya penulisan yang direkomendasikan adalah menggunakan huruf kapital.
- Nama tabel adalah case-sensitive di Linux dan OS X, tapi case-insensitive pada Windows. Jadi untuk tujuan portabilitas, Anda harus selalu memilih satu gaya dan secara konsisten menggunakananya. Gaya yang dianjurkan adalah menggunakan huruf kecil untuk tabel.

Berikut daftar perintah SQL.

Command	Action
ALTER	Alter a database or table
BACKUP	Backup a table
\c	Cancel input
CREATE	Create a database
DELETE	Delete a row from a table
DESCRIBE	Describe a table's columns
DROP	Delete a database or table
EXIT (CTRL-C)	Exit
GRANT	Change user privileges
HELP (\h, \?)	Display help
INSERT	Insert data
LOCK	Lock table(s)
QUIT (\q)	Same as EXIT
RENAME	Rename a table
SHOW	List details about an object
SOURCE	Execute a file
STATUS (\s)	Display the current status
TRUNCATE	Empty a table
UNLOCK	Unlock table(s)
UPDATE	Update an existing record
USE	Use a database

Daftar perintah dalam SQL

Membuat Database

Gunakan perintah berikut untuk membuat database baru bernama publications:

```
CREATE DATABASE publications;
```

Perintah yang sukses akan memunculkan pesan - Query OK, 1 row affected (0.00 sec). Sekarang setelah Anda membuat database, Anda

akan bekerja dengan database itu dengan cara memasukkan perintah berikut:

```
USE publications;
```

Sekarang Anda akan melihat pesan Database changed.

Membuat Pengguna

Untuk membuat pengguna, gunakan perintah GRANT, yang dituliskan seperti contoh berikut (jangan memasukkan perintah ini, perintah ini hanya merupakan contoh dan tidak akan bekerja):

```
GRANT PRIVILEGES ON database.object TO 'username'@'hostname'  
IDENTIFIED BY 'password';
```

Contoh di atas cukup jelas, kecuali pada bagian database.object, yang mengacu pada database itu sendiri dan objek yang dikandungnya, seperti pada tabel berikut.

Arguments	Meaning
<code>*.*</code>	All databases and all their objects
<code>database.*</code>	Only the database called <i>database</i> and all its objects
<code>database.object</code>	Only the database called <i>database</i> and its object called <i>object</i>

Contoh parameter untuk perintah GRANT

Untuk membuat pengguna yang dapat mengakses hanya database publications dan semua objeknya, Anda dapat menuliskan perintah berikut (ganti jim dan mypasswd sesuai dengan pilihan Anda):

```
GRANT ALL ON publications.* TO 'jim'@'localhost'  
IDENTIFIED BY 'mypassword';
```

Yang harus Anda lakukan sekarang adalah memasukkan password ketika diminta. Anda juga dapat menempatkan password Anda segera setelah -p (tanpa spasi). Tapi hal ini dianggap sebagai praktik kurang baik, karena jika orang lain masuk ke sistem Anda, mungkin ada cara

bagi mereka untuk melihat perintah yang Anda masukkan dan mengetahui password Anda.

Membuat Tabel

Pada titik ini, Anda telah berhasil login ke dalam database dengan semua hak istimewa yang diberikan untuk database publications (atau database yang telah Anda buat), sehingga Anda siap untuk membuat tabel. Pastikan database yang tepat sedang digunakan dengan memasukkan perintah berikut (ganti publications dengan nama database Anda jika berbeda):

```
USE publications;
```

Setelah itu masukkan perintah ini:

```
CREATE TABLE klasik (
    penulis VARCHAR(128),
    judul VARCHAR(128),
    jenis VARCHAR(16),
    tahun CHAR(4)) ENGINE MyISAM;
```

Untuk memeriksa apakah tabel baru Anda telah dibuat, masukkan perintah berikut:

```
DESCRIBE klasik;
```

Tabel akan muncul seperti berikut:

Field	Type	Null	Key	Default	Extra
penulis	varchar(128)	YES		NULL	
judul	varchar(128)	YES		NULL	
jenis	varchar(16)	YES		NULL	
tahun	char(4)	YES		NULL	

4 rows in set (0.00 sec)

Perintah DESCRIBE dapat membantu proses debugging ketika Anda perlu memastikan bahwa Anda telah benar membuat tabel database. Anda juga dapat menggunakannya untuk melihat field atau nama kolom beserta jenis datanya masing-masing.

Berikut ini penjelasan masing-masing heading secara rinci:

Field

Nama masing-masing field atau kolom dalam sebuah tabel.

Type

Jenis data yang disimpan di dalam field.

Null

Apakah field diperbolehkan mengandung nilai NULL.

Key

SQL mendukung penggunaan key atau indeks, yang merupakan cara cepat untuk mencari data. Heading key menunjukkan jenis kunci apa (jika ada) yang telah diterapkan.

Default

Nilai default yang akan dimasukkan ke dalam field jika tidak ada nilai yang ditentukan ketika baris baru dibuat.

Extra

Informasi tambahan, seperti apakah field diatur menjadi auto-increment.

Tipe Data dalam SQL

Pada contoh membuat tabel dalam database, Anda mungkin telah memperhatikan bahwa ada tiga field yang bertipe data VARCHAR, dan satu bertipe CHAR. Istilah VARCHAR merupakan singkatan dari VARiable length CHARacter string, dan perintah ini menggunakan nilai numerik yang memberi tahu database panjang maksimal yang diperbolehkan bagi string yang tersimpan dalam field ini.

Tipe data ini sangat berguna untuk merencanakan ukuran database dan melakukan pencarian data dengan lebih mudah.

Field tahun, memiliki ukuran yang lebih mudah diprediksi, jadi Anda dapat menggunakan tipe data CHAR(4) yang lebih efisien. Parameter 4 memungkinkan pengguna untuk memasukkan empat byte data, dari tahun -999 sampai 9999.

CHAR

Tabel berikut menampilkan tipe data CHAR.

Data type	Bytes used	Examples
CHAR(<i>n</i>)	exactly <i>n</i> (< 256)	CHAR(5) "Hello" uses 5 bytes CHAR(57) "Goodbye" uses 57 bytes
VARCHAR(<i>n</i>)	up to <i>n</i> (< 65,536)	VARCHAR(7) "Morning" uses 7 bytes VARCHAR(100) "Night" uses 5 bytes

Tipe data CHAR

Semua tipe ini menggunakan parameter untuk menentukan panjang maksimum string yang diperbolehkan di dalam suatu field. Seperti yang ditunjukkan dalam tabel, masing-masing tipe memiliki jumlah maksimum byte yang dapat digunakan.

BINARY

Tipe data BINARY digunakan untuk menyimpan string byte penuh yang tidak memiliki karakter yang terkait. Sebagai contoh, Anda dapat menggunakan tipe data BINARY untuk menyimpan gambar GIF.

Data type	Bytes used	Examples
BINARY(<i>n</i>) or BYTE(<i>n</i>)	exactly <i>n</i> (< 256)	As CHAR but contains binary data
VARBINARY(<i>n</i>)	up to <i>n</i> (< 65,536)	As VARCHAR but for binary data

Tipe data BINARY

TEXT

Tipe data TEXT dan VARCHAR memiliki beberapa perbedaan:

- SQL versi 5.0.3 dan selanjutnya akan menghapus spasi di depan dan di belakang field berjenis VARCHAR.
- Field TEXT tidak dapat memiliki nilai default.
- SQL melakukan indeks hanya pada karakter n pertama dari kolom TEXT (nilai n ditentukan saat Anda membuat indeks).

Hal ini berarti bahwa VARCHAR adalah tipe data yang lebih baik dan lebih cepat untuk digunakan jika Anda perlu mencari seluruh isi field. Jika Anda tidak akan mencari lebih dari jumlah tertentu dari karakter awal pada suatu field, Anda mungkin sebaiknya menggunakan tipe data TEXT.

Data type	Bytes used	Attributes
TINYTEXT(<i>n</i>)	up to <i>n</i> (< 256)	Treated as a string with a character set
TEXT(<i>n</i>)	up to <i>n</i> (< 65,536)	Treated as a string with a character set
MEDIUMTEXT(<i>n</i>)	up to <i>n</i> (< 1.67e+7)	Treated as a string with a character set
LONGTEXT(<i>n</i>)	up to <i>n</i> (< 4.29e+9)	Treated as a string with a character set

Type data TEXT

BLOB

Istilah BLOB merupakan singkatan dari Binary Large OBject dan oleh karena itu, tipe data BLOB adalah tipe yang cocok untuk data biner yang berukuran lebih dari 65.536 byte. Perbedaan utama lain antara BLOB dan tipe data BINARY adalah bahwa BLOB tidak dapat memiliki nilai default.

Data type	Bytes used	Attributes
TINYBLOB(<i>n</i>)	up to <i>n</i> (< 256)	Treated as binary data—no character set
BLOB(<i>n</i>)	up to <i>n</i> (<= 65,536)	Treated as binary data—no character set
MEDIUMBLOB(<i>n</i>)	up to <i>n</i> (< 1.67e+7)	Treated as binary data—no character set
LONGBLOB(<i>n</i>)	up to <i>n</i> (< 4.29e+9)	Treated as binary data—no character set

Type data BLOB

NUMERIK

SQL mendukung berbagai tipe data numerik mulai dari byte tunggal sampai angka desimal presisi ganda. Meskipun memori field numerik dapat mencapai 8 byte, Anda disarankan untuk memilih tipe data

terkecil yang cukup untuk menangani nilai terbesar yang Anda harapkan. Sehingga database Anda akan menjadi kecil dan cepat diakses.

Tabel berikut berisi daftar tipe data numerik yang didukung oleh SQL dan rentang nilainya masing-masing. Istilah signed adalah angka dengan rentang dari nilai minus, 0, sampai nilai positif, sedangkan unsigned memiliki rentang nilai mulai dari 0 sampai nilai positif.

Data type	Bytes used	Minimum value		Maximum value	
		Signed	Unsigned	Signed	Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32,768	0	32,767	65,535
MEDIUMINT	3	-8.38e+6	0	8.38e+6	1.67e+7
INT or INTEGER	4	-2.15e+9	0	2.15e+9	4.29e+9
BIGINT	8	-9.22e+18	0	9.22e+18	1.84e+19
FLOAT	4	-3.40e+38	n/a	3.40e+38	n/a
DOUBLE or REAL	8	-1.80e+308	n/a	1.80e+308	n/a

Tipe data numerik

Untuk menentukan apakah tipe data numerik signed atau unsigned, Anda dapat menggunakan penanda UNSIGNED. Contoh berikut membuat tabel bernama namatabel dengan field di dalamnya bernama namafield dengan tipe data unsigned INTEGER:

```
CREATE TABLE namatabel (namafield INT UNSIGNED);
```

Ketika membuat field numerik, Anda juga dapat menggunakan angka opsional sebagai parameter, seperti ini:

```
CREATE TABLE namatabel (namafield INT(4) ZEROFILL);
```

Perintah ini menyebabkan angka apapun dengan lebar kurang dari empat karakter, diisi dengan satu atau lebih angka nol untuk membuat tampilan lebar field menjadi empat karakter.

Tanggal dan Waktu

Berikut adalah tipe data berkaitan dengan tanggal dan waktu yang terdapat dalam SQL:

Data type	Time/date format
DATETIME	'0000-00-00 00:00:00'
DATE	'0000-00-00'
TIMESTAMP	'0000-00-00 00:00:00'
TIME	'00:00:00'
YEAR	0000 (Only years 0000 and 1901–2155)

Tipe data tanggal dan waktu

Tipe data DATETIME dan TIMESTAMP ditampilkan dengan cara yang sama. Perbedaan utamanya adalah bahwa TIMESTAMP memiliki rentang yang sangat sempit (dari tahun 1970 sampai 2037), sedangkan DATETIME akan menampung hampir setiap tanggal yang Anda tentukan.

TIMESTAMP tetap sering digunakan, karena dengan tipe data ini SQL dapat menetapkan nilai secara otomatis untuk Anda. Jika Anda tidak menentukan nilai saat menambahkan sebuah baris, waktu saat itu akan secara otomatis dimasukkan. Anda juga dapat memerintahkan SQL memperbarui kolom TIMESTAMP setiap kali Anda mengubah baris.

Menambah Data ke dalam Tabel

Untuk menambahkan data ke dalam tabel, Anda dapat menggunakan perintah INSERT.

Berikut adalah contoh perintah untuk menambahkan data ke dalam tabel klasik:

```
INSERT INTO klasik(penulis, judul, jenis, tahun)
VALUES ('Mark Twain', 'The Adventures of Tom
Sawyer', 'Fiction', '1876');
INSERT INTO klasik(penulis, judul, jenis, tahun)
VALUES ('Jane Austen', 'Pride and
Prejudice', 'Fiction', '1811');
```

```

INSERT INTO klasik(penulis, judul, jenis, tahun)
    VALUES('Charles Darwin','The Origin of Species','Non-
        Fiction','1856');
INSERT INTO klasik(penulis, judul, jenis, tahun)
    VALUES('Charles Dickens','The Old Curiosity
        Shop','Fiction','1841');
INSERT INTO klasik(penulis, judul, jenis, tahun)
    VALUES('William Shakespeare','Romeo and
        Juliet','Play','1594');

```

Setelah setiap dua baris dijalankan, Anda akan melihat pesan Query OK. Setelah semua baris telah dimasukkan, masukkan perintah berikut, yang akan menampilkan isi tabel:

```
SELECT * FROM classics;
```

Tabel akan ditampilkan seperti pada gambar berikut:

```

C:\Windows\system32\cmd.exe
mysql> INSERT INTO classics(author, title, type, year)
-> VALUES('Charles Darwin','The Origin of Species','Non-Fiction','1856');
Query OK, 1 row affected <0.00 sec>

mysql> INSERT INTO classics(author, title, type, year)
-> VALUES('Charles Dickens','The Old Curiosity Shop','Fiction','1841');
Query OK, 1 row affected <0.00 sec>

mysql> INSERT INTO classics(author, title, type, year)
-> VALUES('William Shakespeare','Romeo and Juliet','Play','1594');
Query OK, 1 row affected <0.00 sec>

mysql> SELECT * FROM classics;
+-----+-----+-----+-----+
| author | title | type | year |
+-----+-----+-----+-----+
| Mark Twain | The Adventures of Tom Sawyer | Fiction | 1876 |
| Jane Austen | Pride and Prejudice | Fiction | 1811 |
| Charles Darwin | The Origin of Species | Non-Fiction | 1856 |
| Charles Dickens | The Old Curiosity Shop | Fiction | 1841 |
| William Shakespeare | Romeo and Juliet | Play | 1594 |
+-----+-----+-----+-----+
5 rows in set <0.00 sec>

mysql> 

```

Mengisi tabel klasik dan menampilkannya

Mengganti Nama Tabel

Anda dapat mengganti nama tabel seperti mengganti struktur atau informasi meta dalam tabel, yaitu dengan menggunakan perintah ALTER.

Sebagai contoh, untuk mengganti nama tabel dari klasik menjadi pre1900, Anda dapat menggunakan perintah berikut:

```
ALTER TABLE classics RENAME pre1900;
```

Mengganti Tipe Data Kolom

Untuk mengganti tipe data suatu kolom, Anda juga dapat menggunakan perintah ALTER.

Berikut ini adalah contoh perintah untuk mengubah tipe data kolom tahun:

```
ALTER TABLE klasik MODIFY tahun SMALLINT;
```

Menambah Kolom Baru

Untuk menambahkan kolom baru dalam tabel, Anda dapat menggunakan cara berikut:

```
ALTER TABLE klasik ADD halaman SMALLINT UNSIGNED;
```

Setelah kolom ditambahkan, tabel akan terlihat seperti ini:

```
C:\Windows\system32\cmd.exe
+-----+-----+-----+-----+
| author | varchar(128) | YES | NULL |
| title | varchar(128) | YES | NULL |
| type | varchar(16) | YES | NULL |
| year | smallint(6) | YES | NULL |
+-----+-----+-----+-----+
4 rows in set <0.01 sec>

mysql> ALTER TABLE classics ADD pages SMALLINT UNSIGNED;
Query OK, 5 rows affected <0.02 sec>
Records: 5 Duplicates: 0 Warnings: 0

mysql> DESCRIBE classics;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| author | varchar(128) | YES |   | NULL    |       |
| title | varchar(128) | YES |   | NULL    |       |
| type  | varchar(16)  | YES |   | NULL    |       |
| year  | smallint(6)  | YES |   | NULL    |       |
| pages | smallint(6) unsigned | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set <0.00 sec>

mysql>
```

Menambahkan kolom baru ke dalam tabel

Mengganti Nama Kolom

Anda dapat mengganti nama kolom dengan cara seperti ini:

```
ALTER TABLE klasik CHANGE jenis kategori VARCHAR(16);
```

Kolom jenis akan berganti nama menjadi kategori.

Menghapus Kolom

Untuk menghapus kolom, Anda dapat menggunakan perintah DROP.

```
ALTER TABLE klasik DROP halaman;
```

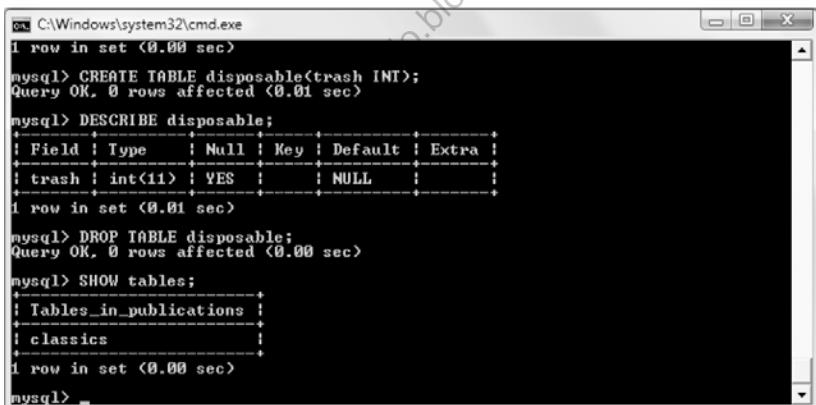
Kolom halaman dan semua isinya akan terhapus.

Menghapus Tabel

Untuk menghapus tabel, Anda dapat menggunakan perintah DROP. Pada contoh berikut, akan dibuat sebuah tabel baru yang nantinya akan dihapus.

```
CREATE TABLE disposable(trash INT);
DESCRIBE disposable;
DROP TABLE disposable;
SHOW tables;
```

Tabel disposable dan semua isinya akan terhapus.



```
C:\Windows\system32\cmd.exe
1 row in set <0.00 sec>

mysql> CREATE TABLE disposable;
Query OK, 0 rows affected <0.01 sec>

mysql> DESCRIBE disposable;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| trash | int(11) | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+
1 row in set <0.01 sec>

mysql> DROP TABLE disposable;
Query OK, 0 rows affected <0.00 sec>

mysql> SHOW tables;
+-----+
| Tables_in_publications |
+-----+
| classics                |
+-----+
1 row in set <0.00 sec>

mysql> _
```

Membuat, menampilkan, dan menghapus tabel

Query dalam Database SQL

Berikut query yang dapat digunakan untuk mengelola data di dalam database.

SELECT

Perintah SELECT digunakan untuk mengambil data dari tabel.

Sintaks dasar SELECT adalah sebagai berikut:

```
SELECT sesuatu FROM namatabel;
```

Untuk menampilkan semua data dalam tabel, Anda dapat menggunakan sintaks berikut:

```
SELECT * FROM namatabel;
```

Berikut ini dua contoh penggunaan SELECT:

```
SELECT penulis, judul FROM klasik;  
SELECT judul, isbn FROM klasik;
```

```
C:\Windows\system32\cmd.exe  
mysql> SELECT author,title FROM classics;  
+-----+-----+  
| author | title |  
+-----+-----+  
| Mark Twain | The Adventures of Tom Sawyer |  
| Jane Austen | Pride and Prejudice |  
| Charles Darwin | The Origin of Species |  
| Charles Dickens | The Old Curiosity Shop |  
| William Shakespeare | Romeo and Juliet |  
+-----+-----+  
5 rows in set <0.00 sec>  
  
mysql> SELECT title,isbn FROM classics;  
+-----+-----+  
| title | isbn |  
+-----+-----+  
| The Adventures of Tom Sawyer | 9781598184891 |  
| Pride and Prejudice | 9780502506286 |  
| The Origin of Species | 9780517123201 |  
| The Old Curiosity Shop | 9780099533474 |  
| Romeo and Juliet | 9780192814968 |  
+-----+-----+  
5 rows in set <0.00 sec>  
  
mysql>
```

Output dari dua statemen SELECT yang berbeda

DELETE

Untuk menghapus suatu baris dari tabel, gunakan perintah DELETE. Sintaksnya mirip dengan perintah SELECT dan memungkinkan Anda untuk mempersempit pemilihan satu baris atau beberapa baris yang akan dihapus menggunakan kualifikasi seperti WHERE dan LIMIT.

Berikut ini contoh penggunaan DELETE:

```
DELETE FROM klasik WHERE judul='Little Dorrit';
```

Contoh tersebut memerintahkan untuk menghapus semua baris yang kolom judulnya mengandung string Little Dorrit.

WHERE

Keyword WHERE memungkinkan Anda untuk mempersempit query dengan mengembalikan hanya data yang sesuai dengan ekspresi tertentu. Contoh sebelumnya mengembalikan hanya baris yang kolom judulnya sama persis dengan string Little Dorrit, menggunakan operator persamaan =. Berikut adalah contoh lain penggunaan WHERE dengan = :

```
SELECT penulis, judul FROM klasik WHERE penulis="Mark Twain";  
SELECT penulis, judul FROM klasik WHERE isbn="9781598184891 ";
```

Anda juga dapat melakukan pencocokan pola dalam pencarian Anda dengan menggunakan kualifikasi LIKE, yang memungkinkan pencarian pada sebagian string. Kualifikasi ini harus digunakan dengan karakter % sebelum atau setelah beberapa teks. Jika ditempatkan sebelum kata kunci, tanda % berarti "apa pun sebelum" dan jika ditempatkan setelah kata kunci itu berarti "apa pun setelah." Contoh berikut berisi tiga pertanyaan yang berbeda dalam penggunaan LIKE:

```
SELECT penulis, judul FROM klasik WHERE penulis LIKE  
    "%Charles%";  
SELECT penulis, judul FROM klasik WHERE judul LIKE "%Species";  
SELECT penulis, judul FROM klasik WHERE judul LIKE "%and%";
```

Berikut ini tampilan hasil pencarinya:

```
on C:\Windows\system32\cmd.exe
mysql> SELECT author.title FROM classics WHERE author LIKE "Charles%";
+-----+-----+
| author | title |
+-----+-----+
| Charles Darwin | The Origin of Species |
| Charles Dickens | The Old Curiosity Shop |
+-----+-----+
2 rows in set <0.00 sec>

mysql> SELECT author.title FROM classics WHERE title LIKE "%Species";
+-----+-----+
| author | title |
+-----+-----+
| Charles Darwin | The Origin of Species |
+-----+-----+
1 row in set <0.00 sec>

mysql> SELECT author.title FROM classics WHERE title LIKE "%and%";
```

author	title
Jane Austen	Pride and Prejudice
William Shakespeare	Romeo and Juliet

```
+-----+-----+
2 rows in set <0.00 sec>
```

Contoh penggunaan LIKE

LIMIT

Kualifikasi LIMIT memungkinkan Anda untuk memilih berapa banyak baris yang akan ditampilkan dalam query, dan baris mana dalam tabel untuk memulai pemilihan. Jika diberikan hanya sebuah parameter tunggal, query memberitahu SQL untuk memulai pemilihan dari awal. Jika Anda memberikan dua parameter, parameter pertama menunjukkan posisi awal pemilihan hasil yang akan ditampilkan, dan yang kedua menunjukkan berapa banyak hasil yang akan ditampilkan.

```
on C:\Windows\system32\cmd.exe
mysql> SELECT author.title FROM classics LIMIT 3;
+-----+-----+
| author | title |
+-----+-----+
| Mark Twain | The Adventures of Tom Sawyer |
| Jane Austen | Pride and Prejudice |
| Charles Darwin | The Origin of Species |
+-----+-----+
3 rows in set <0.00 sec>

mysql> SELECT author.title FROM classics LIMIT 1,2;
+-----+-----+
| author | title |
+-----+-----+
| Jane Austen | Pride and Prejudice |
| Charles Darwin | The Origin of Species |
+-----+-----+
2 rows in set <0.00 sec>

mysql> SELECT author.title FROM classics LIMIT 3,1;
+-----+-----+
| author | title |
+-----+-----+
| Charles Dickens | The Old Curiosity Shop |
+-----+-----+
```

Contoh penggunaan LIMIT

Berikut contoh penggunaan LIMIT:

```
SELECT penulis,judul FROM klasik LIMIT 3;
SELECT penulis,judul FROM klasik LIMIT 1,2;
SELECT penulis,judul FROM klasik LIMIT 3,1;
```

UPDATE ... SET

Konstruksi ini memungkinkan Anda untuk memperbarui isi field. Jika Anda ingin mengubah isi dari satu field atau lebih, Anda harus terlebih dahulu mempersempit hanya pada field yang akan diubah, dengan cara yang sama dengan perintah SELECT sebelumnya. Contoh berikut menunjukkan penggunaan UPDATE ... SET dalam dua cara yang berbeda.

```
UPDATE klasik SET penulis='Mark Twain (Samuel Langhorne
Clemens)' WHERE penulis='Mark Twain';

UPDATE classics SET category='Classic Fiction'
WHERE kategori='Fiction';
```

Tampilan hasil dari query tersebut adalah sebagai berikut:

```
C:\Windows\system32\cmd.exe
mysql>
mysql> UPDATE classics SET author='Mark Twain (Samuel Langhorne Clemens)'
-> WHERE author='Mark Twain';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE classics SET category='Classic Fiction'
-> WHERE category='Fiction';
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> SELECT author,category FROM classics;
+-----+-----+
| author          | category        |
+-----+-----+
| Mark Twain (Samuel Langhorne Clemens) | Classic Fiction
| Jane Austen      | Classic Fiction
| Charles Darwin   | Non-Fiction
| Charles Dickens  | Classic Fiction
| William Shakespeare | Play
+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Update kolom dalam tabel klasik

Saat melakukan update, Anda juga dapat menggunakan kualifikasi seperti LIMIT, ORDER BY, dan GROUP BY.

ORDER BY

ORDER BY mengurutkan hasil yang ditampilkan berdasarkan satu kolom atau lebih secara ascending atau descending. Berikut ini dua contoh query.

```
SELECT penulis,judul FROM klasik ORDER BY penulis;  
SELECT penulis,judul FROM klasik ORDER BY judul DESC;
```

Tampilannya sebagai berikut:

```
C:\Windows\system32\cmd.exe  
mysql> SELECT author,title FROM classics ORDER BY author;  
+-----+-----+  
| author | title |  
+-----+-----+  
| Charles Darwin | The Origin of Species  
| Charles Dickens | The Old Curiosity Shop  
| Jane Austen | Pride and Prejudice  
| Mark Twain <Samuel Langhorne Clemens> | The Adventures of Tom Sawyer  
| William Shakespeare | Romeo and Juliet  
+-----+-----+  
5 rows in set <0.00 sec>  
  
mysql> SELECT author,title FROM classics ORDER BY title DESC;  
+-----+-----+  
| author | title |  
+-----+-----+  
| Charles Darwin | The Origin of Species  
| Charles Dickens | The Old Curiosity Shop  
| Mark Twain <Samuel Langhorne Clemens> | The Adventures of Tom Sawyer  
| William Shakespeare | Romeo and Juliet  
| Jane Austen | Pride and Prejudice  
+-----+-----+  
5 rows in set <0.02 sec>  
mysql> _
```

Contoh penggunaan ORDER BY

GROUP BY

Dengan cara yang sama seperti ORDER BY, Anda dapat menge-lompokkan hasil dari query menggunakan GROUP BY. Misalnya, jika Anda ingin tahu berapa banyak publikasi yang ada untuk masing-masing kategori dalam tabel klasik, Anda dapat menjalankan query berikut:

```
SELECT kategori, COUNT(penulis) FROM klasik GROUP BY kategori;
```

Output dari query tersebut akan terlihat seperti ini:

```
+-----+-----+
| kategori | COUNT(author) |
+-----+-----+
| Classic Fiction | 3 |
| Non-Fiction | 1 |
| Play | 1 |
+-----+
3 rows in set (0.00 sec)
```

Menggunakan Operator Logika

Anda juga dapat menggunakan operator logika AND, OR, dan NOT dalam query WHERE Anda untuk lebih mempersempit pilihan. Contoh berikut menunjukkan penggunaan masing-masing operator, tetapi Anda juga dapat mencampur dan mencocokkan penggunaannya sesuai dengan yang Anda butuhkan.

```
SELECT penulis,judul FROM klasik WHERE
penulis LIKE "Charles%" AND penulis LIKE "%Darwin";
SELECT penulis,judul FROM klasik WHERE
penulis LIKE "%Mark Twain%" OR penulis LIKE "%Samuel Langhorne
Clemens%";
SELECT penulis,judul FROM klasik WHERE
penulis LIKE "Charles%" AND penulis NOT LIKE "%Darwin";
```

###

pustaka-indo.blogspot.com

BAB 06

Membuat Aplikasi Berbasis Database Sederhana

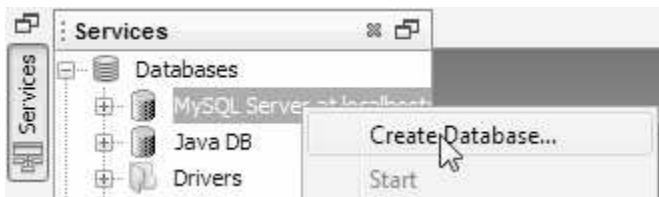
Setelah mengenal Java dan dasar-dasar pemrograman berbasis database, maka sekarang kita akan membuat aplikasi sederhana berbasis database. Proses pembuatan aplikasi ini melibatkan NetBeans sebagai salah satu tool untuk membangun tampilan aplikasi.

Aplikasi berbasis database ini memanfaatkan proses baku yang disebut CRUD, yang merupakan singkatan dari Create, Read, Update, dan Delete. Disebut proses baku karena ketika bekerja menggunakan database, umumnya kita akan melewati proses yang disebut membuat/menambah data, menampilkan data-data tersebut, mengubah atau mengedit data, dan menghapus data. Pada bagian ini Anda akan belajar membuat aplikasi CRUD sederhana dengan menggunakan NetBeans IDE dan database MySQL.

Pembuatan Database

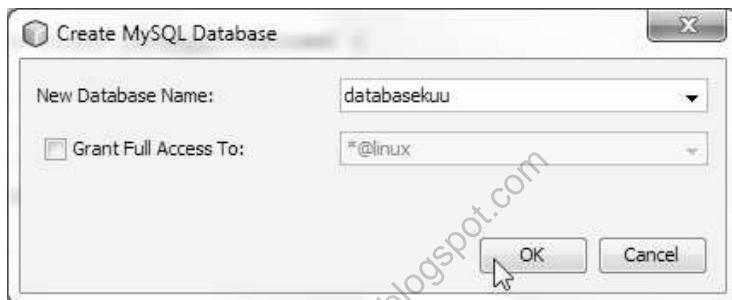
Pertama-tama Anda perlu menyiapkan aplikasi dengan membuka NetBeans dan menjalankan MySQL. Anda perlu menyiapkan database MySQL terlebih dahulu dengan langkah-langkah berikut:

1. Pilih **Services** di sebelah kiri atas IDE dan kemudian klik-kanan **MySQL Server at...**
2. Pilih **Create Database**.



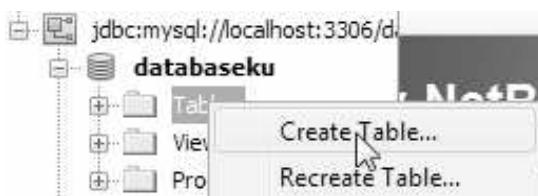
Menyiapkan database MySQL

3. Setelah itu masukkan nama database yang diinginkan dan klik tombol OK.



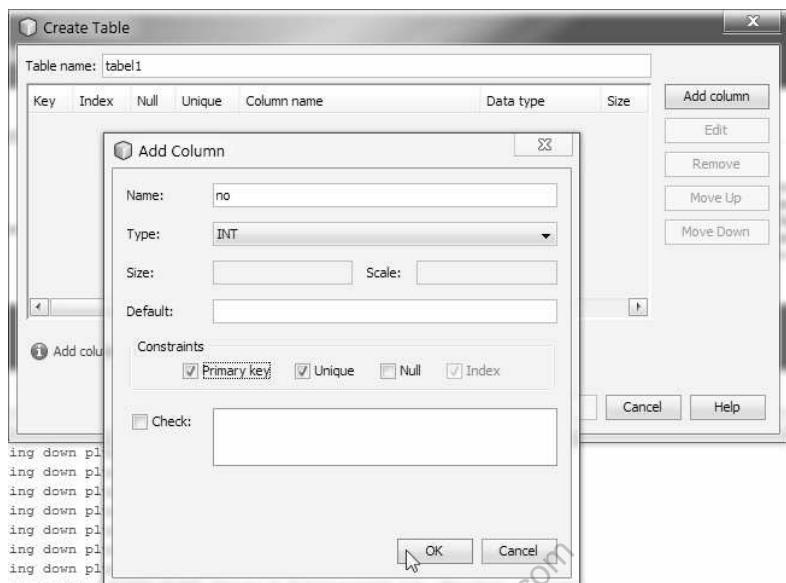
Mengatur nama database

4. Perluas database dan klik-kanan folder Tables.
5. Pilih opsi **Create Table**.



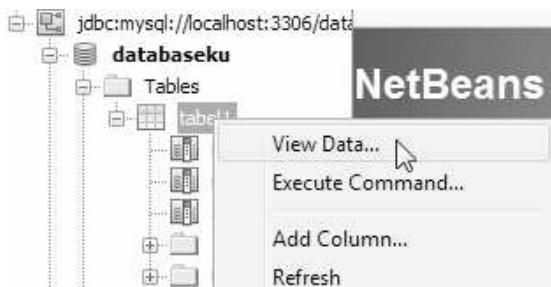
Membuat tabel

6. Beri nama **tabel1**.
7. Pilih **Add column** dan beri nama kolom **no**, tipe **INT**, dan pilih **Primary key**, kemudian klik **OK**.



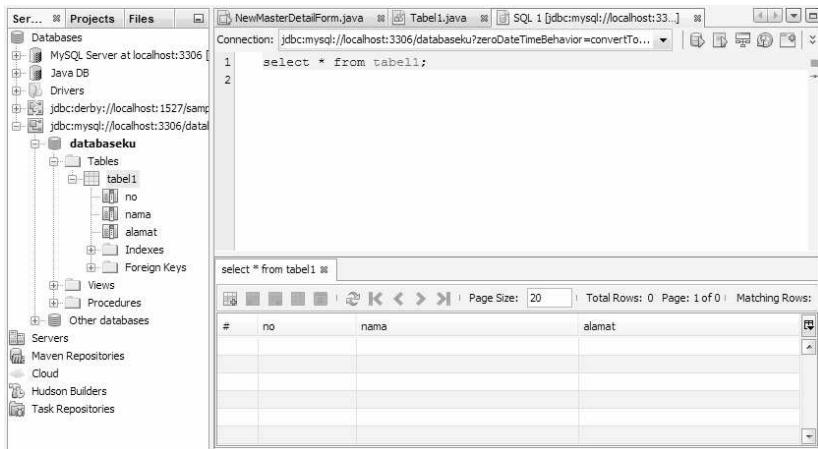
Menambahkan tabel beserta kolom

8. Dengan cara yang sama, tambahkan kolom **nama**, tipe VARCHAR, ukuran **100**, dan kolom **alamat**, tipe TEXT. Kemudian klik OK.
9. Perluas **tabel1**.
10. Klik-kanan dan pilih **View Data...**
11. Anda kemudian dapat melihat **tabel1** ditampilkan dalam IDE.



Menampilkan tabel1

12. Tabel database Anda akan muncul pada IDE.

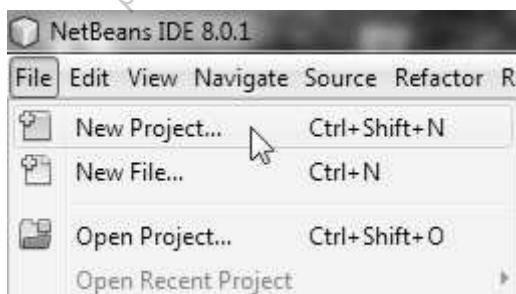


Tampilan tabel database

Menyiapkan Aplikasi

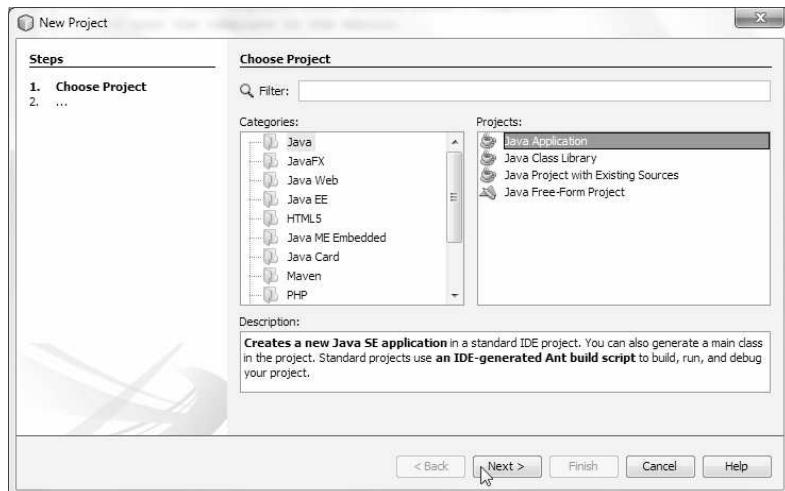
Saat ini Anda sudah siap dengan tabel database untuk tempat penyimpanan data. Setelah membuat tabel database, Anda masuk ke tahap berikutnya dengan membuat proyek aplikasi Java. Berikut langkah-langkah yang bisa Anda lakukan:

1. Pilih File > New Project.



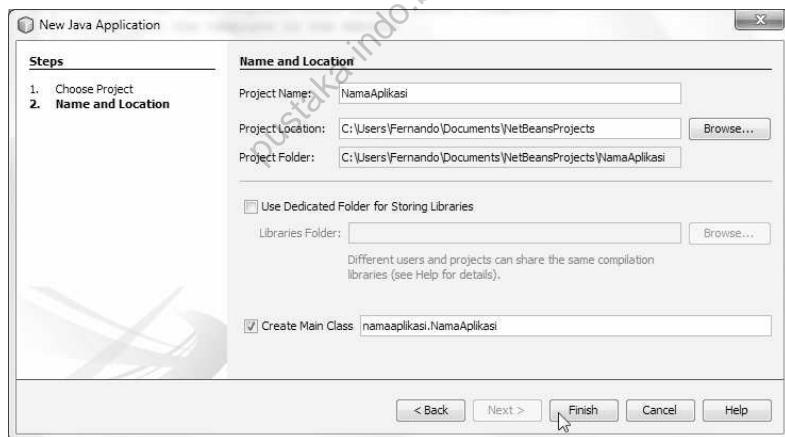
Memilih File > New Project

2. Pilih Java > Java Application, kemudian klik Next.



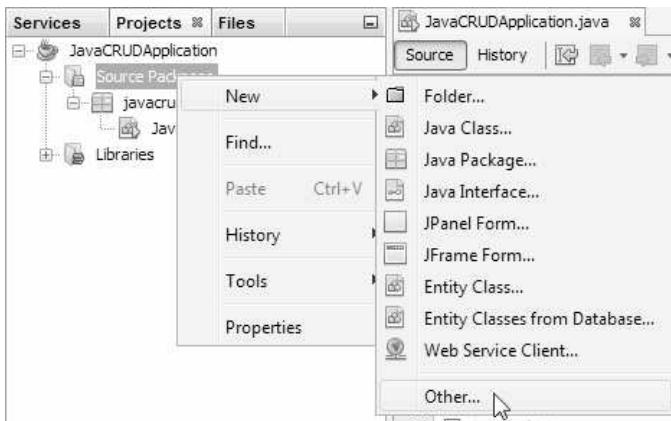
Memilih Java Application

3. Ubah nama proyek sesuai dengan keinginan Anda.
4. Kemudian klik **Finish**.



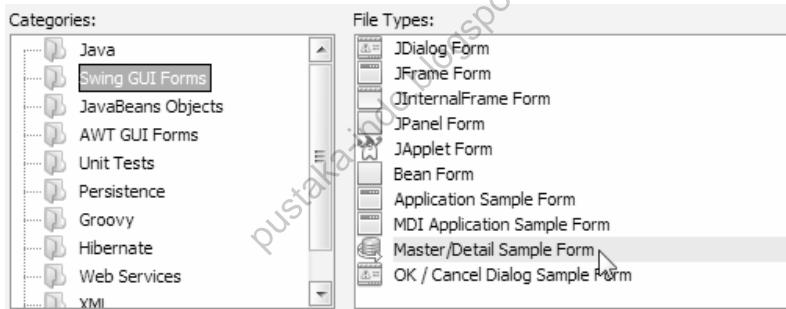
Mengatur nama project

5. Perluas aplikasi dan klik-kanan **Source Packages**.
6. Kemudian pilih **New > Other...**



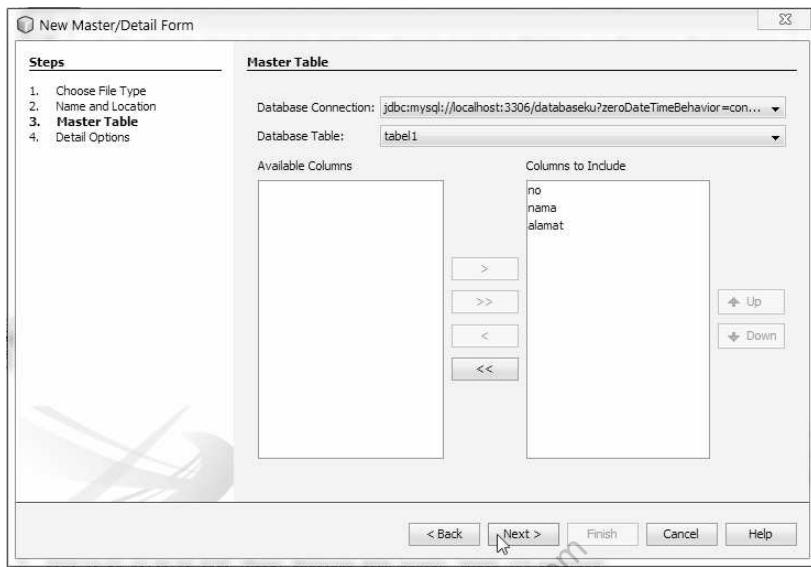
Menambahkan Form

7. Pilih kategori **Swing GUI Forms** dan tipe file **Master/Detail Sample Form** kemudian klik **Next**.



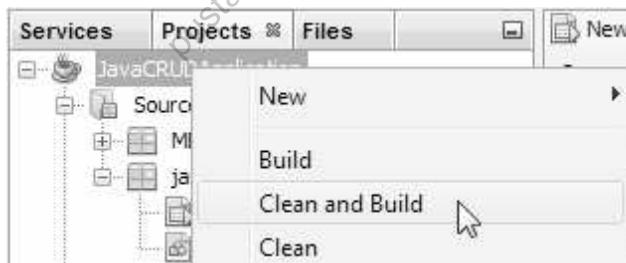
Memilih Form

8. Anda kemudian dapat mengubah nama kelas sesuai dengan keinginan Anda dan kemudian mengklik **Next**.
9. Pada **Master Table** pilih koneksi database yang baru saja Anda buat, pilih **tabel1**,
10. Klik **Next** dan kemudian klik **Finish**.



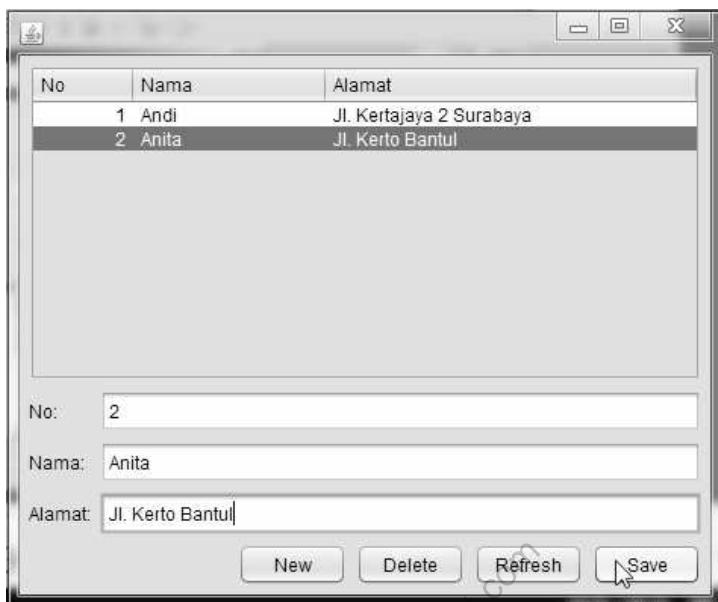
Memilih tabel database

11. Hapus **NamaProyek.java** di sebelah kiri untuk menghindari error saat menjalankan aplikasi.
12. Klik-kanan proyek kemudian pilih **Clean and Build**.



Membersihkan dan mem-build aplikasi

13. Anda kemudian dapat menjalankan aplikasi dengan memilih **Run**.



Aplikasi berbasis database yang dibuat menggunakan Java

BAB 07

Contoh Pembuatan Aplikasi Java Berbasis Database

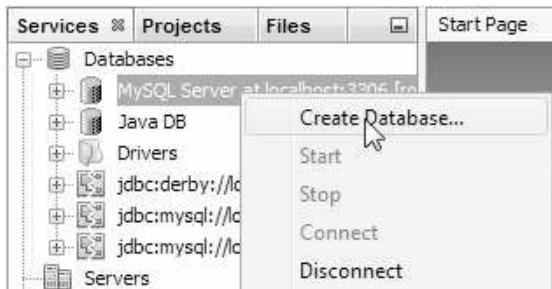
Dalam bab terakhir ini, kita akan mencoba membuat aplikasi Java berbasis database yang lebih bersifat fungsional. Tepatnya, kita akan merancang aplikasi penggajian atau pengupahan yang dapat dikembangkan sendiri untuk keperluan industri UKM. Karena proses penggajian melibatkan banyak tahapan, maka aplikasi ini pun dibuat berdasarkan tahapan-tahapan yang runtut dari penyiapan database hingga finalisasi.

Aplikasi penggajian di dalam bab ini dibuat untuk membantu Anda memahami bagaimana caranya menciptakan aplikasi siap-pakai menggunakan Java yang dapat dikembangkan sendiri seperti harapan Anda.

Menyiapkan Database

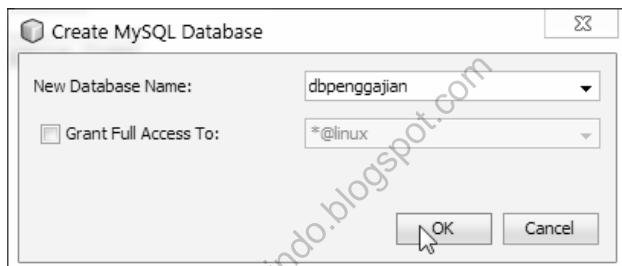
Sebelum membuat desain antar-muka untuk aplikasi, ada baiknya jika kita mempersiapkan database terlebih dulu sehingga ketika pengujian dapat langsung terkoneksi dengan data-data yang akan dimasukkan dan dibaca oleh aplikasi tersebut.

1. Pada jendela **Service**, klik-kanan **MySQL Server** dan pilih **Create Database**.



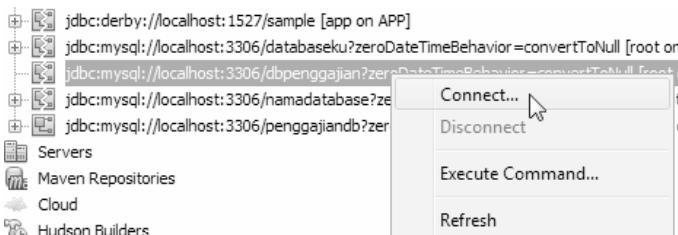
Membuat database

2. Pada kotak dialog yang muncul, masukkan nama database **dbpenggajian** dan kemudian klik OK.



Membuat database dbpenggajian

3. Klik-kanan pada database **dbpenggajian** dan pilih **Connect** untuk menghubungkan dengan database.



Menghubungkan dengan database dbpenggajian

4. Setelah terhubung dengan database, Anda kemudian dapat menambahkan tabel dalam database.

Membuat Tabel dalam dbpenggajian

Dalam database aplikasi penggajian ini, Anda akan membuat tiga tabel yaitu tabel karyawan, jabatan, dan pengguna. Tabel karyawan untuk menampung data karyawan, tabel jabatan untuk menampung deskripsi jabatan, dan tabel pengguna untuk menampung data pengguna aplikasi.

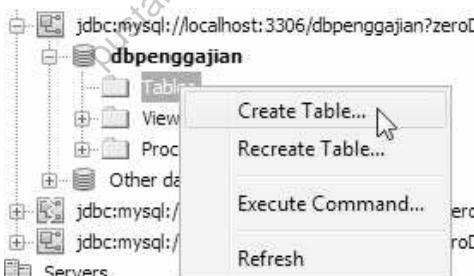
Tabel Karyawan

- Pada **Database Explorer**, perluas database **dbpenggajian** untuk menampilkan tiga folder: **Tables**, **Views**, dan **Procedures**.



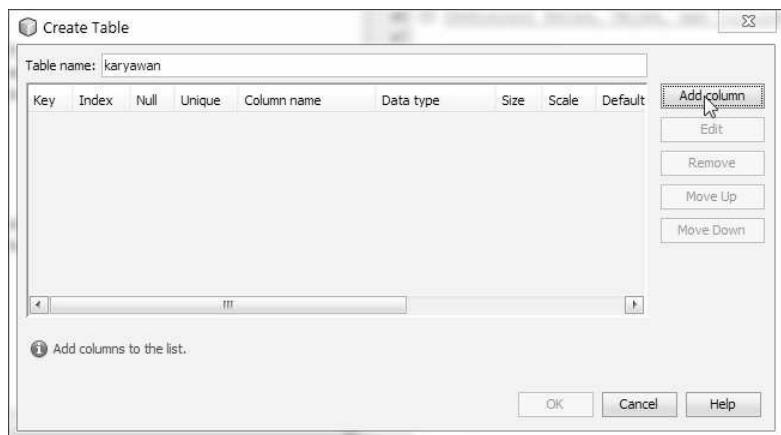
Menampilkan folder dalam database

- Klik-kanan folder **Tables** dan pilih **Create Table...**



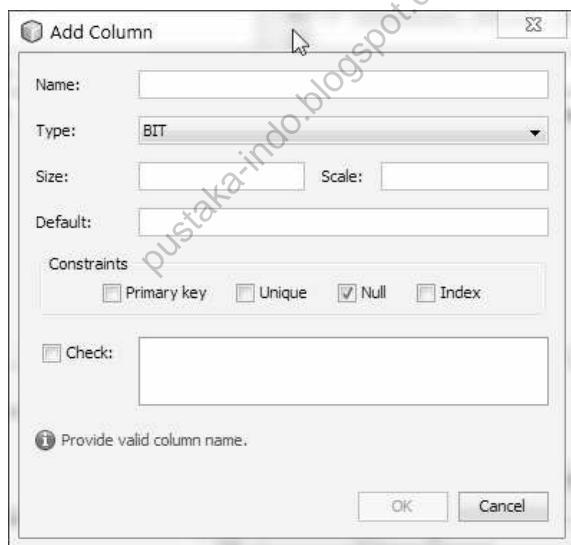
Membuat tabel

- Beri nama tabel **karyawan** dan kemudian klik **Add column**.



Mengatur nama tabel

4. Akan muncul kotak untuk menambahkan kolom pada tabel.



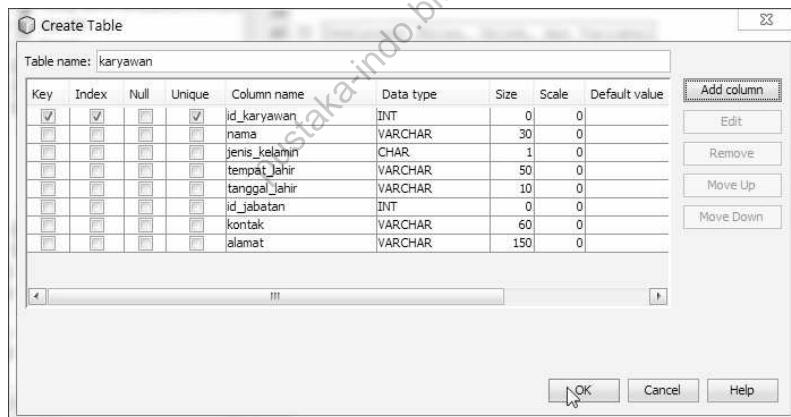
Kotak untuk menambahkan kolom

5. Tambahkan kolom sesuai dengan tabel berikut dan kemudian klik OK untuk menambahkan setiap kolom.

NAME	TYPE	SIZE	CONSTRAINTS
id_karyawan	int		Primary key, Unique, Index
nama	varchar	30	
jenis_kelamin	char	1	
tempat_lahir	varchar	50	
tanggal_lahir	varchar	10	
id_jabatan	int		
kontak	varchar	60	
alamat	varchar	150	

Daftar kolom yang akan ditambahkan

- Setelah menambahkan kolom-kolom pada tabel, klik **OK** untuk membuat tabel karyawan.



Membuat tabel karyawan lengkap dengan kolom-kolomnya

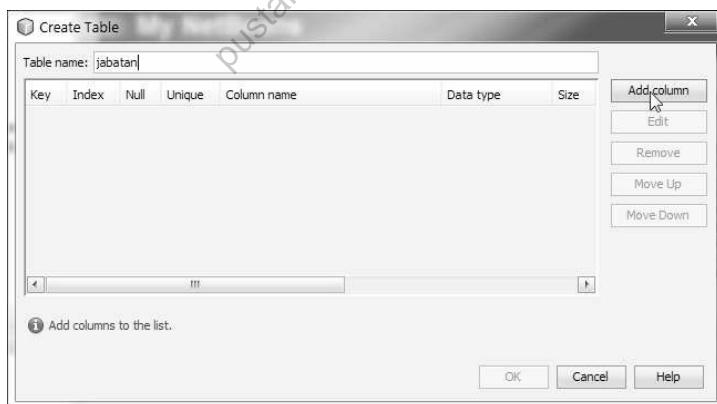
- Untuk memeriksa tabel yang baru saja dibuat, klik-kanan pada folder Tables dan kemudian perluas untuk melihat isinya.



Tabel karyawan

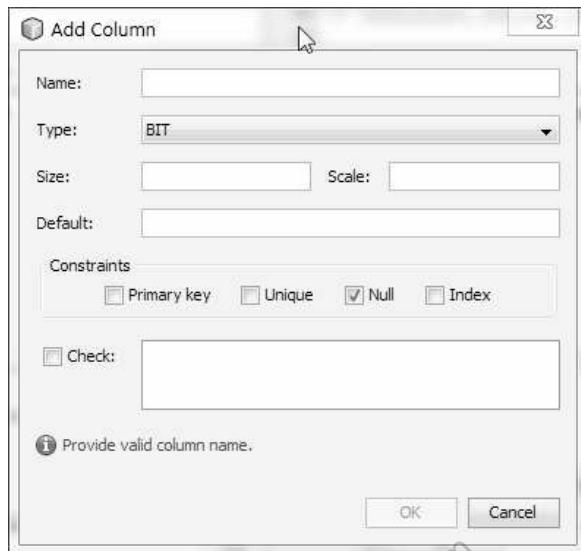
Tabel Jabatan

1. Ikuti langkah pertama dan kedua yang sama seperti pada saat membuat tabel karyawan.
2. Beri nama tabel **jabatan** dan kemudian klik **Add column**.



Mengatur nama tabel

3. Akan muncul kotak untuk menambahkan kolom pada tabel.



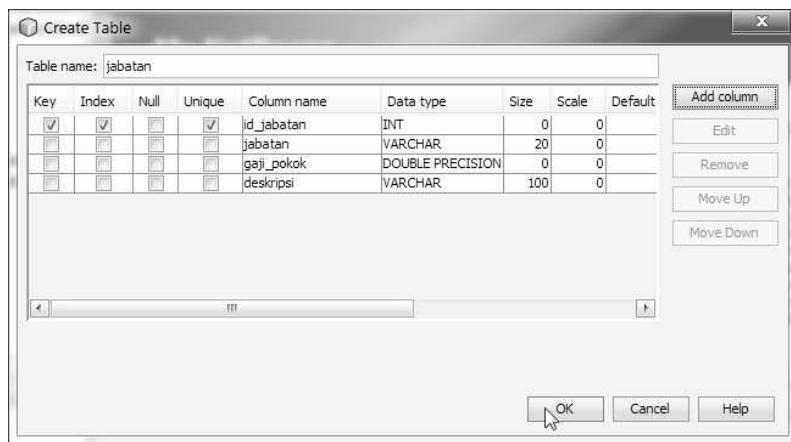
Kotak untuk menambahkan kolom

4. Tambahkan kolom sesuai dengan tabel berikut dan kemudian klik **OK** untuk menambahkan setiap kolom.

NAME	TYPE	SIZE	CONSTRAINTS
id_jabatan	int		Primary key, Unique, Index
jabatan	varchar	20	
gaji_pokok	varchar	10	
deskripsi	varchar	100	

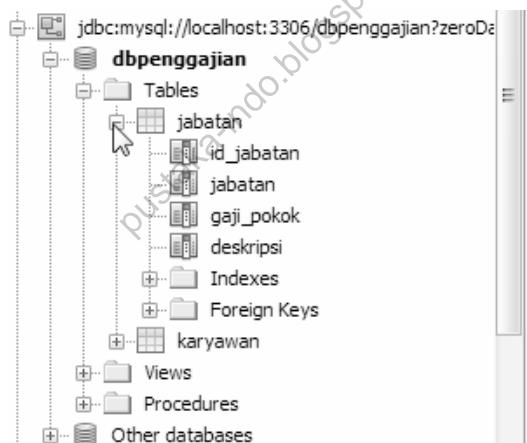
Daftar kolom yang akan ditambahkan

5. Setelah menambahkan kolom-kolom pada tabel, klik **OK** untuk membuat tabel jabatan.



Membuat tabel jabatan lengkap dengan kolom-kolomnya

6. Untuk memeriksa tabel yang baru saja dibuat, klik-kanan pada folder **Tables** dan kemudian perluas untuk melihat isinya.

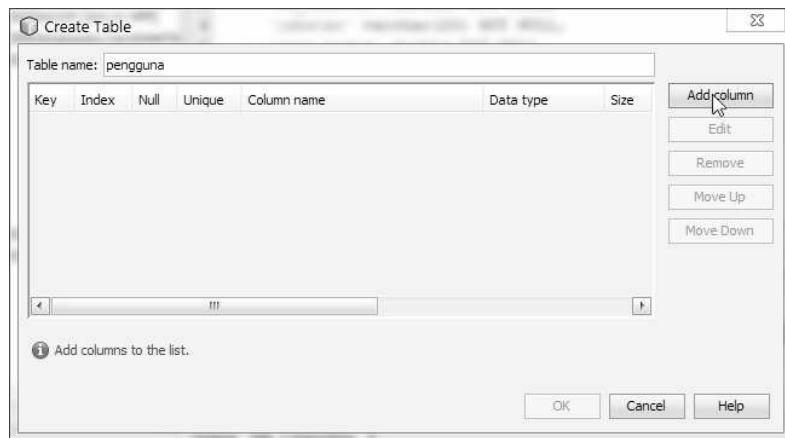


Tabel jabatan

Tabel Pengguna

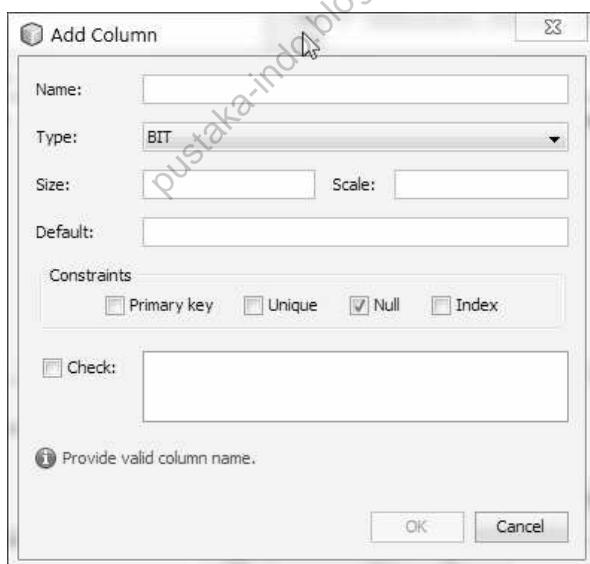
1. Ikuti langkah pertama dan kedua yang sama seperti pada saat membuat tabel karyawan.

2. Beri nama tabel **pengguna** dan kemudian klik **Add column**.



Mengatur nama tabel

3. Akan muncul kotak untuk menambahkan kolom pada tabel.



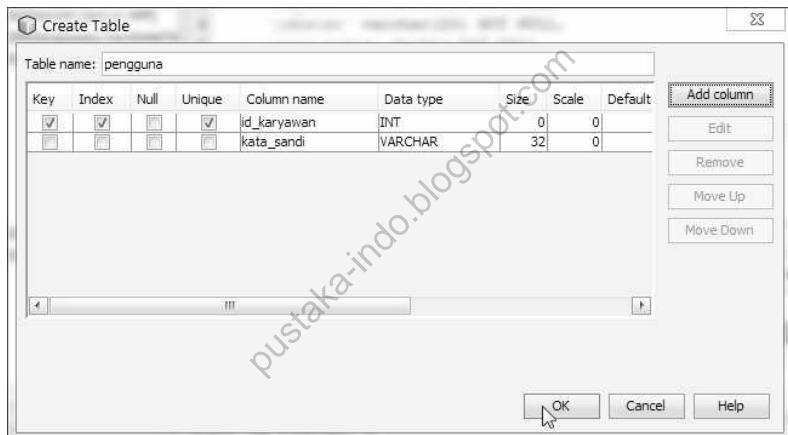
Kotak untuk menambahkan kolom

4. Tambahkan kolom sesuai dengan tabel berikut dan kemudian klik **OK** untuk menambahkan setiap kolom.

NAME	TYPE	SIZE	CONSTRAINTS
id_karyawan	int		Primary key, Unique, Index
kata_sandi	varchar	32	

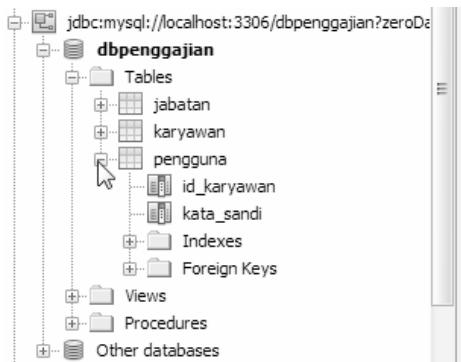
Daftar kolom yang akan ditambahkan

5. Setelah menambahkan kolom-kolom pada tabel, klik **OK** untuk membuat tabel pengguna.



Membuat tabel pengguna lengkap dengan kolom-kolomnya

6. Untuk memeriksa tabel yang baru saja dibuat, klik-kanan pada folder **Tables** dan kemudian perluas untuk melihat isinya.



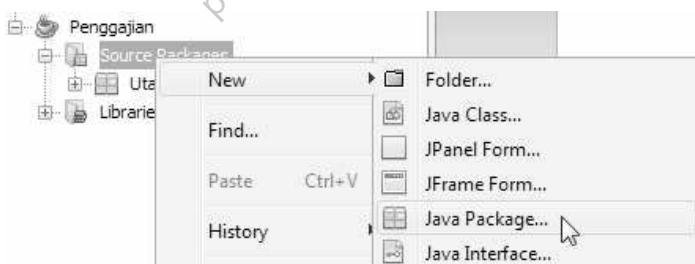
Tabel pengguna

Package

Untuk membuat aplikasi Penggajian, Anda perlu menyiapkan package dan kelas-kelas yang akan digunakan. Berikut ini package yang akan Anda gunakan dalam proyek Penggajian.

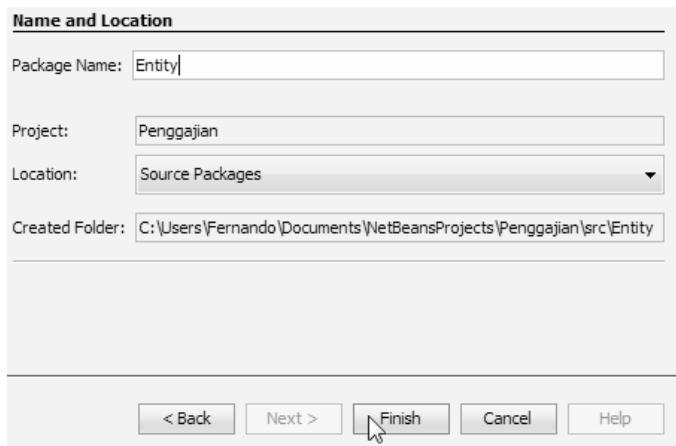
Package Entity

1. Klik-kanan **Source Packages** pada proyek **Penggajian** kemudian pilih **New Java Package...**



Menambahkan package baru

2. Beri nama package **Entity** kemudian klik **Finish**.



Mengatur nama package

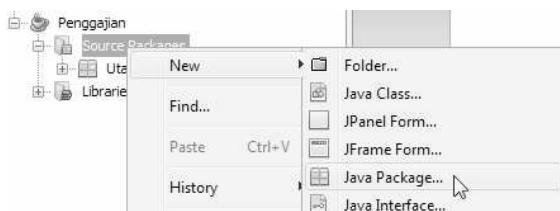
3. Package **Entity** akan ditambahkan pada proyek Penggajian.



Package Entity ditambahkan

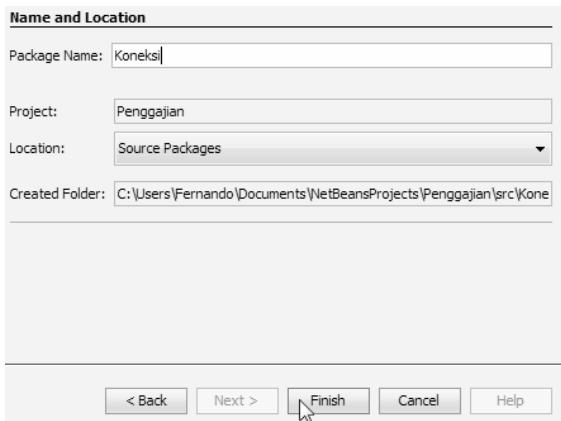
Package Koneksi

1. Klik-kanan **Source Packages** pada proyek Penggajian kemudian pilih **New Java Package...**



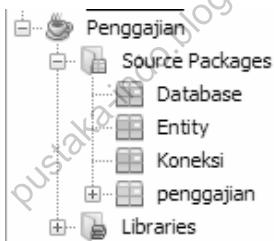
Menambahkan package baru

2. Beri nama package **Koneksi** kemudian klik **Finish**.



Mengatur nama package

3. Package **Koneksi** akan ditambahkan pada proyek **Penggajian**.

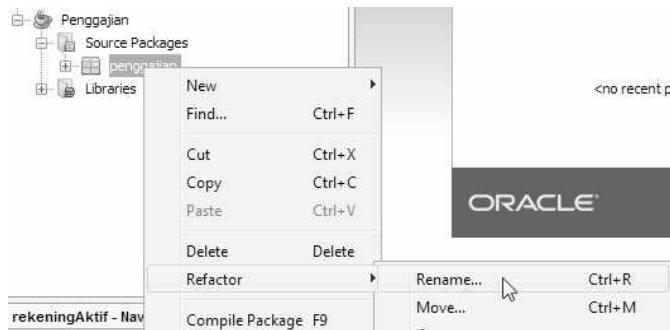


Package Koneksi ditambahkan

Package Utama

Package Utama berisi kelas yang digunakan untuk menjalankan program dan kelas antarmuka aplikasi. Untuk membuat package Utama, Anda hanya perlu mengubah nama package yang ada dengan cara berikut:

1. Klik-kanan package **penggajian** kemudian pilih **Refactor > Rename**.



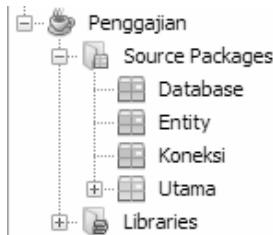
Memilih Refactor > Rename

2. Ubah nama package menjadi package **Utama** kemudian klik **Refactor**.



Mengubah nama package

3. Package **Utama** akan muncul dalam proyek **Penggajian**.



Package Utama

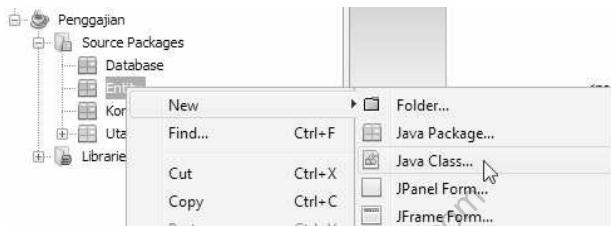
Kelas

Berikut ini kelas yang akan Anda gunakan dalam proyek Penggajian.

Kelas dalam Package Entity

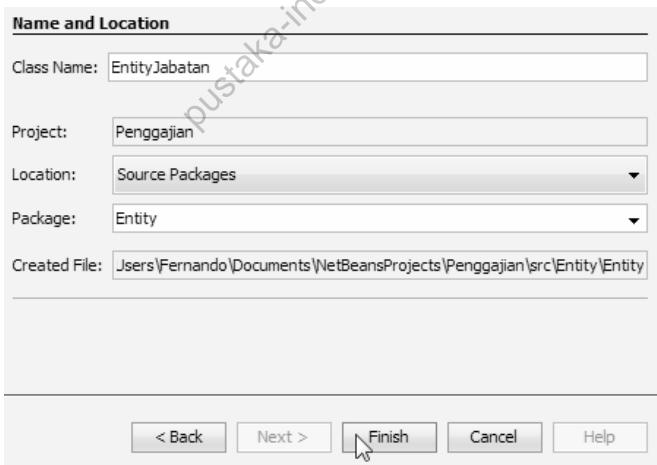
Kelas EntityJabatan

1. Klik-kanan pada package **Entity** kemudian pilih **New > Java Class...**



Memilih Java Class...

2. Beri nama kelas **EntityJabatan** kemudian klik **Finish**.



Mengatur nama kelas

3. Kelas **EntityJabatan** akan ditambahkan dalam proyek **Penggajian**.

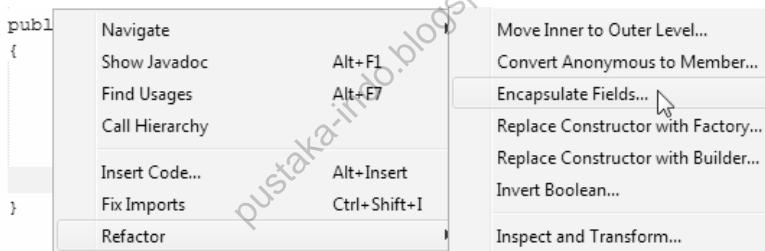
```
11 public class EntityJabatan
12 {
13
14
15
16 }
```

Kelas EntityJabatan

4. Tambahkan kode berikut dalam kelas **EntityJabatan**.

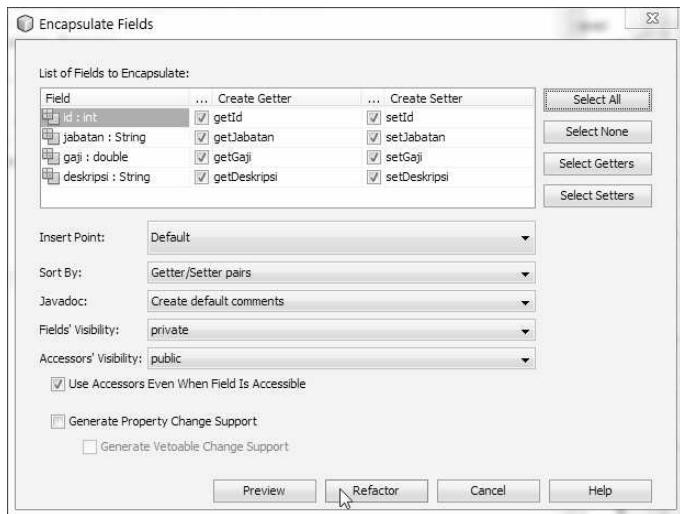
```
int id;
String jabatan;
double gaji;
String deskripsi;
```

5. Lakukan enkapsulasi field dengan cara klik-kanan di dalam kelas dan pilih **Refactor - Encapsulate Fields...**



Memilih Encapsulate Fields...

6. Pilih **Select All** dan klik **Refactor**.



Melakukan enkapsulasi

7. Getter dan setter untuk setiap field akan dibuat secara otomatis.

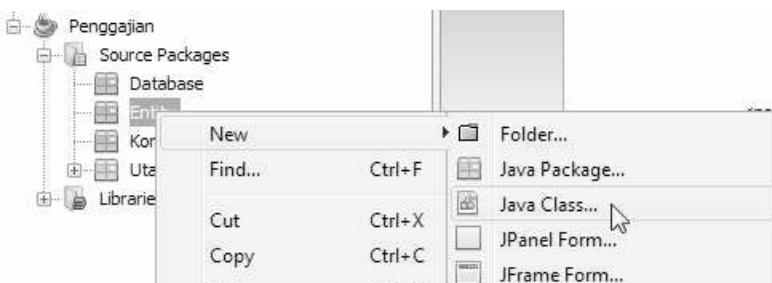
```
12 public class EntityJabatan
13 {
14     private int id;
15     private String jabatan;
16     private double gaji;
17     private String deskripsi;
18
19     /**
20      * @return the id
21     */
22     public int getId() {
23         return id;
24     }
25
26     /**
27      * @param id the id to set
28     */
29     public void setId(int id) {
30         this.id = id;
31     }

```

Getter dan setter dalam kelas EntityJabatan

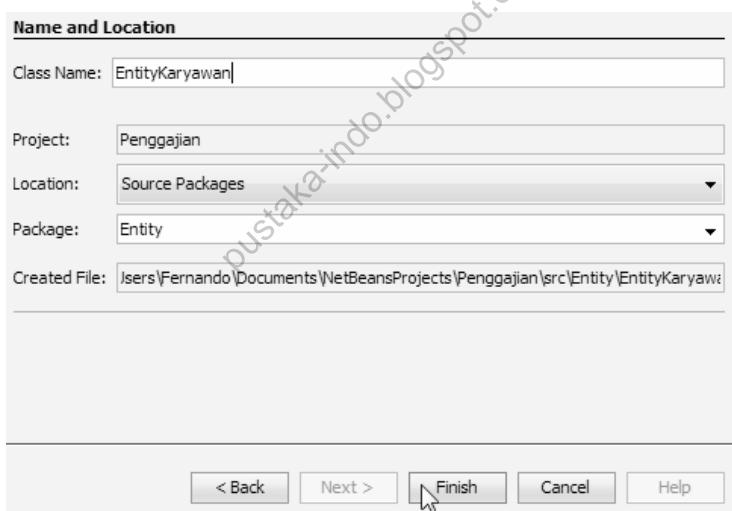
Kelas EntityKaryawan

1. Klik-kanan pada package **Entity** kemudian pilih **New > Java Class...**



Memilih Java Class...

2. Beri nama kelas **EntityKaryawan** kemudian klik **Finish**.



Mengatur nama kelas

3. Kelas **EntityKaryawan** akan ditambahkan dalam proyek **Penggajian**.

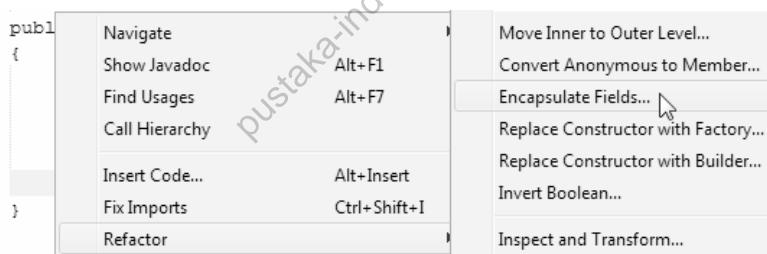
```
11
12 public class EntityKaryawan
13 {
14
15 }
16
```

Kelas EntityKaryawan

4. Tambahkan kode berikut dalam kelas **EntityKaryawan**.

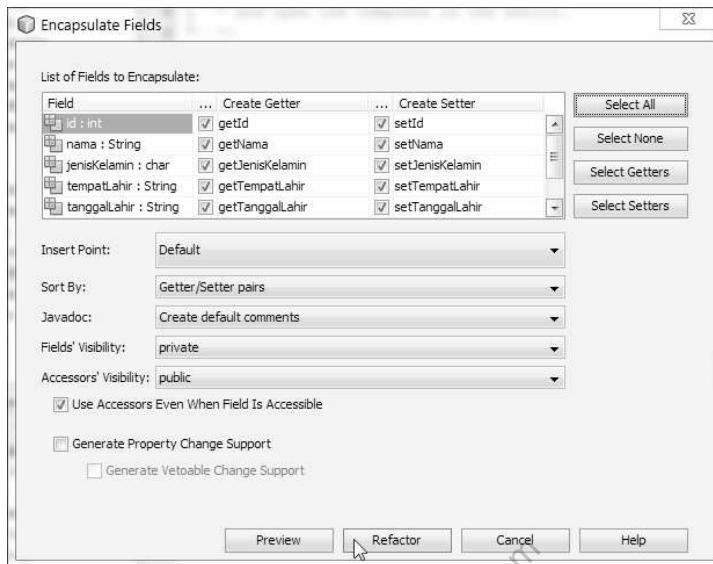
```
int id;
String nama;
char jenisKelamin;
String tempatLahir;
String tanggalLahir;
String kontak;
String alamat;
```

5. Lakukan enkapsulasi field dengan cara klik-kanan di dalam kelas dan pilih **Refactor > Encapsulate Fields...**



Memilih Encapsulate Fields...

6. Pilih **Select All** dan klik **Refactor**.



Melakukan enkapsulasi

7. Getter dan setter untuk setiap field akan dibuat secara otomatis.

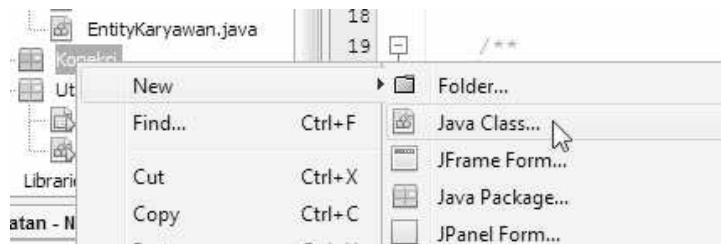
```
12 public class EntityKaryawan
13 {
14     private int id;
15     private String nama;
16     private char jenisKelamin;
17     private String tempatLahir;
18     private String tanggalLahir;
19     private String kontak;
20     private String alamat;
21
22     /**
23      * @return the id
24      */
25     public int getId() {
26         return id;
27     }
28
29     /**
30      * @param id the id to set
31      */
32     public void setId(int id) {
33         this.id = id;
34     }
```

Getter dan setter dalam kelas EntityKaryawan

Kelas dalam Package Koneksi

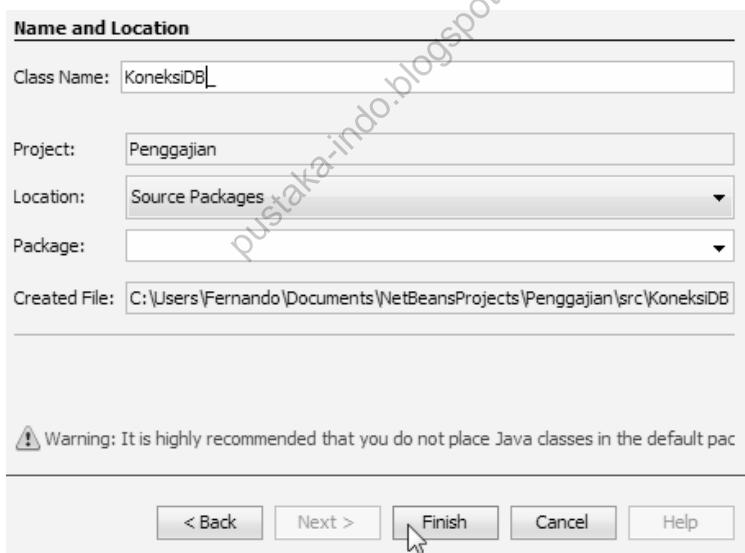
Kelas KoneksiDB

1. Klik-kanan pada package **Koneksi** kemudian pilih **New > Java Class...**



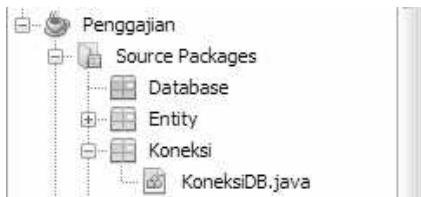
Memilih Java Class...

2. Beri nama kelas **KoneksiDB** kemudian klik **Finish**.



Mengatur nama kelas

3. Kelas **KoneksiDB** akan ditambahkan dalam proyek **Penggajian**.



Kelas KoneksiDB

4. Tambahkan kode berikut dalam kelas **KoneksiDB**.

```
package Koneksi;

import java.sql.*;
import javax.swing.*;

public class KoneksiDB
{
    Connection conn = null;
    public static Connection ConnectDb()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
                DriverManager.getConnection("jdbc:mysql://localhost:3306/dbpenggajian", "root",
                "passwordSQLAnda");
            return conn;
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null, e);
            return null;
        }
    }
}
```

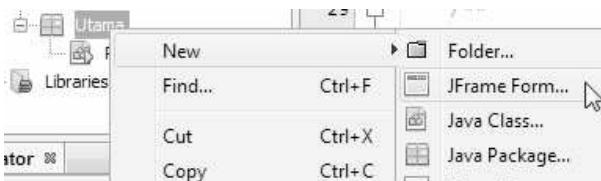
Membuat Antarmuka (Graphical User Interface) Aplikasi

Agar dapat digunakan oleh sembarang orang, aplikasi harus didesain dengan tampilan antarmuka yang menarik. Di sinilah peran NetBeans untuk membantu kita menciptakan form dan berbagai kontrolnya dengan mudah agar aplikasi penggajian ini memiliki bentuk yang memudahkan user dalam menggunakannya.

Berikut ini langkah-langkah untuk membuat antarmuka aplikasi penggajian.

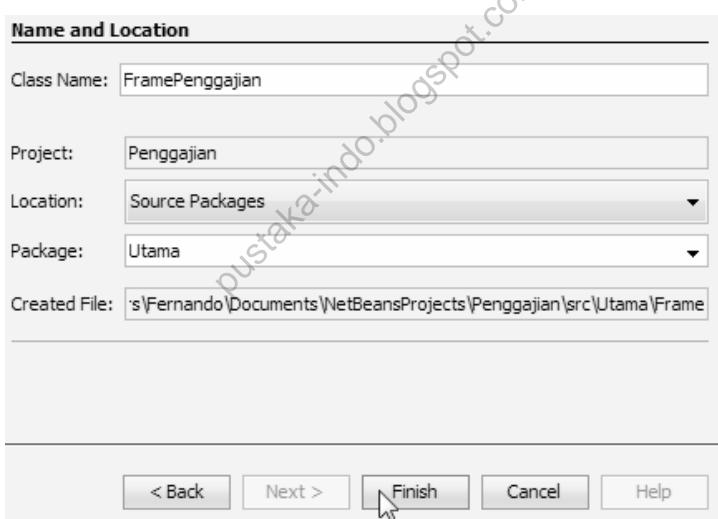
Form

1. Klik-kanan pada package **Utama** dan pilih **New > JFrame Form...**



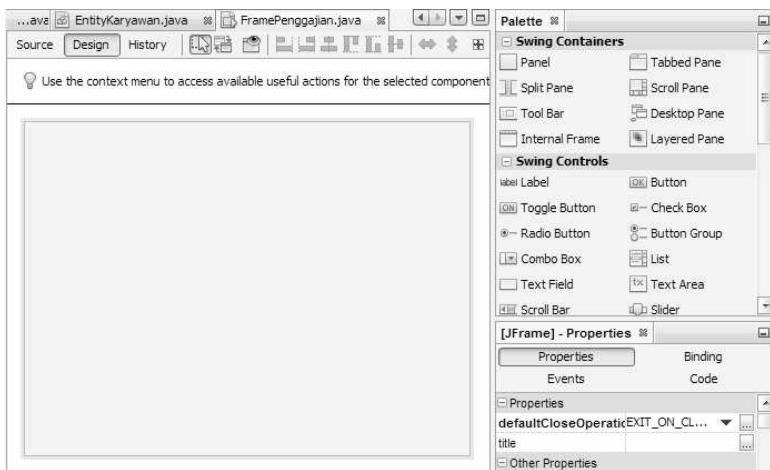
Memilih JFrame Form...

2. Atur nama menjadi **FramePenggajian** kemudian klik **Finish**.



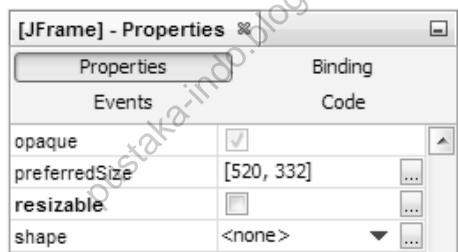
Mengatur nama form

3. Form akan ditambahkan dalam proyek.



Menambahkan form dalam projek

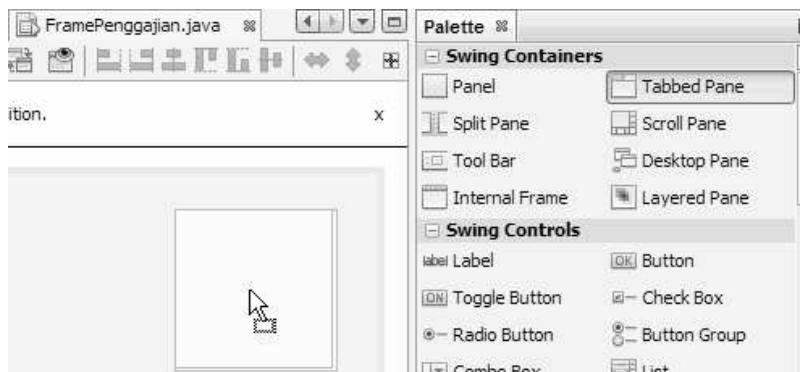
4. Atur ukuran frame menjadi 520, 332 dan hapus centang pada resize dalam jendela Properties dari FramePenggajian.



Mengatur ukuran frame

Tabbed Pane

1. Pilih Tabbed Pane pada jendela Palette kemudian tarik dan lepas ke dalam Frame.



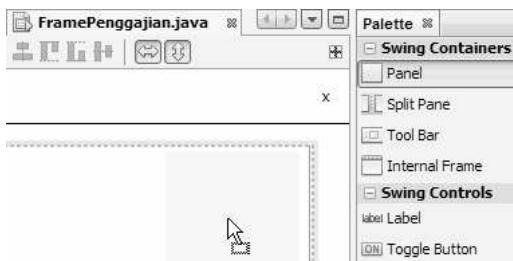
Menambahkan Tabbed Pane

2. Atur ukuran Tabbed Pane sehingga memenuhi seluruh frame dengan cara menarik sisi-sisi dari Tabbed Pane.



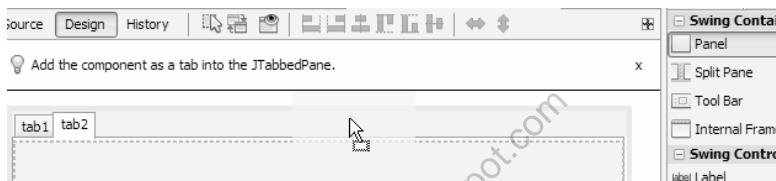
Tabbed Pane memenuhi seluruh frame

3. Pilih Tabbed Pane dan kemudian masukkan panel dengan cara menarik dan melepas Panel ke dalam Tabbed Pane.



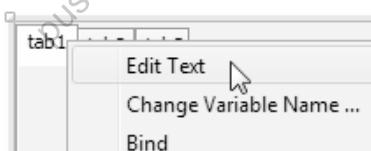
Menambahkan panel

4. Tambahkan dua panel lagi ke dalam Tabbed Pane. Pastikan Anda menambahkan di bagian atas Tabbed Pane seperti gambar berikut.



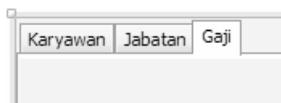
Tambahkan panel di bagian atas Tabbed Pane

5. Ubah teks pada masing-masing panel dengan cara mengklik-kanan nama panel dan kemudian pilih **Edit Text**.



Mengubah teks

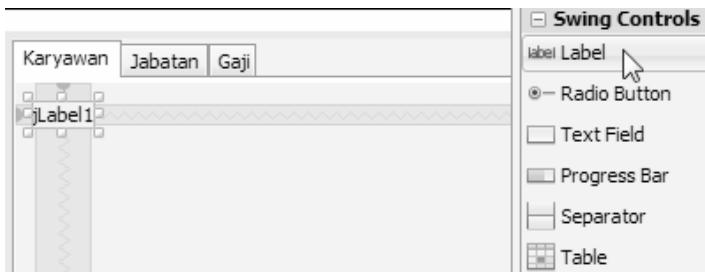
6. Ubah teks kemudian tekan **Enter**. Ubah teks panel sehingga menjadi seperti pada gambar berikut.



Panel Karyawan, Jabatan, dan Gaji

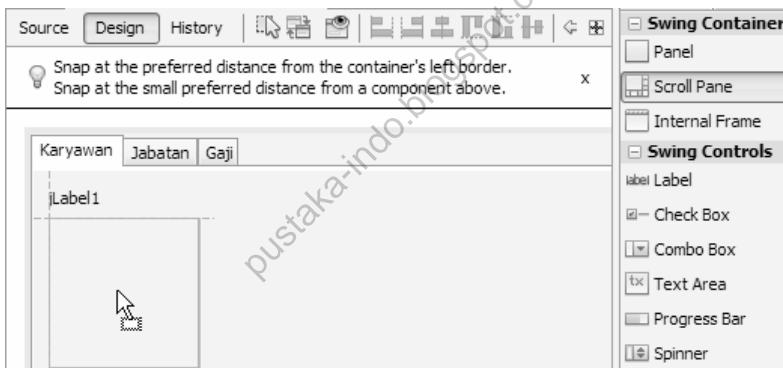
Panel Karyawan

1. Pilih panel **Karyawan** kemudian masukkan label dengan cara menarik **Label** ke dalam panel.



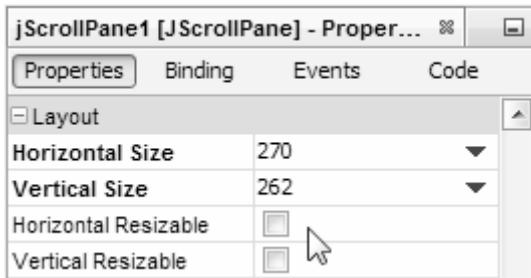
Menambahkan label

2. Tambahkan Scroll Pane ke dalam panel.



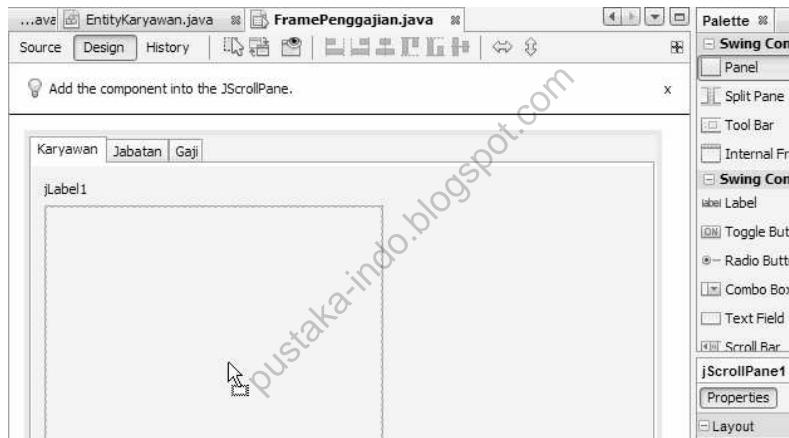
Menambahkan Scroll Pane

3. Pilih Scroll Pane kemudian atur ukuran pada jendela **Properties** menjadi 270, 262 dan hilangkan centang yang ada.



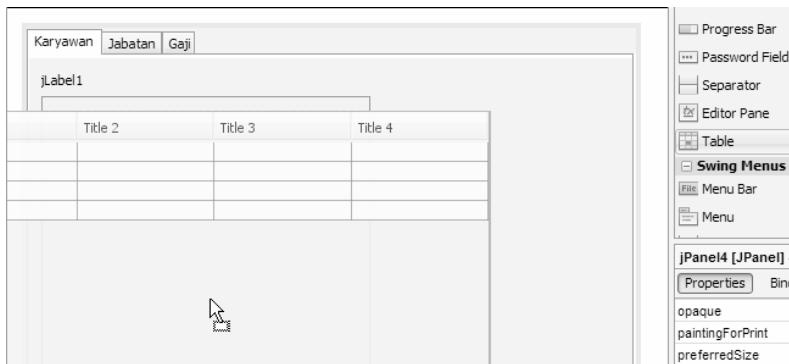
Mengatur ukuran Scroll Pane

4. Tambahkan panel ke dalam Scroll Pane dengan cara memilih Scroll Pane dan kemudian menarik Panel ke dalamnya.



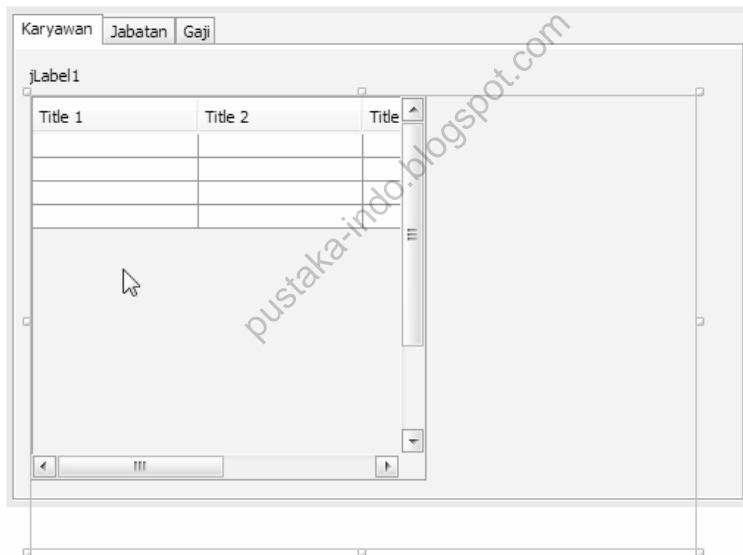
Menambahkan panel ke dalam scroll pane

5. Setelah itu tambahkan tabel ke dalam panel tersebut dengan cara menarik table ke dalam panel.



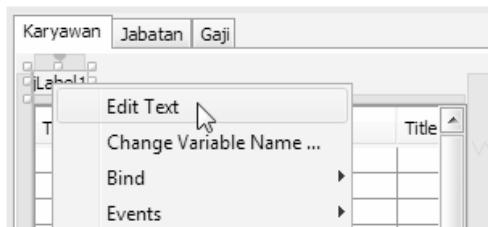
Menambahkan table ke dalam panel

6. Tabel yang telah ditambahkan akan terlihat seperti gambar ini.



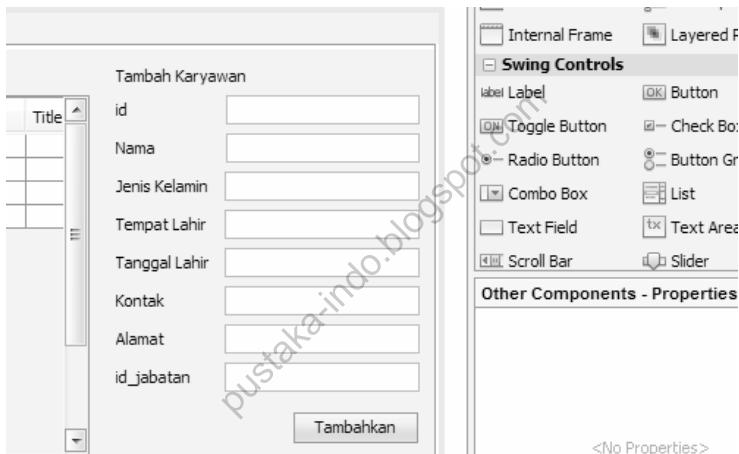
Tabel telah ditambahkan

7. Ubah nama **jLabel1** menjadi **Daftar Karyawan** dengan cara memilih **JLabel1** kemudian klik-kanan dan pilih **Edit Text**.



Mengubah nama label

8. Tambahkan Label, Text Field, dan Button di sebelah kanan panel dan ubah teks pada masing-masing kontrol sehingga menjadi seperti pada gambar berikut.



Menambahkan kontrol lainnya

9. Panel Karyawan akan terlihat seperti ini.

The screenshot shows a Java Swing application window titled "Karyawan". At the top, there are three tabs: "Karyawan", "Jabatan", and "Gaji". The "Karyawan" tab is selected. Below the tabs, there are two main sections: "Daftar Karyawan" and "Tambah Karyawan".

- Daftar Karyawan:** A table with three columns labeled "Title 1", "Title 2", and "Title". It has 6 rows of data.
- Tambah Karyawan:** A form with the following fields:
 - id: A text input field.
 - Nama: A text input field.
 - Jenis Kelamin: A text input field.
 - Tempat Lahir: A text input field.
 - Tanggal Lahir: A text input field.
 - Kontak: A text input field.
 - Alamat: A text input field.
 - id_jabatan: A text input field.
- Action Buttons:** A "Tambahkan" button located at the bottom right of the "Tambah Karyawan" section.

Panel Karyawan

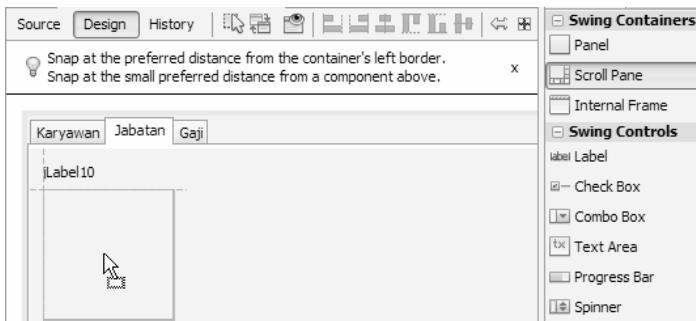
Panel Jabatan

- Pilih panel **Jabatan** kemudian masukkan label dengan cara menarik **Label** ke dalam panel.



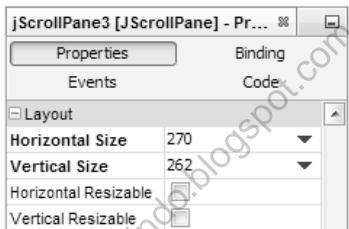
Menambahkan label

- Tambahkan Scroll Pane ke dalam panel.



Menambahkan Scroll Pane

- Pilih Scroll Pane kemudian atur ukuran pada jendela Properties menjadi 270, 262 dan hilangkan centang yang ada.



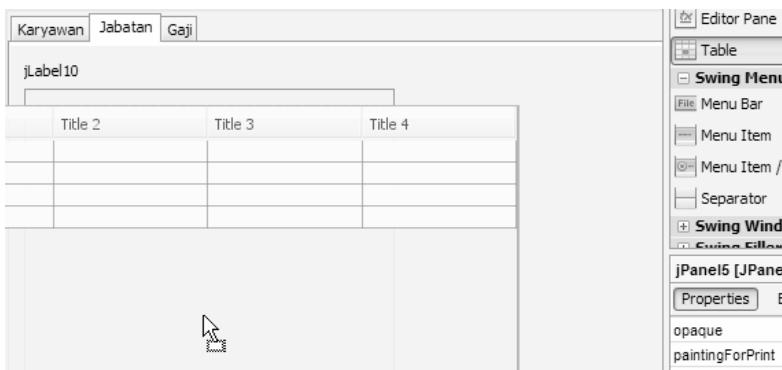
Mengatur ukuran Scroll Pane

- Tambahkan panel ke dalam Scroll Pane dengan cara memilih Scroll Pane dan kemudian menarik Panel ke dalamnya.



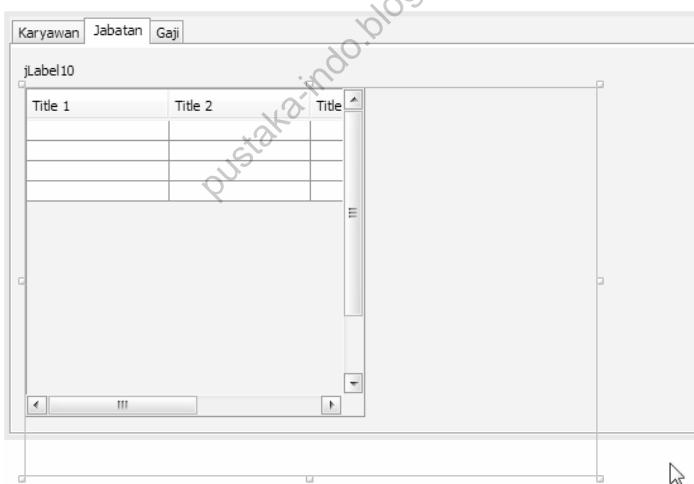
Menambahkan panel ke dalam scroll pane

5. Setelah itu tambahkan tabel ke dalam panel tersebut dengan cara menarik table ke dalam panel.



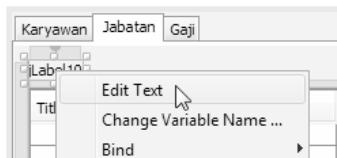
Menambahkan table ke dalam panel

6. Tabel yang telah ditambahkan akan terlihat seperti pada gambar ini.



Tabel telah ditambahkan

7. Ubah nama **jLabel10** menjadi **Daftar Jabatan** dengan cara memilih **JLabel10** kemudian klik-kanan dan pilih **Edit Text**.



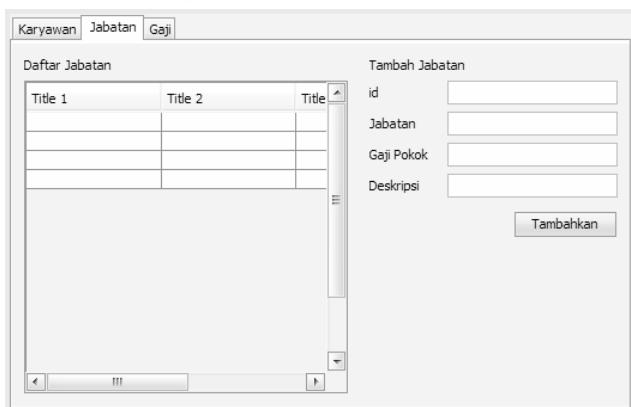
Mengubah nama label

8. Tambahkan Label, Text Field, dan Button di sebelah kanan panel dan ubah teks pada masing-masing kontrol sehingga menjadi seperti pada gambar berikut.



Menambahkan kontrol lainnya

9. Panel **Jabatan** akan terlihat seperti ini:

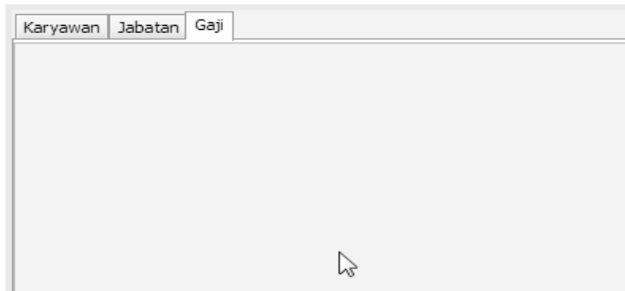


Panel Jabatan

Panel Gaji

Panel Gaji digunakan untuk menampilkan perhitungan gaji karyawan. Berikut langkah-langkah untuk menyiapkan panel Gaji:

1. Pilih panel **Gaji**.



Memilih panel Gaji

2. Tambahkan Label, Text Field, Button, dan Table ke dalam panel kemudian ubah teks pada masing-masing kontrol sehingga menjadi seperti pada gambar berikut.

Title 1	Title 2	Title 3	Title 4

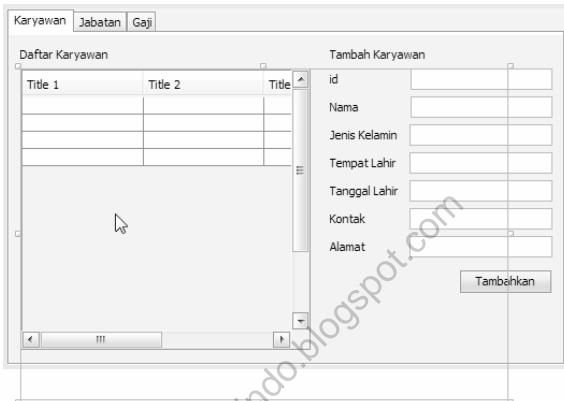
Panel Gaji

Menghubungkan Aplikasi dengan Database

Pada bagian ini Anda akan menghubungkan tabel pada antarmuka aplikasi dengan tabel data pada database.

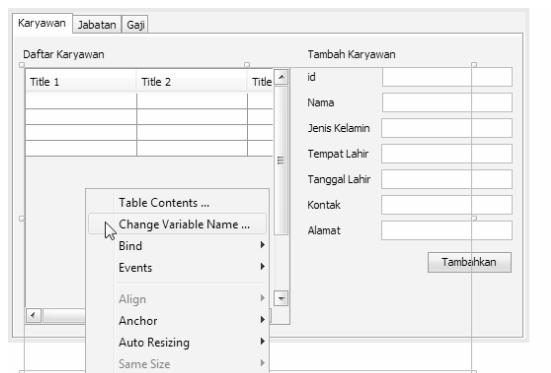
Tabel Karyawan

1. Pada package Utama buka **FramePenggajian**.
2. Pilih panel **Karyawan** dan kemudian pilih tabel yang ada.



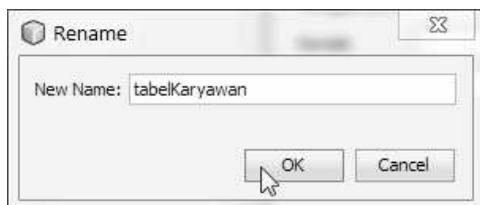
Memilih tabel dalam panel Karyawan

3. Klik-kanan di dalam tabel tersebut dan pilih **Change Variable Name...**



Memilih Change Variable Name...

4. Ubah nama menjadi **tabelKaryawan** kemudian klik OK.



Mengubah nama tabel

5. Setelah mengubah nama tabel, pilih Source di sebelah kiri atas untuk membuka editor kode.

The screenshot shows a Java code editor window titled 'FramePenggajian.java'. The 'Source' tab is selected. The code is as follows:

```
1  /*
2  * To change this license header, choose License Headers :
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package Utama;
7
8 /**
9 *
10 * @author Fernando
11 */
12 public class FramePenggajian extends javax.swing.JFrame {
```

Membuka editor kode

6. Tambahkan kode berikut di atas deklarasi public class **FramePenggajian**....

```
import java.sql.*;
import javax.swing.*;
import net.proteanit.sql.DbUtils;
```

7. Di dalam kelas **FramePenggajian** masukkan kode berikut:

```
Connection conn = null;
ResultSet rs = null;
PreparedStatement pst = null;

private void UpdateTabel()
{
```

```

try
{
    String sql = "select * from karyawan";
    pst = conn.prepareStatement(sql);
    rs = pst.executeQuery();
    tabelKaryawan.setModel(DbUtils.
        resultSetToTableModel(rs));
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e);
}
}

```

8. Tambahkan kode berikut ke dalam konstruktor yang ada.

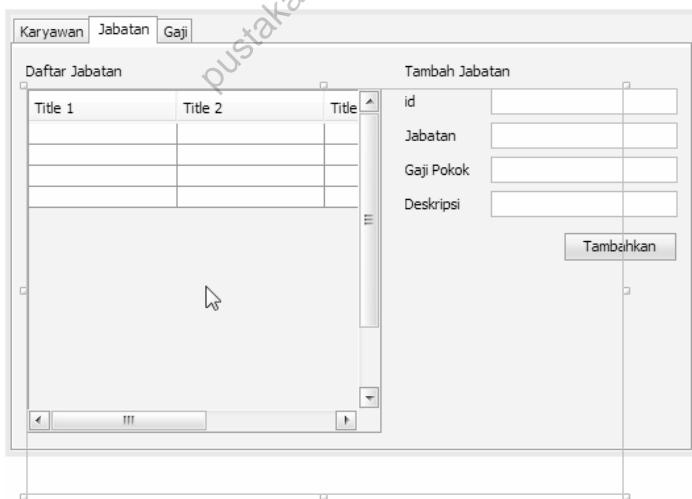
```

public FramePenggajian()
{
    initComponents();
    conn = Koneksi.KoneksiDB.ConnectDb();
    UpdateTabel();
}

```

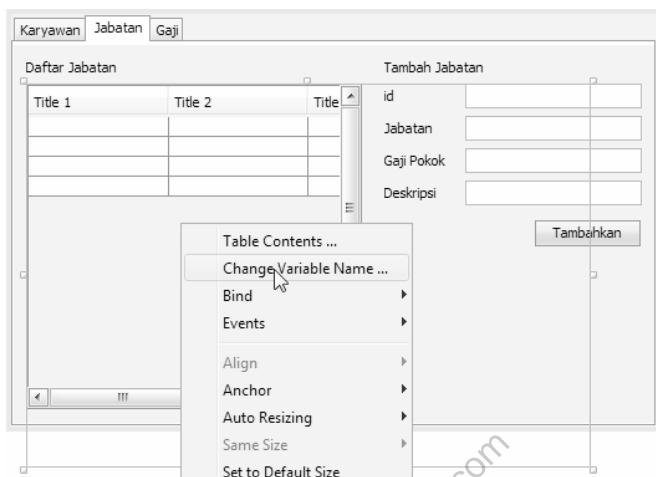
Tabel Jabatan

1. Pada package Utama buka **FramePenggajian**.
2. Pilih panel **Jabatan** dan kemudian pilih tabel yang ada.



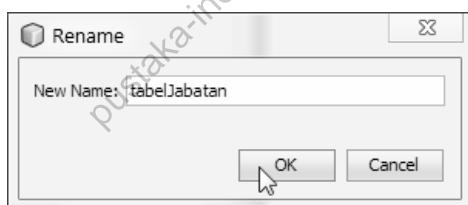
Memilih tabel dalam panel Karyawan

3. Klik-kanan di dalam tabel tersebut dan pilih **Change Variable Name...**



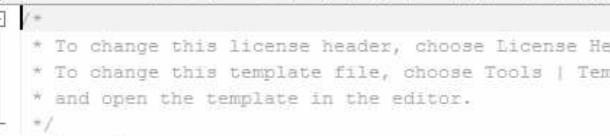
Memilih Change Variable Name...

4. Ubah nama menjadi **tabelJabatan** kemudian klik **OK**.



Mengubah nama tabel

5. Setelah mengubah nama tabel, pilih **Source** di sebelah kiri atas untuk membuka editor kode.



The screenshot shows the Eclipse IDE interface with the title bar "FramePenggajian.java". The "Source" tab is selected. The code editor displays the following Java code:

```
1  /*
2   * To change this license header, choose License Headers
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6 package Utama;
7
8 /**
9  *
10 * @author Fernando
11 */
12 public class FramePenggajian extends javax.swing.JFrame {
```

Membuka editor kode

6. Tambahkan kode berikut di atas deklarasi public class **FramePenggajian**....

```
import java.sql.*;  
import javax.swing.*;  
import net.proteanit.sql.DbUtils;
```

7. Di dalam kelas **FramePengajian** masukkan kode berikut:

```
Connection connj = null;
ResultSet rsj = null;
PreparedStatement pstj = null;

private void UpdateTabelJabatan()
{
    try
    {
        String sqlj = "select * from jabatan;";
        pstj = connj.prepareStatement(sqlj);
        rsj = pstj.executeQuery();
        tabelJabatan.setModel(DbUtils.
            resultSetToTableModel(rsj));
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

8. Tambahkan kode berikut ke dalam konstruktor yang ada.

```
public FramePenggajian()
{
```

```

initComponents();
conn = Koneksi.KoneksiDB.ConnectDb();
connj = Koneksi.KoneksiDB.ConnectDb();
UpdateTabel();
UpdateTabelJabatan();
}

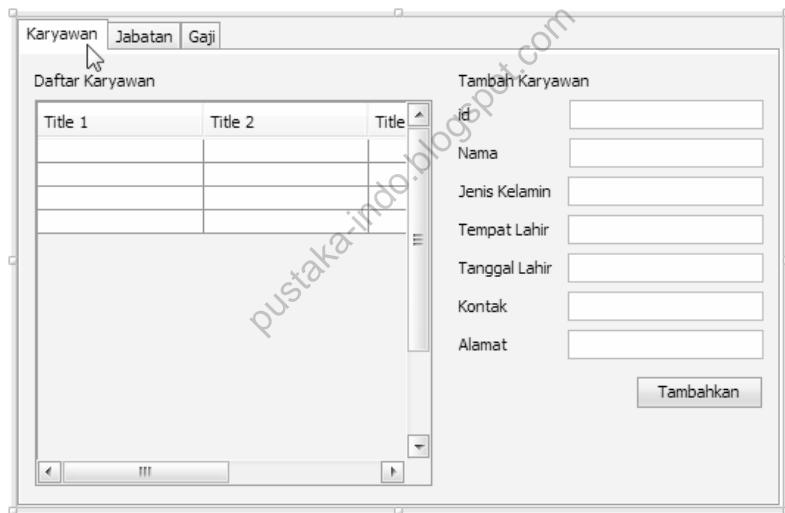
```

Menambahkan Data dalam Database

Untuk menambahkan data dalam database penggajian, Anda dapat menggunakan form dan tombol Tambahkan yang ada pada masing-masing panel karyawan dan jabatan.

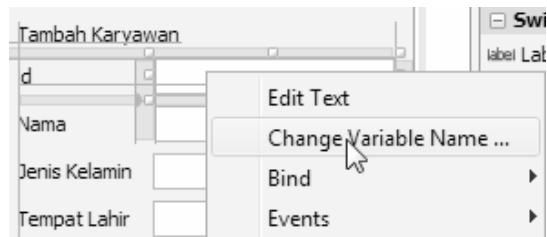
Menambahkan Data Karyawan

- Pada design FramePenggajian pilih panel Karyawan.



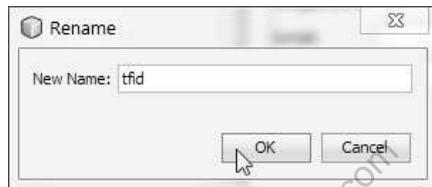
Memilih panel Karyawan

- Pilih textfield di sebelah id dan kemudian klik-kanan dan pilih **Change Variable Name ...**



Memilih Change Variable Name

3. Ubah nama variabel menjadi **tfid** kemudian klik **OK**.



Mengubah nama textfield

4. Ubah nama textfield lainnya sesuai dengan tabel berikut.

LABEL	NAMA TEXTFIELD
id	Tfid
Nama	Tfnama
Jenis Kelamin	Tfjenis
Tempat Lahir	Tftempat
Tanggal Lahir	Tftanggal
Kontak	Tfkontak
Alamat	Tfalamat
id_jabatan	Tfidjb

Daftar nama variabel textfield

5. Klik-ganda tombol **Tambahkan** untuk membuat method **onclick**.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

Method saat tombol dipilih

6. Masukkan kode berikut ke dalam method.

```
private void jButton1ActionPerformed(java.awt.event.
    ActionEvent evt)
{
    Connection conn = null;
    Statement stmt = null;

    String id = tfid.getText();
    String nama = tnama.getText();
    String jeniskelamin = tjenis.getText();
    String tempatlahir = ttempat.getText();
    String tanggallahir = ttanggal.getText();
    String kontak = tkontak.getText();
    String alamat = talamat.getText();
    String idjabatan = tjabatan.getText();

    try
    {
        conn = Koneksi.KoneksiDB.ConnectDb();
        stmt = conn.createStatement();
        stmt.executeUpdate("insert into karyawan values
            ('"+id+"', '"+nama+"', "
            + "'"+jeniskelamin+"', '"+tempatlahir+"',
            '"+tanggallahir+"', "
            + "'"+kontak+"', '"+alamat+"', '"+idjabatan+"');");
        JOptionPane.showMessageDialog
            (null,"Data berhasil ditambahkan!");
        UpdateTabel();
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

7. Jalankan program dan isi form dan klik tombol **Tambahkan** untuk melihat hasilnya.

The screenshot shows a Windows application window with three tabs at the top: 'Karyawan', 'Jabatan', and 'Gaji'. The 'Karyawan' tab is selected, displaying a table titled 'Daftar Karyawan' with columns: id_kary..., nama, jenis_k..., and tempat_. The table contains two rows: 1 Andi L Jakarta and 2 Joni L Bandung. To the right of the table is a panel titled 'Tambah Karyawan' containing fields for id, Nama, Jenis Kelamin, Tempat Lahir, Tanggal Lahir, Kontak, and Alamat. A 'Tambahkan' button is at the bottom right of this panel.

Menambahakan data

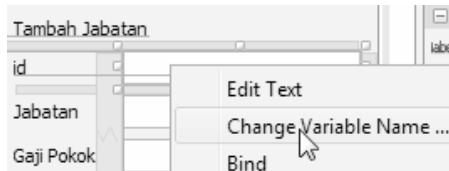
Menambah Data Jabatan

- Pada design FramePenggajian pilih panel Jabatan.

The screenshot shows the 'Jabatan' panel within the application frame. It has a table titled 'Daftar Jabatan' with columns: Title 1, Title 2, and Title. Below the table is a scroll bar. To the right is a panel titled 'Tambah Jabatan' with fields for id, Jabatan, Gaji Pokok, and Deskripsi. A 'Tambahkan' button is at the bottom right of this panel.

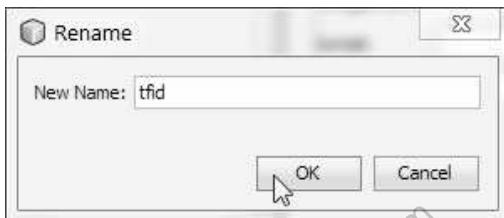
Memilih panel Jabatan

- Pilih textfield di sebelah id dan kemudian klik-kanan dan pilih Change Variable Name ...



Memilih Change Variable Name

3. Ubah nama variabel menjadi **tfidj** kemudian klik **OK**.



Mengubah nama textField

4. Ubah nama textField lainnya sesuai dengan tabel berikut.

Label	Nama TextField
id	tfidj
Jabatan	tfjabatanj
Gaji Pokok	tfgajij
Deskripsi	tfdeskripsij

Daftar nama variabel textField

5. Klik-ganda tombol **Tambahkan** untuk membuat method **onclick**.

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

Method saat tombol dipilih

6. Masukkan kode berikut ke dalam method.

```

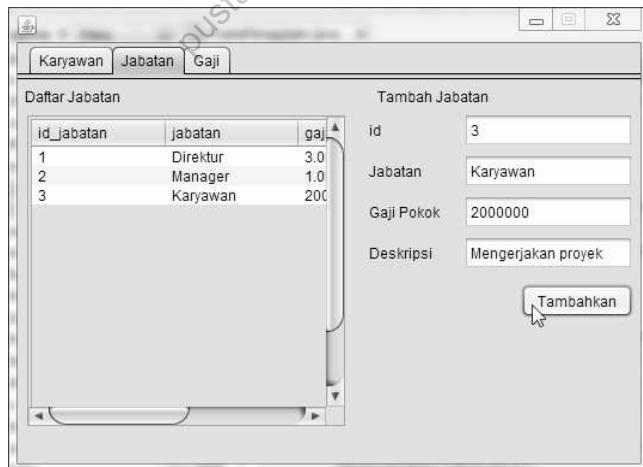
private void jButton2ActionPerformed
    (java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    Connection connj = null;
    Statement stmtj = null;

    String id = tfidj.getText();
    String jabatan = tfjabatanj.getText();
    String gaji = tfgajij.getText();
    String deskripsi = tfdeskripsij.getText();

    try
    {
        connj = Koneksi.KoneksiDB.ConnectDb();
        stmtj = connj.createStatement();
        stmtj.executeUpdate("insert into jabatan
            values ('"+id+"', '"+jabatan+"', "
            + "'"+gaji+"', '"+deskripsi+"');");
        JOptionPane.showMessageDialog
            (null,"Data berhasil ditambahkan!");
        UpdateTabelJabatan();
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

7. Jalankan program dan isi form dan klik tombol **Tambahkan** untuk melihat hasilnya.

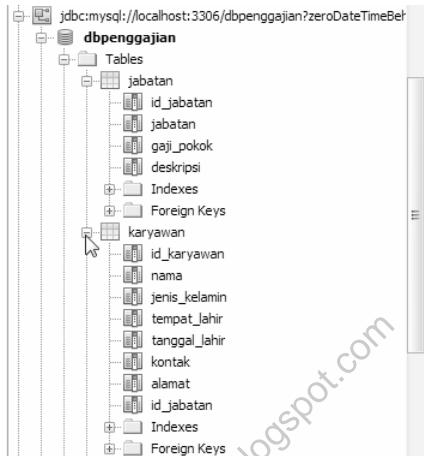


Menambahkan data

Menampilkan Gaji

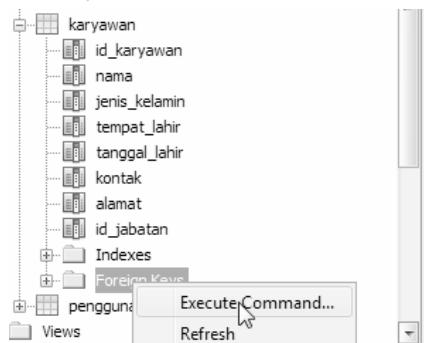
Untuk menampilkan gaji, pertama-tama Anda perlu menghubungkan kedua tabel dalam database dengan langkah-langkah berikut:

1. Pilih jendela **Services** kemudian perluas database **dbpenggajian**.



Database dbpenggajian

2. Pada tabel karyawan, klik-kanan folder **Foreign Keys** kemudian pilih **Execute Command...**

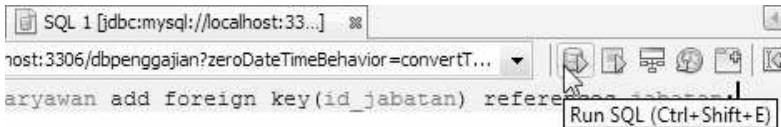


Memilih Execute Command...

3. Masukkan perintah berikut dalam editor SQL.

```
alter table karyawan  
add foreign key(id_jabatan) references  
jabatan(id_jabatan);
```

4. Jalankan perintah tersebut.



Menjalankan perintah SQL

Setelah perintah SQL dijalankan, Anda kemudian perlu mengatur panel gaji pada antarmuka aplikasi.

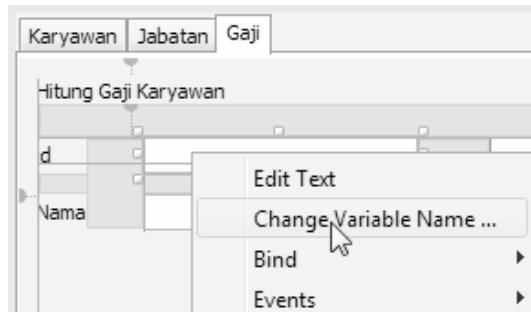
Lakukan langkah-langkah berikut:

1. Pilih panel **Gaji**.

A screenshot of a Java Swing application window titled "Hitung Gaji Karyawan". The window has three tabs at the top: "Karyawan", "Jabatan", and "Gaji", with "Gaji" being the active tab. On the left, there are two text fields: "id" and "Nama". To the right of these fields is a table with four columns labeled "Title 1", "Title 2", "Title 3", and "Title 4". At the bottom center is a button labeled "Tampilkan Gaji".

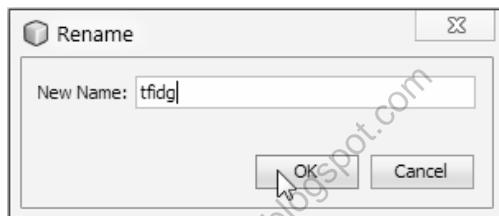
Memilih panel gaji

2. Pilih textfield di sebelah kanan id kemudian klik-kanan dan pilih **Change Variable Name...**



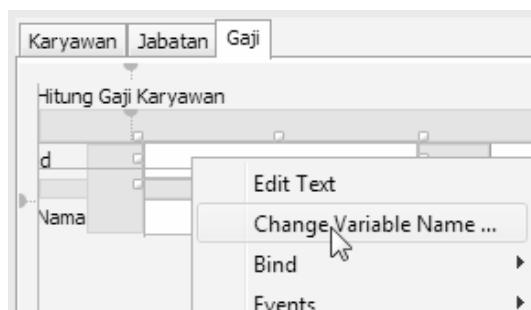
Memilih Change Variable Name...

3. Ubah nama menjadi **tfidg** kemudian klik **OK**.



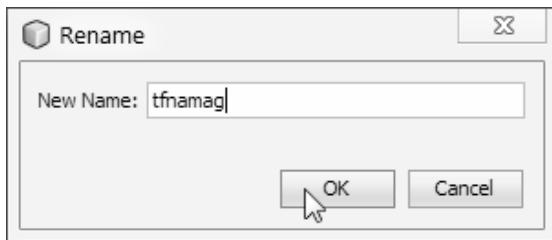
Mengubah nama variabel

4. Pilih textfield di sebelah kanan **Nama** kemudian klik-kanan dan pilih **Change Variable Name...**



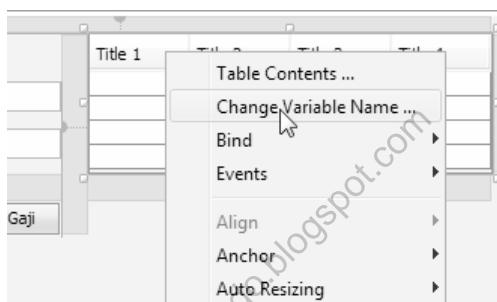
Memilih Change Variable Name...

5. Ubah nama menjadi **tfnamag** kemudian klik **OK**.



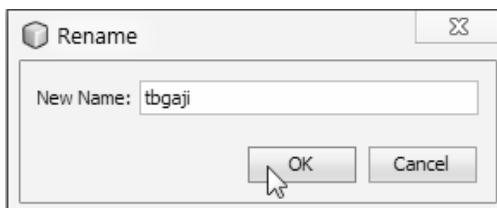
Mengubah nama variabel

6. Pilih tabel di sebelah kanan kemudian klik-kanan dan pilih **Change Variable Name...**



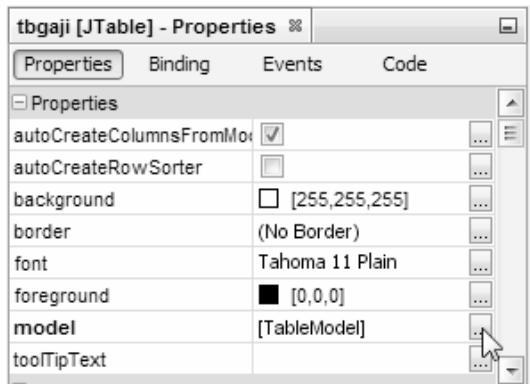
Memilih Change Variable Name...

7. Ubah nama menjadi **tbgaji** kemudian klik **OK**.



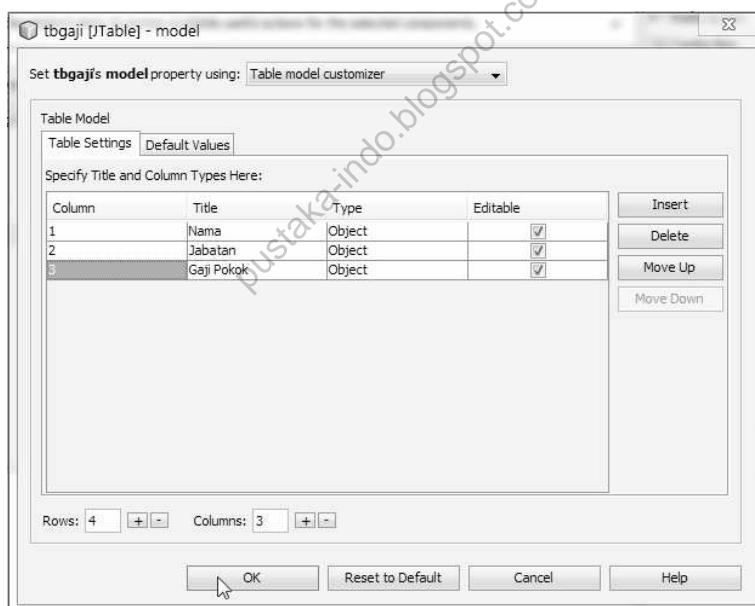
Mengubah nama variabel

8. Pilih tabel kemudian pilih tombol ... di sebelah kanan model pada jendela **Properties**.



Memilih model tabel

9. Pada jendela model tabel atur sehingga menjadi seperti pada gambar berikut.



Mengatur tabel

10. Klik-ganda tombol **Tampilkan Gaji** untuk membuat method **onclick**.

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

}
```

Method dibuat secara otomatis

11. Masukkan kode berikut ke dalam method.

```
private void jButton3ActionPerformed
        (java.awt.event.ActionEvent evt)
{
    Connection conng = null;
    Statement stmtg = null;

    String idg = tfidg.getText();
    String namag = tfnamag.getText();

    DefaultTableModel model =
        (DefaultTableModel)tbgaji.getModel();

    try
    {
        conng = Koneksi.KoneksiDB.ConnectDb();
        stmtg = connj.createStatement();
        String query = "select k.nama, j.jabatan,
            j.gaji_pokok"
            + " from karyawan k, jabatan j"
            + " where k.id_karyawan='"+idg+"'
            or k.nama='"+namag+"' ";
        ResultSet rs = stmtg.executeQuery(query);
        while (rs.next())
        {
            String namak=rs.getString("nama");
            String jabatank=rs.getString("jabatan");
            String gajik=rs.getString("gaji_pokok");

            model.addRow(new Object[] {namak, jabatank, gajik});
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

12. Aplikasi penggajian siap digunakan.

Tentang Penulis

Jubilee Enterprise, telah dipercaya oleh penerbit dan pembaca buku tanah air dalam satu dasawarsa ini. Hingga sekarang, tulisan-tulisannya yang diterbitkan dalam bentuk buku telah mencapai hampir 400 judul, sebagian besar bertema teknologi informasi dan kemudian disusul dengan tema psikologi & parenting, manajemen, fotografi, anak-anak, dan tema umum lainnya. Salah satu imprint Jubilee Enterprise adalah Jubilee Authors Companion yang merupakan sahabat bagi para penulis-penulis profesional yang ingin tetap berkarya.

Anda dapat memperoleh informasi lebih lanjut tentang Jubilee Enterprise lewat situs: www.thinkjubilee.com.

Catatan:

- Untuk melakukan pemesanan buku, hubungi Layanan Langsung PT Elex Media Komputindo:
Gramedia Direct
Jl. Palmerah Barat No. 29-37, Jakarta 10270
Telemarketing/CS: 021-53650110/111 ext: 3901/3902
Email: endang@gramediapublishers.com atau
yunisunarti@gramediabooks.com

Mengenal Java dan Database dengan NetBeans

Belajar Java yang dipadukan dengan database adalah investasi untuk masa depan. Oleh karena itu, bacalah buku ini yang mengajarkan Anda prinsip pemrograman Java dan database yang disusun menggunakan NetBeans.

Anda akan mempelajari prinsip Java dan database menggunakan MySQL. Semua tool dan sumber daya lainnya disediakan secara gratis lewat situs-situs penyedia Java, database, dan NetBeans sehingga Anda tidak perlu mengeluarkan biaya ekstra.

Di akhir bab, Anda akan belajar membuat aplikasi siap-pakai yang diciptakan menggunakan Java dan database.

Buku ini cukup sederhana pembahasannya dan bersifat tutorial sehingga segala proses instalasi hingga pembuatan aplikasi mengikuti alur langkah demi langkah yang mudah diikuti.

PT ELEX MEDIA KOMPUTINDO
Kompas Gramedia Building
Jl. Palmerah Barat 29-37, Jakarta 10270
Telp. (021) 53650110-53650111, Ext 3214
Webpage: <http://elexmedia.co.id>

Kelompok
Pemrograman
Keterampilan
<input checked="" type="checkbox"/> Tingkat Pemula
<input checked="" type="checkbox"/> Tingkat Menengah
<input type="checkbox"/> Tingkat Mahir
Jenis Buku
<input checked="" type="checkbox"/> Referensi
<input checked="" type="checkbox"/> Tutorial
<input type="checkbox"/> Latihan

