

HENDRI AHMADIAN, M.I.M
HENDY MIZUARDY, M.Sc
KHAIRAN AR, M.Kom

MAHIR

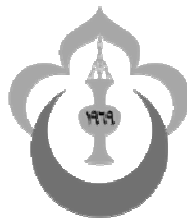
PEMROGRAMAN VISUAL DENGAN

JAVA



UNIMAL PRESS

MAHIR PEMROGRAMAN VISUAL DENGAN JAVA



**universitas
MALIKUSSALEH**

HENDRI AHMADIAN, M.I.M
HENDY MIZUARDY, M.Sc
KHAIRAN AR, M.Kom

MAHIR PEMROGRAMAN VISUAL DENGAN JAVA

UNIMAL PRESS

Judul: **MAHIR PEMROGRAMAN VISUAL DENGAN JAVA**

viii + 58 hal., 15 cm x 23 cm

Cetakan Pertama: September, 2017

Hak Cipta © dilindungi Undang-undang. *All Rights Reserved*

Penulis:

HENDRI AHMADIAN, M.I.M

HENDY MIZUARDY, M.Sc

KHAIRAN AR, M.Kom

Perancang Sampul dan

Penata Letak: Eriyanto

Pracetak dan Produksi: **Unimal Press**

Penerbit:

UNIMAL PRESS

Unimal Press

Jl. Sulawesi No.1-2

Kampus Bukit Indah Lhokseumawe 24351

PO.Box. 141. Telp. 0645-41373. Fax. 0645-44450

Laman: www.unimal.ac.id/unimalpress.

Email: unimalpress@gmail.com

ISBN: **978-602-1373-90-3**

Dilarang keras memfotocopy atau memperbanyak sebahagian atau seluruh buku ini tanpa seizin tertulis dari Penerbit

Kata Pengantar

Puji syukur kehadiran Allah SWT atas limpahan rahmat dan karunianya sehingga Buku Belajar Pemrograman Visual dengan Java telah dapat diselesaikan. Buku ini merupakan bahan ajar dalam mata kuliah Pemrograman Visual untuk mahasiswa semester V pada program studi Pendidikan Teknologi Informasi UIN Ar-Raniry Banda Aceh.

Terimakasih disampaikan kepada Dr. Hj, Chairan M. Nur, M.Ag selaku Ketua Program Studi Pendidikan Teknologi Informasi beserta sekretaris dan staf. Terimakasih juga disampaikan rekan-rekan Dosen dalam penyempurnaan buku ini dan semua pihak yang telah ikut membantu dalam penyelesaian buku ini. Semoga buku ini dapat memberi manfaat bagi mahasiswa di bidang informatika dan para praktisi yang menggeluti dunia teknologi informasi.

Kami menyadari masih terdapat kekurangan dalam buku ini untuk itu kritik dan saran terhadap penyempurnaan buku ini sangat diharapkan.

Banda Aceh, September 2017

Tim Penulis

This page is intentionally left blank

Daftar Isi

Kata Pengantar	v
Daftar Isi	vii
Daftar Tabel	viii
Daftar Gambar	viii
BAB I	
PENGANTAR JAVA DAN NETBEANS	1
1.1 Mengapa Java.....	1
1.2 Mengapa Netbeans.....	3
1.3 Menyiapkan Aplikasi.....	4
1.4 Memasang tools (aplikasi).....	5
BAB II	
BERKENALAN DENGAN JAVA	7
2.1 Penulisan Kode Bahasa Pemrograman Java	7
2.2 Identifier.....	8
2.3 Keyword.....	10
2.4 Access Modifier	10
2.5 Package dan Import.....	11
BAB III	
OPERATOR, PERCABANGAN, PERULANGAN DAN TIPE DATA.....	15
3.1 Operator.....	15
3.2 Percabangan.....	20
3.3 Perulangan.....	23
3.4 Tipe Data	24
BAB IV	
PEMROGRAMAN GUI NETBEANS.....	27
4.1 Pengantar	27
4.2 Java Foundation Class.....	27
4.3 AWT	28
4.4 Latihan 3.....	33
4.5 Javax Swing.....	34
4.6 Penjelasan singkat pengenalan komponen swing	38
4.6 Latihan GUI.....	43
BAB V	
KONEKSI DATABASE	53
5.1 Mengenal JDBC.....	53
5.2 Database Driver	54
5.3 Membuat Koneksi.....	54
5.4 Menambahkan pustaka MySql JDBC Driver	56
DAFTAR PUSTAKA.....	57
BIODATA PENULIS	58

Daftar Tabel

Tabel 1.	Keyword	10
Tabel 2.	Tipe data 1	25
Tabel 3.	Tipe data 2	25
Tabel 4.	Java Foundation Class	28
Tabel 5.	Menu palette 1	36
Tabel 6.	Menu palette 2	38

Daftar Gambar

Gambar 1.	Struktur penulisan kode bahasa java.....	7
Gambar 2.	Import	13
Gambar 3.	Operator 3.....	16
Gambar 4.	Operator 4.....	16
Gambar 5.	Operator 5.....	16
Gambar 6.	Operator 6.....	17
Gambar 7.	Operator 7.....	17
Gambar 8.	Operator 8.....	18
Gambar 9.	Operator 9.....	18
Gambar 10.	Percabangan 1	20
Gambar 11.	Percabangan 2	20
Gambar 12.	Percabangan 3.....	21
Gambar 13.	Percabangan 4.....	21
Gambar 14.	Percabangan 5.....	22
Gambar 15.	AWT 1	29
Gambar 16.	AWT 2	30
Gambar 17.	AWT 3	31
Gambar 18.	AWT 4	32
Gambar 19.	AWT 5	33
Gambar 20.	Latihan AWT 1	33
Gambar 21.	Latihan AWT 2	34
Gambar 22.	Tampilan area kerja Netbeans	35
Gambar 23.	Tampilan lengkap netbeans.....	36
Gambar 24.	Form manual.....	39
Gambar 25.	Form dengan netbeans.....	40
Gambar 26.	Button	41
Gambar 27.	Button icon	41
Gambar 28.	Memberi aksi button manual	42
Gambar 29.	Memberi aksi button netbeans.....	43

BAB I

PENGANTAR JAVA DAN NETBEANS

1.1 Mengapa Java1

Java adalah bahasa pemrograman yang akan digunakan dalam proses pembelajaran (praktikum) mata kuliah pemrograman visual bagi mahasiswa Pendidikan Teknologi Informasi UIN Ar-Raniry tahun ajaran 2015/2016. Sedangkan untuk berkenalan dengan bahasa pemrograman java, mahasiswa dapat merujuk pada milis-milis java berbahasa indonesia seperti: <http://www.jug.or.id>, <http://groups.yahoo.com/group/jug-joglosemar/> (pengguna java di jogja dan jawa tengah), <http://groups.yahoo.com/group/jug-geulis/> (pengguna java bandung), <http://tech.groups.yahoo.com/group/jug-bonek/> (pengguna java di jawa timur) ataupun forum resmi java-oracle berbahasa inggris di <http://www.oracle.com/technetwork/java/community/index.html>, sebagai pengantar singkat pengenalan bahasa pemrograman java, dapat dilihat pada penjelasan singkat di bawah ini.

Java merupakan bahasa pemrograman yang sejatinya berbasis object oriented programming (OOP) atau dalam terjemahan bahasa indonesia-nya lebih dikenal dengan pemrograman berorientasi objek (PBO). Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems, dan pada tahun 2010 java resmi diambil alih oleh oracle.

Dalam pembuatan kode (sintak)-nya, aplikasi-aplikasi berbasis java umumnya akan dikompilasi ke dalam p-code (*bytecode*) agar dapat dijalankan pada berbagai sistem operasi melalui java virtual machine (JVM), hal ini tidaklah berlebihan karena java dikenal dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan,

¹ Sumber: <https://id.wikipedia.org/wiki/Java> - diakses terakhir 14/09/2016, Pukul: 11.38

dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi, dengan beberapa kelebihan diantaranya:

- a. **Multiplatform.** Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip *tulis sekali, jalankan di mana saja*. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan di atas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut.
- b. **OOP** (*Object Oriented Programming* - Pemrogram Berorientasi Objek).
- c. **Perpustakaan Kelas Yang Lengkap**, Java terkenal dengan kelengkapan *library* (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
- d. **Pengumpulan sampah** otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas)

Untuk keperluan development (pengembangan) menggunakan bahasa pemrograman java ada beberapa paket platform yang dapat dimanfaatkan, yaitu²:

- a. Java SE (Standard Edition), yang khusus digunakan untuk pengembangan aplikasiaplikasi pada PC atau workstation.
- b. Java ME (Micro Edition), yaitu khusus digunakan untuk pengembangan aplikasi-aplikasi yang ada di perangkat mobile seperti HP, smartphome, PDA, tablet dsb.
- c. Java EE (Enterprise Edition), yaitu khusus digunakan untuk pengembangan aplikasi skala besar (enterprise), dan aplikasi web berbasis java.

1.2 Mengapa Netbeans

Netbeans adalah tool (alat) yang akan digunakan dalam menulis kode-kode pemrograman java dalam mata kuliah pemrograman visual ini. Netbeans merupakan salah satu alat Integrated Development Environment (IDE) yang sejatinya tidak hanya dipergunakan dalam pengembangan perangkat lunak berbasis java, akan tetapi juga dipergunakan oleh beberapa bahasa pemrograman seperti: C++ ataupun PHP. IDE secara sederhana dapat diartikan sebagai aplikasi komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Sebuah IDE, atau secara bebas dapat diterjemahkan sebagai Lingkungan Pengembangan Terpadu, setidaknya memiliki fasilitas:

- a. Editor, yaitu fasilitas untuk menuliskan kode sumber dari perangkat lunak.
- b. Compiler, yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubah dalam bentuk binari yang sesuai dengan bahasa mesin.
- c. Linker, yaitu fasilitas untuk menyatukan data binari yang beberapa kode sumber yang dihasilkan compiler sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program komputer yang siap dieksekusi.

² Sumber: E-book Pemrograman Java – Rosihan Ari Yuana, S.Si, M.Kom – dilihat terakhir 14092016 Pukul: 11.38

- d. Debugger, yaitu fasilitas untuk mengetes jalannya program, untuk mencari bug/kesalahan yang terdapat dalam program.

Sampai tahap tertentu IDE modern dapat membantu memberikan saran yang mempercepat penulisan. Pada saat penulisan kode, IDE juga dapat menunjukkan bagian-bagian yang jelas mengandung kesalahan atau keraguan.

1.3 Menyiapkan Aplikasi

Sebelum memulai menggunakan bahasa pemrograman java, ada beberapa langkah yang harus diperhatikan, yaitu:

- a. Mengunduh JDK

JDK adalah suatu paket yang terdiri dari: development tools, java runtime environment (JRE), sekumpulan library, java DB (Java Relational Database), dan komponen lain yang diperlukan dalam membangun aplikasi dengan bahasa pemrograman java, dalam mata kuliah ini penulisan kode program diarahkan ke J2SE yang dikhususkan untuk memanfaatkan komponen SWING. Untuk mengunduh JDK silahkan kunjungi link berikut ini:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Langkah-langkah menyiapkan aplikasi adalah:

- 1) Silahkan unduh JDK tersendiri (apabila kita akan mengunduh IDE Netbeans secara terpisah), atau silahkan unduh IDE Netbeans with JDK (apabila kita akan mengunduh keduanya)
- 2) Silahkan pilih platform JDK + Netbeans yang sesuai dengan sistem operasi yang terpasang di komputer kita
- 3) Sebenarnya setelah mendownload dua paket di atas kita sudah dapat menulis kode dengan bahasa pemrograman java, akan tetapi jika anda ingin menambahkan silahkan di unduh java SE documentation (berisi library-library yang biasanya dibutuhkan dalam penulisan kode bahasa pemrograman java)

- b. Mengunduh IDE netbeans

Paket IDE netbeans yang digunakan pada penulisan kode program java dalam mata kuliah pemrograman visual ini

sebenarnya dapat diunduh pada link oracle yang tertera di atas. Akan tetapi untuk lebih mengenal lingkup IDE netbeans serta untuk dapat memilah IDE netbeans yang akan diunduh, ada baiknya mengunduh netbeans pada link berikut ini:

<https://netbeans.org/downloads/index.html>, disarankan hanya mengunduh Java SE saja, walaupun tidak ada larangan untuk mengunduh bundel (bundles) netbeans secara keseluruhan (*all*).

1.4 Memasang tools (aplikasi)

Proses pemasangan tools java sangat sederhana. Cukup kita buka file JDK yang telah kita unduh, selanjutnya klik dua kali installernya kemudian klik next dan tunggu sampai selesai. Cara untuk pemasangan JDK + Netbeans tidak terlalu berbeda dengan pemasangan JDK, kelebihan JDK + Netbeans hanya paket netbeans yang sudah include (termasuk) di dalam paket JDK-nya.

Untuk langkah-langkah pemasangan JDK + Netbeans adalah³:

- a. Buka file JDK + Netbeans yang telah di unduh, selanjutnya klik dua kali.
- b. Klik button next.
- c. Kemudian pilih *"I accept the terms license agreement, Install Junit"*, selanjutnya klik next kembali.
- d. Tentukan (pilih) PATH (lokasi) pemasangan JDK.
- e. Tentukan (pilih) PATH (lokasi) pemasangan Netbeans IDE.
- f. Hilangkan tanda centang pada tulisan *"check for update"*, selanjutnya klik button install dan tunggu sampai selesai ^{4*}.



³ Sumber: E-book Pemrograman Java – Rosihan Ari Yuana, S.Si, M.Kom – dilihat terakhir 14092016 Pukul: 11.38

⁴ *Langkah-langkah pemasangan JDK + Netbeans IDE di atas dapat berubah sesuai dengan update terakhir dari versi JDK + Netbeans IDE-nya, walaupun tidak akan terlalu berbeda. Silahkan disesuaikan.

This page is intentionally left blank

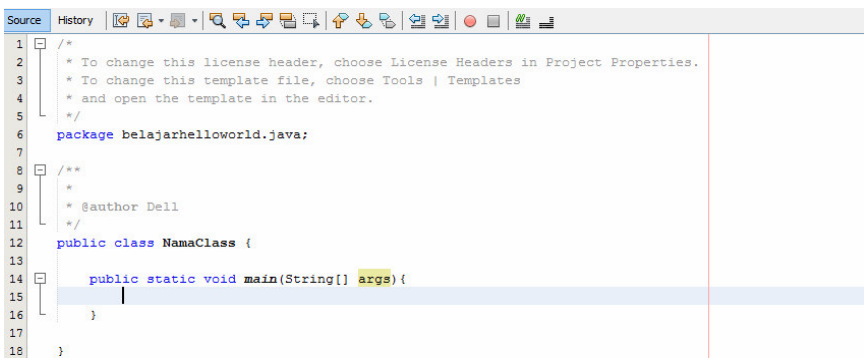
BAB II

BERKENALAN DENGAN JAVA

2.1 Penulisan Kode Bahasa Pemrograman Java

Untuk memudahkan penulisan kode bahasa pemrograman java, kita bisa menggunakan berbagai aplikasi editor yang kita kenal misalnya notepad, notepad ++, bahkan editor yang lebih canggih seperti eclipse dan netbeans IDE. Dalam penulisan bahasanya, ada beberapa hal yang perlu kita perhatikan yaitu, Penulisan kode bahasa pemrograman java bersifat case-sensitive, artinya bahasa pemrograman java membedakan antara huruf besar dan huruf kecil ataupun tanda baca yang kita gunakan.

Secara sederhana struktur penulisan bahasa pemrograman java dimulai dengan penulisan **public class nama_class**⁵ yang diikuti kurung kurawal buka ({) dan diakhiri dengan kurung kurawal tutup (}). Selain itu di dalam class harus dibuat main programnya yaitu, **public static void main(string[] args)**. Main program berfungsi sebagai tempat untuk mengeksekusi (run) runutan kode yang telah ditulis sebelumnya. Gambaran singkat struktur penulisan kode bahasa pemrograman java dapat dilihat pada gambar di bawah ini:



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package belajarhelloworld.java;
7
8  /**
9   *
10  * @author Dell
11  */
12  public class NamaClass {
13
14      public static void main(String[] args) {
15
16      }
17
18  }
```

Gambar 1. Struktur penulisan kode bahasa java

⁵ * Apabila anda menulis kode bahasa pemrograman java menggunakan editor sederhana seperti notepad. Ketika penyimpanan file-nya, nama file dan nama class harus sama (misal: nama class HelloWorld, nama file juga harus HelloWorld.java

2.2 Identifier

Penulisan nama-nama elemen pemrograman java ada aturan-aturan (kaidah) yang harus dipatuhi, seperti penamaan identifier harus berbeda dengan nama keyword. Apabila aturan ini dilanggar maka program yang kita buat tidak dapat di eksekusi.

a. Identifier

Identifier adalah nama untuk mengidentifikasi elemen program atau nama-nama yang bisa dideklarasikan dalam bahasa java tetapi bukan merupakan bagian dari keyword. Yang termasuk dalam identifier antara lain: penamaan kelas, penamaan variabel/property, interface, dan method. Penamaan identifier dalam bahasa java diatur dalam beberapa aturan⁶:

- 1) Aturan dari compiler untuk menentukan apakah nama identifiers diperbolehkan atau tidak. Aturan-aturan dari compiler sebagai berikut:
 - a) Aturan pertama sudah kita bahas sebelumnya adalah semua keyword java tidak boleh digunakan sebagai identifiers.
 - b) Identifiers harus diawali oleh huruf, simbol mata uang dolar(\$) atau karakter penghubung underscore (_). Angka tidak boleh digunakan sebagai karakter pertama identifiers.
 - c) Setelah karakter pertama, berikutnya boleh diikuti oleh huruf, simbol mata uang dolar, karakter penghubung, dan angka.
 - d) Tidak ada pembatasan panjang identifiers
 - e) Identifiers di java bersifat case-sensitif, foo dengan Foo adalah dua buah identifiers berbeda.
 - f) Nama public class harus sama persis dengan nama file .java
- 2) Konvensi penamaan identifiers dari Sun yang biasa disebut sebagai "Java Code Convention". Berikut beberapa aturan yang ada pada java code convention:
 - a) Class dan Interface selalu diawali dengan huruf besar. Setiap kata selalu diawali dengan huruf besar untuk memudahkan pembacaan. Gaya ini biasa disebut dengan "Camel Case". Contohnya: Runnable, HashMap, ArrayList dan

⁶ Sumber: Java Desktop – Ifnu Bima

seterusnya. Selain itu, class haruslah merupakan kata benda, bukan kata sifat atau kata kerja.

- b) Method selalu diawali dengan huruf kecil. Setiap kata setelah huruf pertama diawali dengan huruf besar. Method haruslah kata kerja untuk menandakan bahwa method ini melakukan suatu kegiatan / aksi. Contohnya : `getIndex`, `setIndex`, `println`, `paint`, dan seterusnya.
- c) Variabel menggunakan camel case yang diawali dengan huruf kecil, seperti method. Variabel sebaiknya pendek, jelas, terdengar enak dan kata benda. Contohnya : `index`, `panjang`, `lebar`, `indexPertama` dan seterusnya.
- d) Konstanta di java dibuat dengan mendeklarasikan sebuah variabel sebagai `static` dan `final`, semua hurufnya adalah huruf besar yang antar kata dipisahkan oleh simbol underscore (`_`). Contohnya : `FRAME_WIDTH`, `ERROR_MESSAGE` dan seterusnya.

3) Standard penamaan JavaBean.

Konsep javabean dibuat sebagai dasar dari komponen-komponen java. Pemanfaatan javabean dalam lingkup java dapat kita temukan pada penggunaan komponen-komponen swing yang telah divisualisasikan dalam IDE Netbeans. Karena komponen-komponen javabean access-modifiernya bersifat `private`, maka perlu dibuatkan method untuk mengakses propertisnya, yang biasa disebut dengan method setter dan getter dengan aturan sebagai berikut:

- a) Kalau tipe data properties bukan boolean maka method untuk mengakses properties diawali dengan `get`. misalnya `getWidth`, `getSize`, `getIndex` dan seterusnya.
- b) Kalau tipe data properties adalah boolean maka method untuk mengakses properties diawali dengan `is`. Misalnya `isEmpty`, `isRunning` dan seterusnya.
- c) Semua method setter harus diawali dengan `set`. Misalnya `setSize`, `setIndex`, `setWidth` dan seterusnya.
- d) Nama method diturunkan dari nama variabel yang diberi awalan `get`, `set` atau `is`. Aturan penulisan camel case berlaku untuk method getter dan setter.
- e) Method setter harus `public`, `return void` dengan satu parameter yang tipe datanya sama persis dengan tipe data variabel.

- f) Method setter harus public, return tipe data yang sama dengan tipe data variabel, dantampa parameter.
- g) JavaBean harus mempunyai default constructor, yaitu constructor yang tidak mempunyai parameter sama sekali.

2.3 Keyword⁷

Kosa kata (keyword) dalam aturan bahasa pemrograman java, tak ubahnya seperti aturan pembentukan kosa kata bahasa indonesia yang bertujuan untuk membentuk suatu kalimat dengan SPOK ataupun bahasa inggris dengan gramer-nya. Dalam bahasa pemrograman java ada beberapa keyword yang harus diperhatikan yang tentu saja penulisannya harus dibedakan dengan penulisan kode untuk identifier (misalnya, menyamakan nama identifier nama variabel dengan keyword). Kumpulan keyword java disajikan pada tabel 1.

Tabel 1. Keyword

abstract	Boolean	break	byte	case	catch
char	class	const	continue	default	do
double	else	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
assert	enum				

2.4 Access Modifier⁸

public, *protected*, *default* dan *private* adalah empat buah level access modifier, fungsi dari access modifier adalah mengatur bagaimana bagian-bagian kode java diakses dari bagian yang lain. Ada bagian yang boleh diakses oleh siapapun karena kode di dalamnya sudah dirancang untuk itu, ada juga bagian yang hanya boleh diakses oleh class yang sama karena memang kodenya

⁷ Sumber: Java Desktop – Ifnu Bima

⁸Sumber: Java Desktop – Ifnu Bima

tergantung dengan bagian lain dari class tersebut dan tidak bisa digunakan oleh bagian lain.

Access modifier public menandakan bisa diakses oleh siapapun tanpa batasan. *Access modifier protected* bisa diakses oleh class turunannya dan class-class lain yang berada dalam package yang sama. *Access modifier default* tidak memerlukan keyword, kalau tidak ada salah satu dari tiga access modifier lain maka yang digunakan adalah *access modifier default*. Kalau *access modifier default* digunakan, maka hanya class dari package yang sama saja yang bisa mengakses, termasuk class itu sendiri. Yang terakhir adalah *access modifier private* yang hanya mengijinkan diakses oleh class yang sama.

2.5 Package dan Import⁹

Bahasa pemrograman java dilengkapi dengan fasilitas pembuatan paket (*package*) yang biasa digunakan untuk mengorganisasikan (mengelompokkan) class-class java. Pengelompokan class ini dimaksudkan untuk meletakkan modul-modul pemrograman yang sejenis, yang biasanya disebut dengan pustaka (library). Selain itu java juga dikenal dengan salah satu bahasa pemrograman yang memiliki pustaka yang sangat lengkap hal ini tidaklah mengeherankan karena java mampu mendefenisikan class-class yang dibangun dengan sangat baik, pengelompokkan class-class ini disebut dengan file jar. Untuk memakai suatu class di java, dilakukan dengan cara import class tersebut. berikut penjelasan singkat *package* dan *import*.

a. Package

Pakcage dalam java adalah sebuah mekanisme untuk mengorganisasi penamaan class ke dalam modul-modul. Class yang mempunyai fungsionalitas serupa dan kemiripan cukup tinggi biasanya diletakkan dalam satu package yang sama. Kalau ingin menggunakan class lain yang berada dalam package yang berbeda harus diimport terlebih dahulu menggunakan keyword *import*. Class-class dalam package agar mudah didistribusikan biasanya diletakkan dalam satu buah jar yang pada dasarnya adalah sebuah file zip saja.

Paragraf di atas menerangkan hubungan antara *package*, *import* dan jar dalam aplikasi java. Selanjutnya kita akan belajar

⁹Sumber: Java Desktop – Ifnu Bima

bagaimana membuat *package* dan mengimport class dari *package* lain, kemudian membuat file jar dari class yang sudah dicompile. Selain bertujuan untuk mengorganisasi *class*, *package* juga digunakan untuk menghindari penamaan class yang bisa bertabrakan dalam aplikasi Java. Kalau kita membuat sebuah class dengan nama yang sangat umum, misalnya class *User*, kemungkinan besar *developer* lain akan membuat class dengan nama yang sama, nah bagaimana kalau kita menggunakan library yang didalamnya terdapat nama class yang sama dengan class yang kita buat? class manakah yang akan dipilih oleh Java? masalah penamaan ini dipecahkan dengan menggunakan *package*.

Package dimana sebuah class berada akan menjadi bagian dari nama lengkap sebuah class, misalnya class *String* sebenarnya nama lengkapnya adalah *java.lang.String* karena class *String* berada dalam *package* *lang.util*. Untuk menghindari penamaan class yang sama, setiap *developer* disarankan untuk menggunakan *package* yang *unique* untuk aplikasi yang digunakan.

Misalnya ada 2 buah class dengan nama *ClassA*, yang satu berada di dalam *package* *a.b.c* sehingga nama lengkapnya adalah *a.b.c.ClassA* sedangkan satu lagi berada di dalam *package* *d.e.f* sehingga nama classnya adalah *d.e.f.ClassA*. Bagaimana menjamin nama *package* yang *unique*? gunakan nama domain website institusi anda, maka anda akan mendapatkan nama *package* yang *unique*.

Ada sebuah aturan tidak tertulis dari Sun untuk menggunakan nama domain institusi yang dibalik untuk digunakan sebagai *package* diikuti dengan nama aplikasi. Misalnya kita bekerja untuk perusahaan PT coding sejahtera yang mempunyai website *codings.com*, kemudian kita membuat aplikasi keuangan yang disingkat dengan AKCS (aplikasi keuangan coding sejahtera) maka kita akan membuat *package* dengan *com.codings.akcs*. Bagaimana kalau kita membuat aplikasi *opensource*? gunakan nama domain dimana project tersebut dihosting. Misalnya untuk class-class yang digunakan di buku ini akan menggunakan *package* *com.googlecode.projecttemplate.pos*, hal ini karena kode dalam buku ini dihosting di *project-template.googlecode.com* dan nama aplikasinya adalah *pos* (*point of sales*).

Package pada dasarnya adalah struktur folder untuk meletakkan kode file java, tetapi tidak bisa sembarangan menyusun struktur folder ini, hal ini dimaksudkan agar kode lebih rapi, teratur dan tidak bercampur campur. Berikut adalah contoh pembuatan *package* di IDE netbeans. Klik kanan → new → *package*.

b. Import

Import digunakan untuk menyederhanakan penulisan class. Tanpa menggunakan import kita harus menuliskan nama lengkap class beserta packagenya. Dengan menggunakan import, kita mendeklarasikan di mana class yang digunakan tersebut berada sehingga selanjutnya tidak perlu lagi menuliskan nama package dari sebuah class.

Ada dua pengecualian di mana import tidak diperlukan, pertama untuk class-class yang berada dalam package yang sama dan kedua adalah class-class yang berada dalam package `java.lang`. Pada IDE netbeans pendefinisian import disesuaikan dengan kebutuhan class yang akan dipakai, dan biasanya import dilakukan ketika akan memulai atau sedang melakukan penulisan pemrograman java. Berikut tampilan import pada IDE netbeans.

```
package unbrah.ac.id.siakad.mahasiswa;

import javax.swing.JTextField;

/**
 *
 * @author khairan ar
 */
public class Mahasiswa {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        JTextField nim = new JTextField();
    }
}
```

Gambar 2. Import

Untuk lebih menyederhanakan import, dapat dilakukan dengan memberikan tanda bintang setelah nama paket class yang akan digunakan didefinisikan. Misalnya, `import javax.swing.*;`



BAB III

OPERATOR, PERCABANGAN, PERULANGAN DAN TIPE DATA

3.1 Operator¹⁰

Secara umum operator dimiliki oleh seluruh jenis bahasa pemrograman, tidak terkecuali java. Operator digunakan untuk mengolah/mengubah nilai dari suatu variabel. Ada beberapa operator yang digunakan dalam pemrograman java, seperti: operator aritmatika, bitwise, assignment, kondisi, instanceof, bit shift, operator logika, dan operator relasi. Namun dalam modul ini hanya akan dibahas beberapa jenis operator yakni, operator aritmatika, kondisi, logika, dan operator relasi.

a. Operator aritmatika

Operator aritmatik pasti sudah sangat familiar, karena operator ini tidak jauh berbeda dengan operator aritmatik yang ada di pelajaran matematika. Simbol yang digunakan dalam java adalah :

- + (penambahan)
- - (pengurangan)
- * (perkalian)
- / (pembagian)

Penggunaan keempat operator di atas sangat standard, sama persis seperti di pelajaran matematika.

Java juga mengenal operator modulus (%) atau biasa dikenal sebagai pembagi sisa. Misalnya 10 % 5 hasilnya adalah 0 (10 dibagi 5 sisa 0), atau 10 % 3 hasilnya adalah 1 (10 dibagi 3 sisa 1). Operator + (plus) selain digunakan sebagai penambahan juga digunakan sebagai penggabungan String (String concatenation). Contohnya :

```
public static void main(String[] args) {  
    // TODO code application logic here  
    String s = "ini " + " String";  
}
```

¹⁰Sumber: Java Desktop – Ifnu Bima

Gambar 3. Operator 3

String *concatenation* menjadi sedikit kompleks kalau dirangkai dengan angka, contohnya:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    String s = "ini " + " String";  
    int a = 10;  
    long b = 100;  
    System.out.println(s + a + b);  
}
```

Gambar 4. Operator 4

Kalau kita ingin menjumlahkan variable a dan b terlebih dahulu sebelum digabungkan dengan String s, maka kita bisa meletakkan kurung di antara variabel a dan b, seperti kode di bawah ini :

```
public static void main(String[] args) {  
    // TODO code application logic here  
    String s = "ini " + " String";  
    int a = 10;  
    long b = 100;  
    System.out.println(s + (a + b));  
}
```

Gambar 5. Operator 5

Java, seperti halnya C/C++ juga mengenal operator increment dan decrement :

- ++ (increment (postfix dan prefix))
- -- (decrement (postfix dan prefix))

Kedua operator ini disebut juga unary operator karena hanya membutuhkan satu operand saja, sedikit berbeda dengan operator aritmatik lain yang membutuhkan dua buah operand. Operator increment akan menaikkan nilai dari operand sebesar satu

satuan, sedangkan decrement akan menurunkan nilai dari operand sebanyak satu satuan. Kedua operator ini adalah bentuk lebih pendek lagi dari operator gabungan ($+=$ dan $-=$). Berikut ini contoh bentuk paling panjang hingga paling pendek :

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int x = 10;  
    x = x + 1; //bentuk paling panjang  
    x += 1; //menggunakan operator gabungan +=  
    x++; //menggunakan operator increment  
    int y = 100;  
    y = y - 1; //bentuk paling panjang  
    y -= 1; //menggunakan operator gabungan -=  
    y--; //menggunakan operator decrement  
}
```

Gambar 6. Operator 6

Operator *increment* dan *decrement* bisa diletakkan setelah operand (*postfx*) maupun sebelum operand (*prefx*), kedua bentuk ini sama-sama menaikkan atau menurunkan nilai operand sebanyak satu satuan, tetapi evaluasi terhadap operasinya mempunyai nilai berbeda. Hasil evaluasi operator increment yang diletakkan setelah operand sama dengan nilai operand sebelum dievaluasi, sedangkan hasil evaluasi operator increment yang diletakkan sebelum operand adalah lebih banyak satu satuan daripada nilai operand sebelum dievaluasi, berikut contohnya:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int x = 10;  
    System.out.println("x++ = " + x++);  
    System.out.println("Setelah evaluasi, x = " + x);  
    x = 10;  
    System.out.println("++x = " + ++x);  
    System.out.println("Setelah evaluasi, x = " + x);  
}
```

Gambar 7. Operator 7

b. Operator Kondisi

Operator kondisi adalah *ternary* operator, artinya operator ini mempunyai tiga buah operand. Operator kondisi akan mengevaluasi suatu kondisi yang nilainya benar (true) atau salah (false), kemudian meng-*assign* suatu nilai ke dalam variabel. Dengan kata lain, operator ini mirip dengan if tetapi bertujuan untuk mengassign nilai ke sebuah variabel berdasarkan suatu kondisi. Operator kondisi menggunakan simbol ? (tanda tanya) dan : (titik dua). Contohnya sebagai berikut :

```
public static void main(String[] args) {
    // TODO code application logic here
    int x = 100;
    String s = (x < 10) ? "kurang dari sepuluh" : "lebih dari sama dengan sepuluh";
}
```

Gambar 8. Operator 8

Kode di atas artinya: kalau variabel x kurang dari sepuluh maka assign String "kurang dari sepuluh" ke dalam variabel s, sebaliknya: kalau x lebih dari sama dengan sepuluh maka assign String "lebih dari sama dengan sepuluh" ke dalam variabel s.

Operator kondisi bisa dirangkai untuk lebih dari satu kondisi, misalnya kalau kondisi pertama benar *assign* suatu nilai ke variabel, kalau salah tes lagi kondisi kedua, kalau benar assign nilai yang lain ke variabel dan akhirnya kalau kedua kondisi juga salah assign nilai terakhir ke variabel. Struktur kodenya seperti di bawah ini :

```
public static void main(String[] args) {
    // TODO code application logic here
    int x = 100;
    String s = (x < 10) ? "kurang dari sepuluh" : (x > 100)
    ? "lebih dari seratus"
    : "lebih besar sama dengan sepuluh dan kurang dari sama dengan seratus";
}
```

Gambar 9. Operator 9

Kode di atas artinya: kalau variabel x kurang dari sepuluh assign String "kurang dari sepuluh" ke variabel s, selainya: kalau x lebih dari seratus assign String "lebih dari seratus" ke dalam variabel s, kalau kedua kondisi salah: assign String "lebih besar sama dengan sepuluh dan kurang dari sama dengan seratus" ke dalam variabel s.

c. Operator Logika

Di kurikulum ilmu komputer, operasi logika diajarkan dalam satu semester penuh di mata kuliah Logika Matematika, kenapa sampai satu matakuliah sendiri untuk mempelajari logika matematika? Hal ini dikarenakan operasi logika sangat penting dalam sebuah aplikasi, membuat rangkaian logika yang rapi dan mudah dimengerti untuk aplikasi yang rumit adalah pekerjaan yang cukup sulit. Terkadang kita akan mendapati kode dengan banyak sekali kondisi sehingga yang membaca kode menjadi sangat bingung bagaimana alur aplikasi berjalan. Pengalaman saya juga sering mendapati cukup banyak bug aplikasi diakibatkan oleh operasi logika yang salah. Jadi mari kita bahas operator logika di Java dengan keseriusan yang tinggi. Ada enam logical operator yang bisa digunakan dalam Java :

- & operator dan, di bab sebelumnya operator ini juga digunakan sebagai operator bitwise kalau operandnya bertipe angka
- | operator or, di bab sebelumnya operator ini juga digunakan sebagai operator bitwise kalau operandnya bertipe angka
- ^ operator xor, di bab sebelumnya operator ini juga digunakan sebagai operator bitwise kalau operandnya bertipe angka
- ! operator logika negasi
- && operator singkat dan
- || operator singkat or

d. Operator Relasi

Operator relasi selalu menghasilkan data boolean (true atau false), operator ini sering digunakan untuk mengecek sebuah kondisi dan diletakkan di dalam percabangan (if). Operator relasi ada enam jenis, antara lain :

- < lebih kecil
- > lebih besar
- <= lebih kecil sama dengan
- >= lebih besar sama dengan
- == perbandingan
- != tidak sebanding

3.2 Percabangan¹¹

Kontrol terhadap aliran eksekusi aplikasi adalah salah satu feature utama dari sebuah bahasa pemrograman. Java menyediakan beberapa cara untuk melakukan kontrol aliran eksekusi aplikasi. if dan switch adalah dua jenis yang paling sering digunakan, selain itu ada mekanisme exception yang bisa mempengaruhi aliran eksekusi program. Dalam bab ini kita akan membahas ketiganya secara intensif, terutama bagaimana best practice penggunaan exception.

a. Percabangan IF

Statement if digunakan dalam aplikasi untuk mengatur aliran eksekusi program berdasarkan kondisi tertentu. Sintaks dasarnya adalah sebagai berikut:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int x = 10;  
    if(x == 10 ) {  
        System.out.println("lakukan sesuatu di sini kalau " +  
            "kondisi di dalam if bernilai true");  
    }  
}
```

Gambar 10. Percabangan 1

Ada juga bentuk else if yang dirangkai untuk mengetes beberapa kondisi, seperti di bawah ini:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int nilai = 78;  
    if(nilai < 40 ) {  
        System.out.println("nilai < 40 = D");  
    } else if (nilai >= 40 && nilai < 60) {  
        System.out.println("nilai >= 40 && nilai < 60 = C");  
    } else if (nilai >= 60 && nilai < 70) {  
        System.out.println("nilai >= 60 && nilai < 80 = B");  
    } else if (nilai >= 80 && nilai < 100) {  
        System.out.println("nilai >= 80 && nilai <=100 = A");  
    } else {  
        System.out.println("range nilai harus antara 0 - 100");  
    }  
}
```

Gambar 11. Percabangan 2

¹¹Sumber: Java Desktop – Ifnu Bima

Tanda kurung kurawal tidak diperlukan kalau blok kode yang dieksekusi di dalam if hanya satu baris saja, tetapi praktek ini tidak dianjurkan karena menimbulkan kebingungan, misalnya:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int x = 0;  
    if(x >= 0)  
        System.out.println("nilai positif");  
}
```

Gambar 12. Percabangan 3

Perhatikan bahwa kalau if tidak diikuti dengan tanda kurung kurawal maka yang dianggap sebagai blok kode di dalam if adalah satu statement tepat di setelah if, kalau ada beberapa statement setelah if, maka sisanya tidak dianggap bagian dari blok kode if.

b. Switch

Switch biasanya digunakan kalau kita ingin mengevaluasi nilai dari sebuah variabel yang mempunyai banyak kemungkinan. Bentuk sintaks switch lebih singkat dan bersih dibanding kalau menggunakan if-else if-else. Hingga Java 6 switch hanya bisa digunakan kalau variabelnya bertipe angka atau enum. Berikut ini bentuk dasar dari switch:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int x = 10;  
    switch(x) {  
        case 1 :  
            System.out.println("satu");  
            break;  
        case 2 :  
            System.out.println("dua");  
            break;  
        default :  
            System.out.println("bukan satu atau dua");  
    }  
}
```

Gambar 13. Percabangan 4

Dalam switch terdapat konsep yang disebut dengan fall-through, konsep ini kalau kita bayangkan mirip dengan efek domino, kalau kartu pertama rubuh akan menimpa kartu berikutnya menimpa kartu berikutnya lagi seterusnya hingga ada yang menghentikan efek dominonya. Lihat kode di atas, terdapat keyword break yang digunakan untuk menghentikan fall-through. Kalau kita hilangkan break dari kode di atas, maka kalau ada satu kondisi dari case yang dipenuhi maka kode di dalam case di bawahnya akan terus dieksekusi hingga ketemu break atau blok kode switch berakhir. Agar memudahkan pemahaman tentang fall-through ini, perhatikan contoh dibawah ini:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int x = 1;  
    switch(x){  
        case 1 :  
            System.out.println("satu");  
        case 2 :  
            System.out.println("dua fall-through");  
        case 3 :  
            System.out.println("tiga fall-through");  
        case 4 :  
            System.out.println("empat fall-through");  
        default :  
            System.out.println("default fall-through");  
    }  
}
```

Gambar 14. Percabangan 5

if dan switch adalah flow control yang digunakan kalau aplikasi berjalan dengan baik, artinya flow control oleh if dan switch digunakan untuk mengendalikan aliran eksekusi aplikasi kalau semesta berjalan seperti yang dikehendaki. Java mempunyai cara untuk mengontrol aliran aplikasi kalau terjadi error di dalam aplikasi, namanya adalah exception.

c. Exception

Aplikasi biasanya tidak pernah berjalan mulus 100% setiap waktu, justru banyak sekali kejadian dimana aplikasi mengalami kondisi yang tidak diinginkan, entah ada masalah dengan data yang

tidak benar, input/output yang tiba-tiba terputus, koneksi ke database yang tidak lancar hingga error yang sengaja dibuat oleh user untuk mencari celah keamanan aplikasi. Kode yang ditulis untuk handle masalah dan membuat aplikasi tangguh dalam segala medan sangat penting perannya, bahkan menurut penelitian jumlahnya mencapai 80% dari total kode yang dibuat developer. Mengingat pentingnya penanganan kesalahan, Java menyediakan feature penanganan kesalahan yang elegan, mudah digunakan dan tepat sasaran. Feature ini disebut dengan exception handling, dengan feature ini kita tidak perlu mengetest hasil kembalian dari suatu fungsi untuk mengetahui apakah ada error apa tidak, dengan begitu pengecekan kesalahan bisa lebih elegan dan jelas.

d. Sintaks Exception

Keyword yang digunakan dalam exception handling ada tiga : *try*, *catch*, *finally*, *throw* dan *throws*. Kita akan belajar bagaimana menggunakan *try* dan *catch* terlebih dahulu baru kemudian kita belajar tentang *finally*. Contoh-contoh kode akan menggunakan beberapa class input/output untuk membaca *file*, tidak perlu bingung bagaimana caranya menggunakan I/O, kita akan membahas I/O dalam satu bab tersendiri, yang penting pahami dahulu bagaimana sintaks exception handling serta best practicenya. Karena pentingnya konsep ini, sebaiknya ulangi membaca bab ini jika belum paham benar bagaimana konsepnya.

3.3 Perulangan

Perulangan atau dalam bahasa java dikenal dengan istilah iterasi memiliki beberapa statemen, seperti: *while*, *do-while*, dan *for*.

a. Statemen perulangan (*loop*) *while*

Merupakan statemen perulangan yang fundamental dalam pemrograman java, statemen ini akan mengeksekusi secara berulang statemen atau blok statemen selama ekspresi pengendali bernilai benar atau *true*. Format umum perulangan *while*:

```
while (kondisi) {  
    //statemen atau blok statemen  
}
```

b. Statemen perulangan (*loop*) *do-while*

Merupakan statemen perulangan yang hanya mengeksekusi statemen atau blok statemen minimal sebanyak sekali, hal ini

dikarenakan ekspresi kondisi berada dibawah loop. Proses perulangan pada do-while pertama-tama mengeksekusi statemen atau blok statemen selanjutnya akan dilakukan evaluasi pada ekspresi kondisionalnya. Selama ekspresi bernilai benar atau true, maka loop akan berulang. Berikut adalah format umum perulangan do-while:

```
do {  
    //statemen atau blok statemen  
}while(kondisi)
```

c. Statemen perulangan (Loop) for

Format umum dari perulangan for adalah sebagai berikut:

```
for (inisialisasi; kondisi; iterasi){  
    //statemen atau blok statemen  
}
```

Yang perlu diingat dalam perulangan for ini adalah, jika hanya ada satu statemen yang akan dieksekusi secara berulang, maka kurung kurawal ({}) tidak diperlukan.

3.4 Tipe Data¹²

Seperti bahasa pemrograman lainnya, pemrograman java memiliki tipe data yang berfungsi untuk menyimpan nilai dari sebuah data. Ada beberapa tipe data dalam java, seperti tipe data primitive, array, dan tipe data bawaan java yang merupakan padanan dari tipe data primitive yang biasa disebut dengan wrapper class. Tipe data primitive yang sering dipakai dalam pemrograman java.

¹²Sumber: Java Desktop – Ifnu Bima

Tabel 2. Tipe data 1

Tipe data	Ukuran	Min	Max	Keterangan
byte	8 bit	-128	127	Gunakan tipe data ini kalau range datanya kecil untuk menghemat memori dibanding dengan int, terutama kalau datanya berupa array
short	16 bit	-32768	32767	Sama dengan byte, gunakan kalau perlu optimisasi penggunaan memory, terutama kalau datanya berupa array
int	32 bit	-2147483648	2147483647	int adalah tipe data yang paling sering digunakan untuk representasi angka bulat. int adalah pilihan default jika tidak diperlukan optimisasi memori. Kalau jangkauan angkanya kurang lebar gunakan long.
long	64 bit	-9.22E+018	9.22E+018	Gunakan tipe data ini jika jangkauan angkanya lebih lebar daripada int
float	32 bit	-	-	float adalah tipe data pecahan yang didefinisikan oleh standard IEEE 754. Digit di belakang koma tidak bisa diatur, jika ingin merepresentasikan angka yang bisa diatur jumlah digit di depan dan di belakang koma, gunakan BigDecimal
double	64 bit	-	-	double adalah tipe data pecahan yang didefinisikan oleh standard IEEE 754. Digit di belakang koma tidak bisa diatur, jika ingin merepresentasikan angka yang bisa diatur jumlah digit di depan dan di belakang koma, gunakan BigDecimal.
boolean	-	-	-	Tipe data boolean hanya mempunyai dua nilai, true atau false.
char	16	'\u0000' atau 0	'\uffff' atau 65535	char adalah tipe data yang bisa menyimpan nilai unicode. Digunakan untuk merepresentasikan karakter unicode yang didalamnya terdapat abjad, angka, simbol dan karakter lainnya.

Berikut adalah padanan tipe data primitive dengan wrapper class.

Tabel 3. Tipe data 2

Primitif	Wrapper class
byte	Integer
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean
char	Character
	String



This page is intentionally left blank

BAB IV

PEMROGRAMAN GUI NETBEANS

4.1 Pengantar

Berbicara GUI Builder Java kita akan menemukan dua komponen penting yang dijadikan sebagai landasan pemrograman GUI, yang pertama adalah komponen Abstract Windowing Toolkit (AWT) dan javax swing. Pustaka AWT adalah pustaka yang digunakan lebih awal jika dibandingkan dengan javax swing. Artinya adalah javax swing adalah pustaka untuk memperbaiki kelemahan-kelemahan dan memperkaya pustaka komponen AWT. Maka tidak mengherankan jika ada sebagian komponen-komponen turunan javax swing memiliki kesamaan class dan penamaan komponen AWT. Sebelum kita membicarakan kedua pustaka ini, ada baiknya kita mengenal Java Foundation Class yang merupakan sekumpulan class yang digunakan sebagai fondasi dalam membangun pemrograman berbasis GUI di java.

4.2 Java Foundation Class¹³

Java Foundation Class (JFC) merupakan sekumpulan class-class Java yang digunakan untuk mengembangkan perangkat lunak berbasis GUI (Graphical User Interface). Selain itu, JFC juga mempunyai class-class yang digunakan untuk menambahkan fungsi dan kemampuan interaksi yang variatif dari pemrograman Java. Dari definisi ini, JFC tidak hanya berisi class-class GUI saja tetapi juga class-class lain yang dapat meningkatkan kemampuan pemrograman Java baik dari segi fungsionalitasnya maupun dari segi kemampuan interaksi pemrograman Java yang sangat kaya.

¹³Sumber: Java Desktop – Ifnu Bima

Tabel 4. Java Foundation Class

Fitur-fitur yang dimiliki oleh JFC	
Fitur	Deskripsi
Komponen Swing	Memuat semua class-class yang dibutuhkan untuk membuat aplikasi berbasis GUI, dari tombol, table, tab, menu, toolbar dan sebagainya
Look and Feel (LaF)	Memberikan kemampuan kepada program Java yang dikembangkan menggunakan library Swing untuk memilih tema tampilan. Misalnya sebuah program yang sama dapat mempunyai tampilan windows LaF atau Java LaF, atau LaF lain yang dikembangkan oleh komunitas seperti JGoodies.
Accessibility API	Fasilitas untuk mengembangkan aplikasi bagi penyandang cacat, misalnya dukungan untuk membuat huruf braille, kemampuan mengambil input dari layar sentuh dan sebagainya.
Java 2D API	Berisi kumpulan class-class yang dapat digunakan untuk memanipulasi object-object 2 dimensi, seperti garis, kotak, lingkaran, kurva dan lain sebagainya. Selain itu Java 2D API juga memberikan kemampuan program yang ditulis menggunakan Java untuk mencetak output ke alat pencetak seperti printer.
Drag-and-drop	Menyediakan kemampuan drag-and-drop antara program Java dan program lain yang ditulis spesifik untuk suatu platform sistem operasi tertentu.
Internationalization (i18n)	Membantu pengembang perangkat lunak untuk membangun aplikasi yang dapat mendukung semua bahasa dan huruf yang ada di dunia.

4.3 AWT

Abstract Windowing Toolkit (AWT) adalah pustaka windowing yang ditujukan untuk menyediakan kelas untuk membuat pemrograman GUI di java. AWT memiliki kemampuan untuk membuat window, menggambar, dan komponen-komponen lain yang mendukung pemrograman GUI seperti: button, scrollbar, checkbox, textfield, label, dll. AWT merupakan versi awal dari pemrograman GUI di java, seperti dijelaskan sebelumnya bahwa paket javax swing ditujukan untuk melengkapi pustaka serta menyempurnakan kekurangan-kekurangan yang ada pada AWT.

IDE netbeans sudah menyediakan bentuk visual AWT yang dapat kita lihat dan gunakan secara langsung dalam ruang kerja (frame) kita. Untuk lebih mengenal komponen visual java AWT, ada baiknya kita buat secara manual penggunaan beberapa komponen AWT-nya. Sehingga kita dapat memahami struktur dan fungsi penggunaan java AWT.

a. Praktik Label

Ketiklah dengan menggunakan template Applet untuk mengeluarkan tampilan Label komponen AWT berikut ini.

```
public void init() {  
    // TODO start asynchronous download of heavy resources  
    Label pertama = new Label("Label Pertama");  
    Label kedua = new Label("Label Kedua");  
    Label ketiga = new Label("Label Ketiga");  
    Label keempat = new Label("Label Keempat");  
    Label kelima = new Label("Label Kelima");  
  
    add(pertama);  
    add(kedua);  
    add(ketiga);  
    add(keempat);  
    add(kelima);  
}
```

Gambar 15. AWT 1

b. Praktik Tombol

Ketiklah dengan menggunakan template Applet untuk mengeluarkan tampilan Button komponen AWT berikut ini.

```
public class DemoTombol extends Applet implements ActionListener {  
    String psn = "";  
    Button ya, tdk, mungkin;  
  
    public void init() {  
        ya = new Button("Ya");  
        tdk = new Button("Tidak");  
        mungkin = new Button("Mungkin");  
  
        add(ya);  
        add(tdk);  
        add(mungkin);  
  
        ya.addActionListener(this);  
        tdk.addActionListener(this);  
        mungkin.addActionListener(this);  
    }  
}
```

```
public void actionPerformed(ActionEvent ae) {
    String str = ae.getActionCommand();

    if(str.equals("Ya")) {
        psn = "Anda menekan Ya.";
    }
    else if(str.equals("Tidak")) {
        psn = "Anda menekan Tidak.";
    }
    else {
        psn = "Anda menekan Mungkin.";
    }
    repaint();
}

public void paint(Graphics g) {
    g.drawString(psn, 6, 100);
}
}
```

Gambar 16. AWT 2

c. Praktik Teks

Ketiklah dengan menggunakan template Applet untuk mengeluarkan tampilan Teks komponen AWT berikut ini.

```
public class DemoBidangTeks extends Applet
    implements ActionListener {
    TextField nama, pass;

    public void init() {
        Label namap = new Label("Nama: ", Label.RIGHT);
        Label passp = new Label("Password: ", Label.RIGHT);

        nama = new TextField(12);
        pass = new TextField(8);
        pass.setEchoChar('?');

        add(namap);
        add(nama);
        add(passp);
        add(pass);

        // meregister untuk menerima event
        nama.addActionListener(this);
        pass.addActionListener(this);
    }
}
```

```
// Event jika pengguna menekan Enter.
public void actionPerformed(ActionEvent ae) {
    repaint();
}

public void paint(Graphics g) {
    g.drawString("Nama: " + nama.getText(), 6, 60);
    g.drawString("Teks nama: "
        + nama.getSelectedText(), 6, 80);
    g.drawString("Password: " + pass.getText(), 6, 100);
}
}
```

Gambar 17. AWT 3

d. Praktik Combo (Choice)

Ketiklah dengan menggunakan template Applet untuk mengeluarkan tampilan Combo (choice) komponen AWT berikut ini.

```
public class DemoChoice extends Applet implements ItemListener {
    Choice os, browser;
    String psn = "";

    public void init() {
        os = new Choice();
        browser = new Choice();

        // menambahkan item-item pada daftar os
        os.add("Windows XP");
        os.add("Windows Vista");
        os.add("Solaris");
        os.add("Mac OS");

        // menambahkan item-item pada daftar browser
        browser.add("Internet Explorer");
        browser.add("Firefox");
        browser.add("Opera");

        // menambahkan daftar pilihan pada jendela
        add(os);
        add(browser);
    }
}
```



```

        // meregister untuk menerima event
        os.addItemListener(this);
        browser.addItemListener(this);
    }

    public void itemStateChanged(ItemEvent ie) {
        repaint();
    }

    // Menampilkan pilihan terkini.
    public void paint(Graphics g) {
        psn = "OS terkini: ";
        psn += os.getSelectedItemAt();
        g.drawString(psn, 6, 120);
        psn = "Browser terkini: ";
        psn += browser.getSelectedItemAt();
        g.drawString(psn, 6, 140);
    }
}
]

```

Gambar 18. AWT 4

e. Praktik Checklist

Ketikhlah dengan menggunakan template Applet untuk mengeluarkan tampilan Checklist komponen AWT berikut ini.

```

public class DemoKotakPeriksa extends Applet implements ItemListener {
    String psn = "";
    Checkbox winXP, winVista, solaris, mac;

    public void init() {
        winXP = new Checkbox("Windows XP", null, true);
        winVista = new Checkbox("Windows Vista");
        solaris = new Checkbox("Solaris");
        mac = new Checkbox("Mac OS");

        add(winXP);
        add(winVista);
        add(solaris);
        add(mac);

        winXP.addItemListener(this);
        winVista.addItemListener(this);
        solaris.addItemListener(this);
        mac.addItemListener(this);
    }

    public void itemStateChanged(ItemEvent ie) {
        repaint();
    }
}

```

```
// Menampilkan keadaan terkini dari tiap kotak periksa.
public void paint(Graphics g) {
    psn = "Keadaan terkini: ";
    g.drawString(psn, 6, 80);

    psn = " Windows XP: " + winXP.getState();
    g.drawString(psn, 6, 100);

    psn = " Windows Vista: " + winVista.getState();
    g.drawString(psn, 6, 120);

    psn = " Solaris: " + solaris.getState();
    g.drawString(psn, 6, 140);

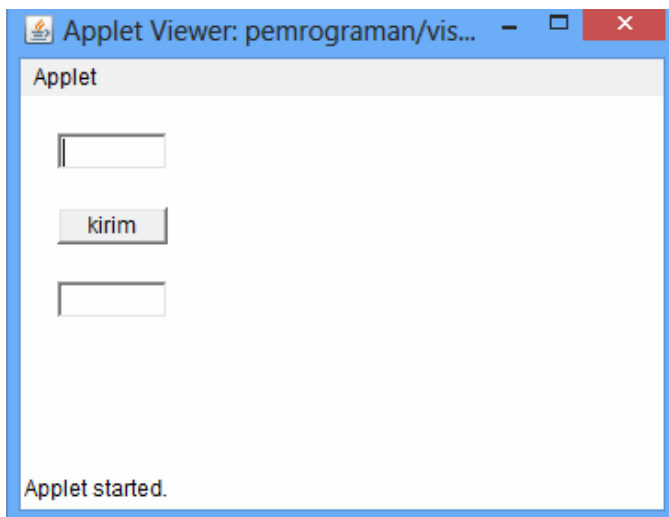
    psn = " Mac OS: " + mac.getState();
    g.drawString(psn, 6, 160);
}
}
```

Gambar 19. AWT 5

4.4 Latihan 3.

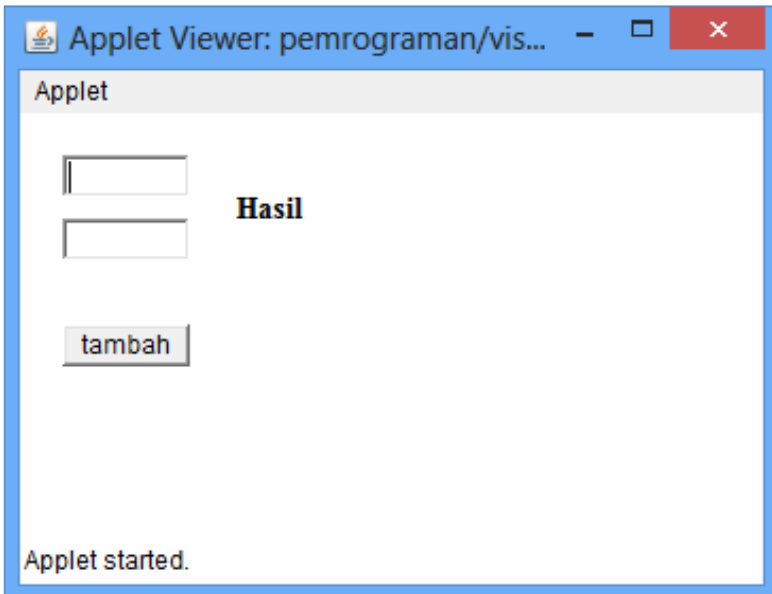
Kerjakanlah latihan-latihan di bawah ini.

1. Buatlah dengan menggunakan “kode” tampilan applet di bawah ini, selanjutnya tambahkan aksi ke button “kirim” untuk mengirimkan nilai string dari textfield pertama ke textfield ke dua.



Gambar 20. Latihan AWT 1

2. Buatlah dengan menggunakan “kode” tampilan applet di bawah ini, selanjutnya tambahkan aksi ke button “tambah” untuk menambahkan nilai dari textfield pertama dan textfield ke dua, keluarkan hasilnya di label hasil.



Gambar 21. Latihan AWT 2

4.5 Javax Swing

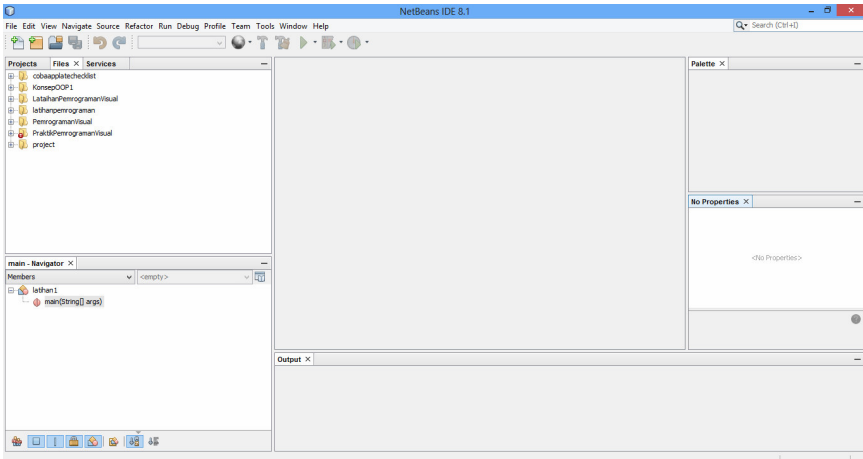
Untuk melihat dokumentasi lengkap javax swing, silahkan kunjungi <https://docs.oracle.com/javase/7/docs/api>, banyak sekali dokumentasi javax swing yang dapat kita gunakan. Dalam modul ini, tidak semua disajikan hanya beberapa dokumentasi paket-paket penting untuk mendukung perkuliahan pemrograman visual. Untuk membantu pemahaman komponen swing, kita akan mencoba mengeksplorasinya menggunakan IDE netbeans.

a. Menjalankan Netbeans

Pada saat pertama kali menjalankan netbeans, kita akan disuguhkan tampilan splash screen.

b. Area kerja Netbeans

Setelah tampilan splash screen dimunculkan, maka akan tampak tampilan ruang kerja (area kerja) netbeans seperti terlihat pada tampilan di bawah ini.



Gambar 22. Tampilan area kerja Netbeans

Ada beberapa pembagian area kerja utama netbeans, yakni:

- 1) Posisi kiri: Project, Files, Services, dan navigator
- 2) Posisi tengah: Tempat perancangan atau design form.

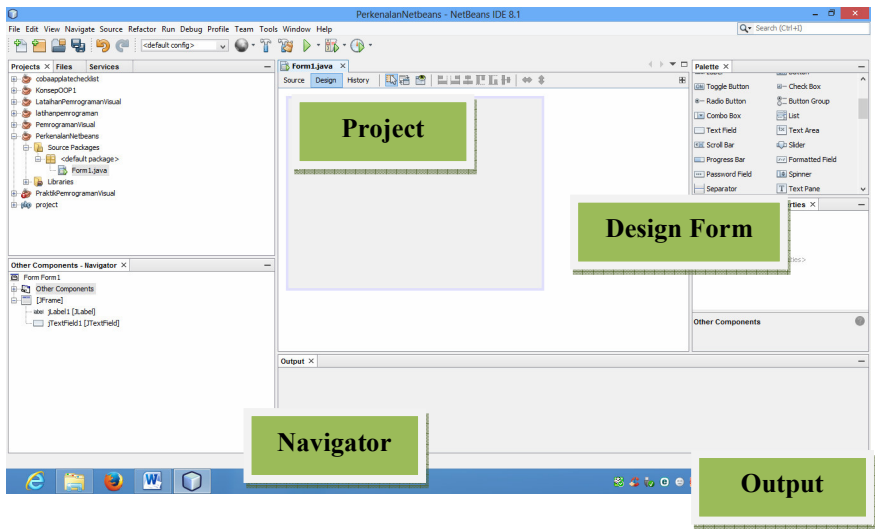
Design form memiliki dua menu, yakni source dan design. Source adalah tempat mengetik kode program, sementara design adalah tempat untuk mendesain tampilan GUI.

- 3) Posisi kanan: Palette dan Properties.

Palette adalah sekumpulan fasilitas GUI yang dapat kita gunakan dengan cara klik dan drag ke form sesuai rancangan tampilan GUI kita. Sementara properties merupakan nilai dari setiap komponen rancangan GUI kita.

- 4) Tampilan lengkap

Secara lengkap tampilan ruang kerja Netbeans dapat kita lihat di bawah ini.

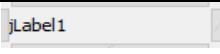
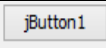




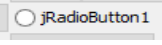



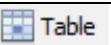
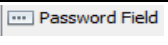
Gambar 23. Tampilan lengkap netbeans



c. Pengenalan komponen palette

Palette memiliki komponen-komponen yang dapat membantu kita dalam mendesign tampilan pemrograman GUI. Tetapi dalam modul ini hanya memfokuskan pada komponen-komponen penting yang akan sering digunakan oleh praktikan, diantaranya:




Tabel 5. Menu palette 1

Ikon	Nama	Keterangan
	JLabel	Digunakan untuk mengeluarkan tulisan atau keterangan.
	JButton	Digunakan sebagai tombol, atau navigasi yang akan diberi aksi tertentu. Ketika menggunakan aksi JButton membutuhkan actionPerformed yang berada di bawah class ActionListener.
	JTextField	Digunakan sebagai komponen input berupa teks. Sama seperti JButton, JTextField

		keteika diberikan aksi, harus merepresentasikan class <code>actionListener</code> .
	JComboBox	Digunakan untuk memberikan isian pilihan, ketika diberikan aksi harus merepresentasikan <code>itemListener</code> .
	JRadioButton	Digunakan untuk pilihan dan bertipe boolean (biasanya hanya satu pilihan), misalnya jenis kelamin. Sama dengan JComboBox harus merepresentasikan <code>itemListener</code>
	ButtonGroup	Digunakan untuk menyatukan beberapa JRadioButton dan bertipe boolean.
	JTextArea	Digunakan untuk input teks yang lebih dari satu baris (space yang lebih besar dari <code>JTextField</code>).
	JList	Digunakan untuk mengeluarkan teks dalam bentuk list.
	JTable	Digunakan untuk menampilkan data ke dalam bentuk tabel (kolom dan baris) atau berbentuk grid.
	JPasswordField	Digunakan untuk input teks, yang tulisan teks-nya tidak terlihat (cocok untuk input kata sandi). Sama merepresentasikan <code>actionListener</code> , tetapi JPasswordField bertipe array of character (<code>char[]</code>) bukan String. Jadi ketika kita mendefenisikan variabelnya: <code>char[] passwordChar = JPasswordField.getPassword();</code> <code>String password = new</code>

		String(passwordChar); //ubah tipe
 Password Field	JFormattedText	Digunakan untuk menampilkan teks atau tulisan dalam format khusus (number, tanggal, jam, uang, dll).
 Check Box	JCheckBox	Digunakan untuk input terseleksi dan bertipe boolean.

Tabel 6. Menu palette 2

Ikon	Nama	Keterangan
 File Menu Bar	JMenuBar	Digunakan untuk membuat menu utama pada design program.
 Menu	JMenu	Merupakan anak JMenuBar, dan biasanya memiliki beberapa menu item (sub menu).
 Menu Item	JMenuItem	Merupakan item menu dari Jmenu.

4.6 Penjelasan singkat pengenalan komponen swing.

Netbeans memberikan kemudahan bagi pengguna dengan tampilan dan fasilitas GUI Builder yang sangat baik. Sebelum kita memanfaatkan fasilitas netbeans tersebut dengan cara “klik dan drag”, alangkah baiknya kita mengeluarkan beberapa komponen swing secara manual untuk mengetahui struktur-nya.

a. Form

Form dalam pemrograman java digunakan sebagai media untuk menampung komponen-komponen swing lainnya ketika kita merancang aplikasi desktop J2SE, sederhananya form adalah tampilan dari sebuah aplikasi. Contoh form dapat kita lihat pada tampilan utama dari IDE Netbeans yang kita gunakan untuk praktik pemrograman visual ini, sementara form yang ada dalam

pemrograman java disebut dengan JFrame yang direpresentasikan oleh kelas JFrame.

1) Membuat Form secara manual

Langkah-langkahnya sebagai berikut: klik kanan pada paket project kita → selanjutnya pilih new → kemudian pilih other → setelah terbuka jendela baru → pilih java → kemudian pilih java main class. Selanjutnya tulis kode java-nya seperti di bawah ini:

```
package javaapplication4;

import javax.swing.JFrame;

/**
 *
 * @author Dell
 */
public class Form1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        JFrame form = new JFrame();
        form.setSize(500, 500);
        form.setLocation(50, 50);
        form.setVisible(true);
    }
}
```

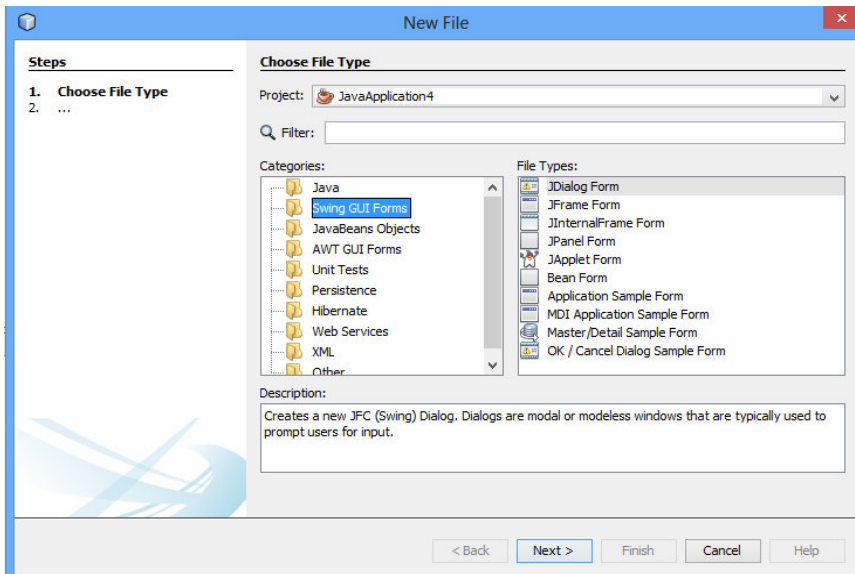
Gambar 24. Form manual

Pada contoh di atas, kita membuat kelas “Form1” sebagai kelas dari JFrame kita. Agar kelas “Form1” dapat digunakan kita harus membuat objek untuk kelas tersebut yang kita beri nama dengan form (JFrame form = new JFrame();). Sebenarnya setelah objek tersebut kita buat, kita telah berhasil membuat sebuah form, tetapi form tersebut belum dapat digunakan karena seluruh ukuran dari form tersebut masih dibuat secara default seluruhnya, sehingga kita perlu menentukan ukuran¹⁴(setSize), lokasi(setLocation), serta menampilkan objek form (setVisible) yang telah kita buat tersebut.

¹⁴ Secara default size/ukuran di java menggunakan type pixel (px), selain itu secara default ukuran lebar dan tinggi adalah 0*0.

2) Membuat Form dengan GUI Builder Netbeans

Untuk menggunakan JFrame kita dapat memanfaatkan fasilitas “GUI Builder” yang telah diberikan netbeans dengan langkah-langkahnya sebagai berikut:Klik kanan pada paket project kita →selanjutnya pilih new →dan pilih other. Setelah itu akan keluar tampilan jendela baru →kemudian silahkan pilih swing GUI forms →dan pilih JFrame Form.



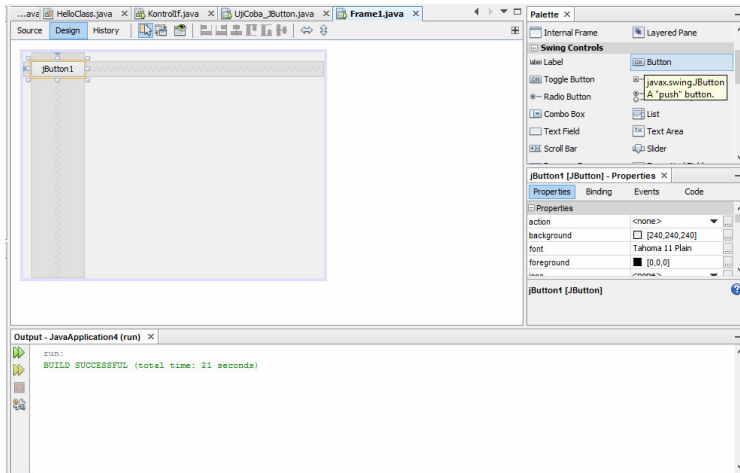
Gambar 25. Form dengan netbeans

Setelah itu, silahkan buat nama JFrame kita sesuai dengan tema form yang ingin kita buat.

b. Tombol

3) JButton

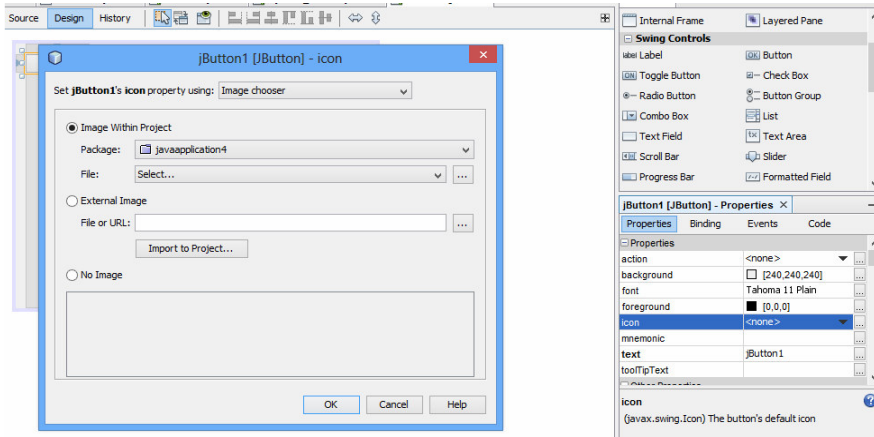
JButton merupakan default tombol di java swing. Untuk menambahkan pada bagian palette klik dan drag (tarik) ke form yang telah kita buat. Untuk merubah text/tulisan JButton pada tombol silahkan klik JButton kemudian masuk ke bagian propertis, selanjutnya cari propertis text lalu ubah tulisan JButton dengan tulisan lain sesuai dengan keinginan kita.



Gambar 26. Button

Secara default, netbeans akan memberikan nama pada jButton yang kita tarik dengan nama "jButton1". Jika ingin merubah nama "jButton1" ini dengan nama yang kita inginkan, silahkan klik kanan dan klik pada "change variabel name"¹⁵.

Untuk menambahkan icon pada jButton yang kita buat langkahnya adalah, klik jButton kemudian cari teks icon pada properti.



Gambar 27. Button icon

¹⁵ Ingat ketika anda merubah nama dari jButton, anda merubah struktur default dari jButton tersebut sehingga sekarang nama tombol jButton bukan lagi "jButton1" tetapi telah berubah dengan nama tombol yang baru.

Catatan: masih banyak fungsi-fungsi lain yang ada di propertis seperti: merubah tulisan (font), menebalkan huruf, mengatur posisi tulisan pada tombol, dll. silahkan anda eksplorasi sendiri.

c. Memberikan aksi pada tombol

Tombol sama dengan memberikan aksi¹⁶ !. Ketika kita menggunakan tombol maka sebenarnya kita harus memberikan aksi pada tombol, ketika tombol tersebut kita klik. Pemberian aksi tersebut pada java swing disebut dengan action performed “actionPerformed”. Action performed berada di bawah class (kelas) action listener (actionListener). Jadi kesimpulannya, pemberian aksi berada di bawah method action performed, method action performed dapat digunakan setelah direpresentasikan terlebih dahulu oleh kelas action listener. Untuk contohnya dapat dilihat di bawah ini:

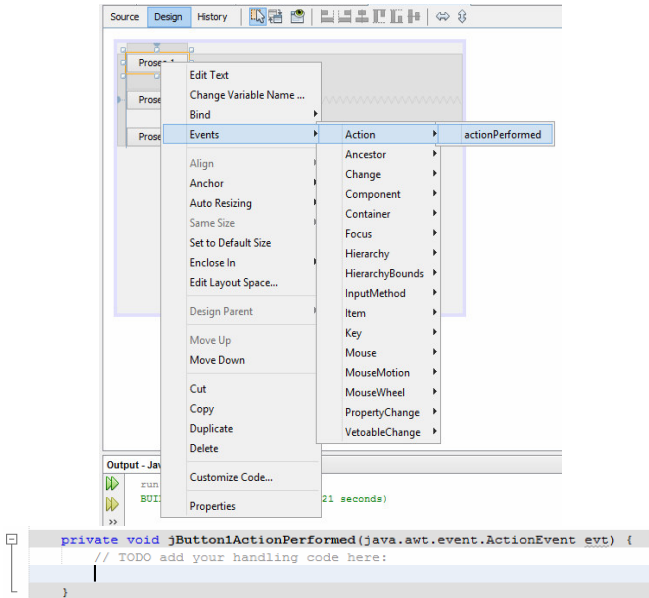
```
public class Frame1 extends javax.swing.JFrame {  
  
    /**  
     * Creates new form Frame1  
     */  
  
    public Frame1() {  
        initComponents();  
        jButton1.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent ae) {  
                System.out.println("Tombol Proses 1 Di Klik");  
            }  
        });  
    }  
}
```

Gambar 28. Memberi aksi button manual

Pada contoh di atas, kita menambahkan action listener di tombol “jButton1”. Ketika kita menambahkan action listener, maka action listener akan membuat method action performed sebagai tempat kita untuk merepresentasikan/membuat aksi-aksi pada tombol “jButton1”. Kode java di atas berlaku jika kita melakukan penulisan kode java secara manual, sebenarnya kita bisa meminta netbeans untuk menambahkan method action performed secara otomatis dengan cara klik kanan pada tombol yang akan kita berikan

¹⁶ Aksi disini dapat diartikan dengan menulis kode java (kode) untuk melakukan perintah tertentu.

aksi, kemudian pilih event, pilih action selanjutnya pilih *action performed*.

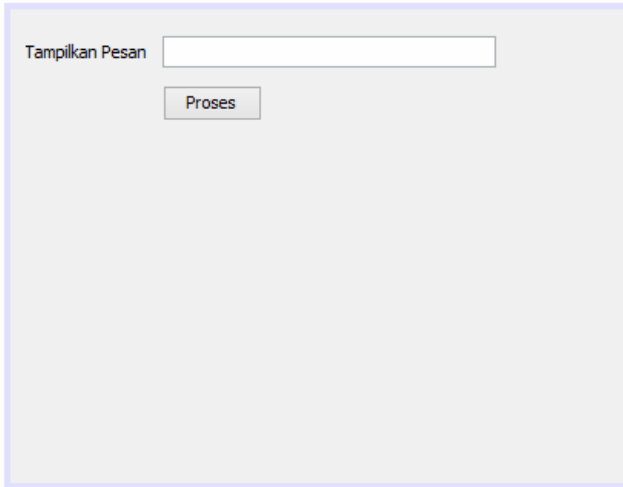


Gambar 29. Memberi aksi button netbeans

Setelah melihat dua contoh struktur pengeluaran form dan tombol secara manual, diharapkan praktikan sudah memahami struktur pembentukan dua komponen tersebut dan pengeluaran struktur komponen pada netbeans dengan fasilitas “klik dan drag” lebih kurang sama dengan pengeluaran struktur yang telah kita praktikkan di atas. untuk lebih memahami mengeluarkan komponen swing secara manual, silahkan praktikkan latihan di bawah ini.

4.6 Latihan GUI.

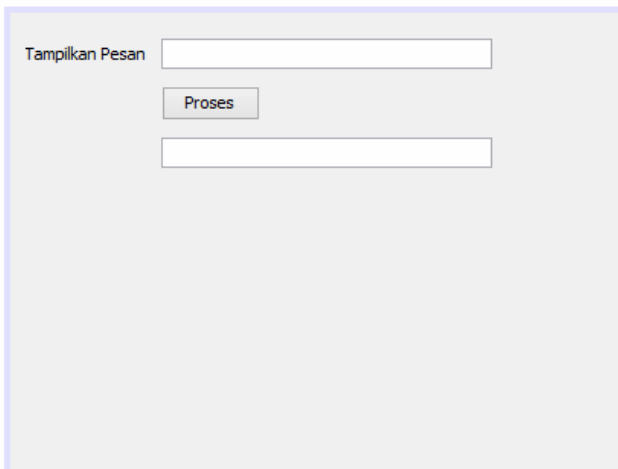
- 1) Buatlah project baru dengan nama tampilan_pesanan.
- 2) Isikanlah ke dalam JFrame satu JLabel, satu JTextField, dan satu JButton. Ubahlah nama (variabel name) komponen JLabel dengan labelPesanan, JTextField dengan textPesanan, dan JButton dengan btnPesanan.



- 3) Kemudian berikan aksi pada tombol JButton dengan kode di bawah ini:

```
private void btnPesanActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    textPesan.setText("selamat dan semangat belajar java");  
}
```

- 1) Buatlah file baru dengan nama kirm_pesani.
- 2) Isikanlah ke dalam JFrame satu JLabel, dua JTextField, dan satu JButton. Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.



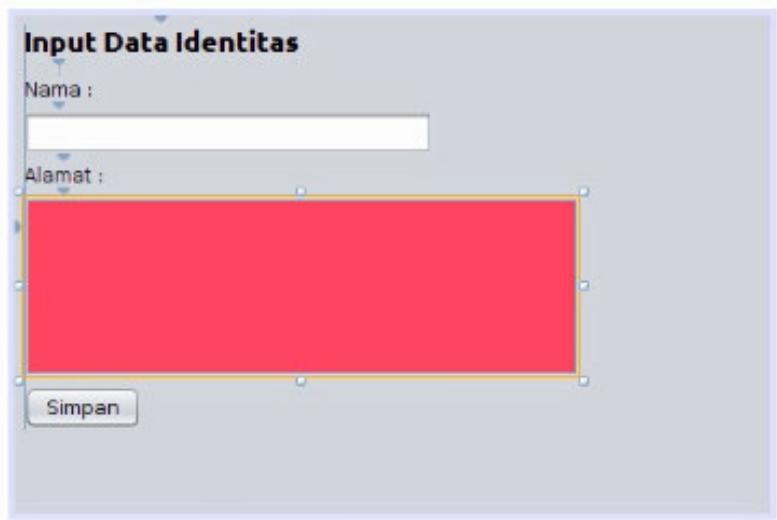
- 3) Kemudian berikan aksi pada tombol JButton dengan kode di bawah ini:

```
private void btnPesanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String kirim = textPesan.getText();
    jTextField1.setText(kirim);
}
```

- 4) Kemudian berikan aksi pada txtPesan dengan kode di bawah ini:

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

- 1) Buatlah file baru dengan nama belajar_Textarea.
- 2) Isikanlah ke dalam JFrame dua JLabel, satu JTextField, satu JButton, dan satu JTextArea (ubah latar jadi warna merah pada properties). Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.



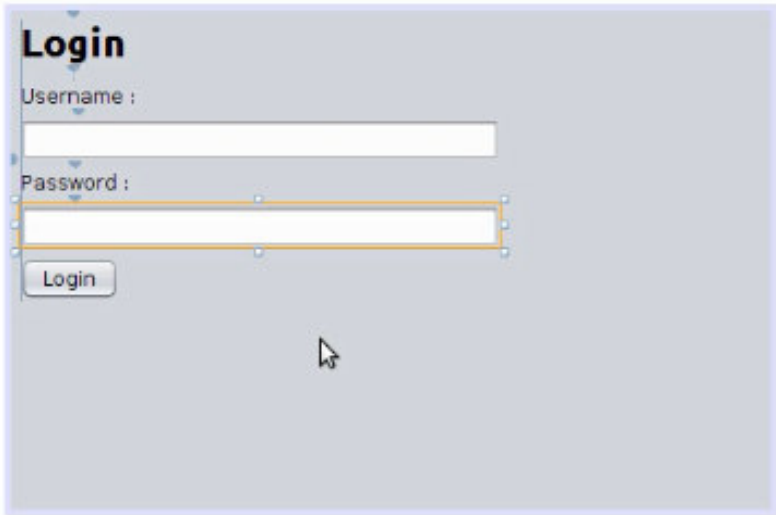
- 3) Jalankan
- 4) Kemudian ubah line wrap JTextArea menjadi "true" dan silahkan ceklis wrap style word pada properties.
- 5) Kemudian isikan kode di bawah ini pada JButton:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
    // TODO add your handling code here:
    String nama = jTextFieldNama.getText();
    String alamat = JTextAreaAlamat.getText();

    System.out.println("Nama : "+nama);
    System.out.println("Alamat : "+alamat);
}
```

- 1) Buatlah file baru dengan nama belajar_password.

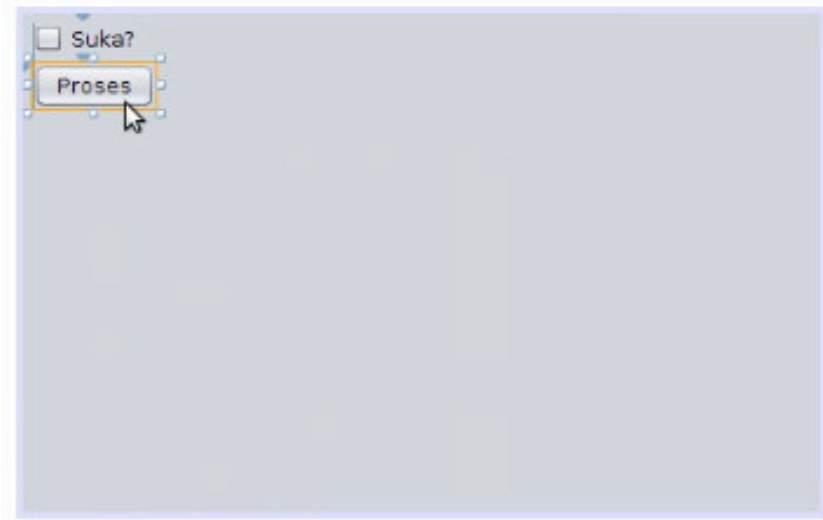
- 2) Isikanlah ke dalam JFrame dua JLabel, satu JTextField, satu JButton, dan satu JPasswordField. Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.



- 3) Jalankan
- 6) Kemudian isikan kode di bawah ini pada JButton:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent  
    // TODO add your handling code here:  
  
    String username = jTextFieldUsername.getText();  
    char[] passwordChars = jPasswordField1.getPassword();  
    String password = new String(passwordChars);  
  
    System.out.println("Username : "+username);  
    System.out.println("Password : "+password);  
}
```

- 1) Buatlah file baru dengan nama belajar_checkbox.
- 2) Isikanlah ke dalam JFrame satu JcheckBox dan satu JButton. Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.



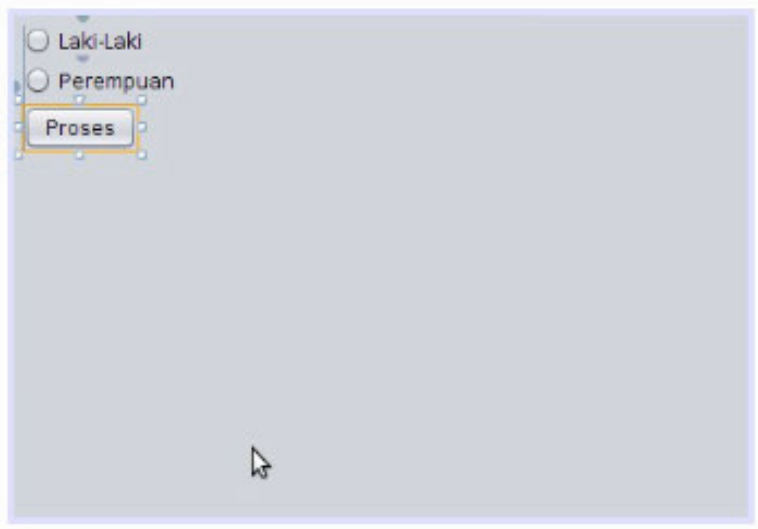
3) Jalankan

4) Kemudian isikan kode di bawah ini pada JButton:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent  
    // TODO add your handling code here:  
    boolean hasil = jCheckBox1.isSelected();  
    System.out.println("Hasil = "+hasil);  
}
```

1) Buatlah file baru dengan nama belajar_radiobutton.

2) Isikanlah ke dalam JFrame dua JRadioButton dan satu JButton. Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.

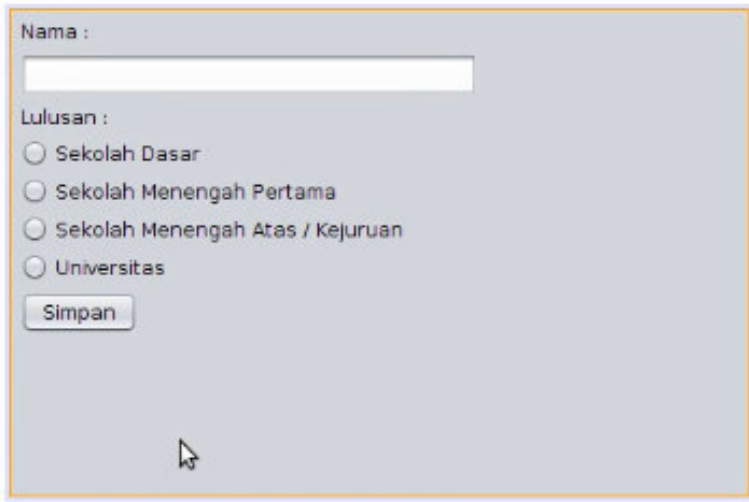


- 3) Jalankan
- 4) Kemudian isikan kode di bawah ini pada JButton:

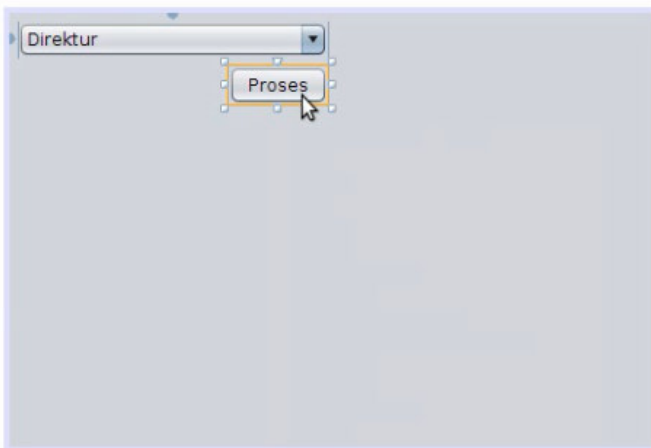
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent)
{
    // TODO add your handling code here:
    boolean lakilaki = jButton1.isSelected();
    boolean perempuan = jButton2.isSelected();

    System.out.println("Laki-Laki : "+lakilaki);
    System.out.println("Perempuan : "+perempuan);
}
```

- 1) Buatlah file baru dengan nama belajar_buttongroup.
- 2) Isikanlah ke dalam JFrame satu JTextField, lima JRadioButton, dan satu JButton. Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.



- 3) Jalankan.
 - 4) Tambahkan JbuttonGroup pada JFrame, selanjutnya lihat pada navigator telah ditambahkan satu Jbuttongroup. Klik setiap JradioButton, kemudian ambil properties dan cari buttonGroup selanjutnya pilih JbuttonGroup yang telah kita tambahkan tadi.
 - 5) Jalankan.
- 1) Buatlah file baru dengan nama belajar_combobox.
 - 2) Isikanlah ke dalam JFrame satu JComboBox dan satu JButton. Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.

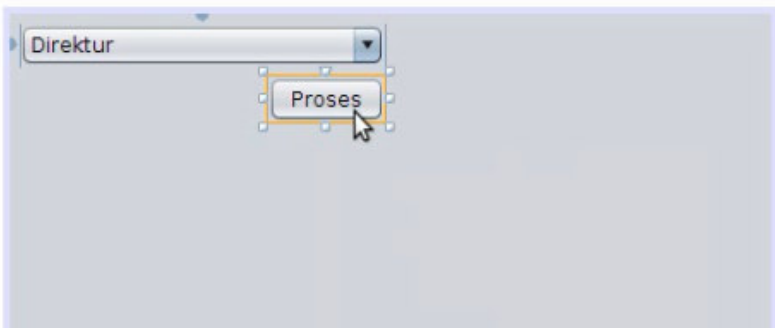


- 3) Jalankan.

- 4) Ambil properties → model → ubah nilai model (dosen, Kabag TU, Kabag Umum, dan rektor).
- 5) Kemudian isikan kode di bawah ini pada JButton:

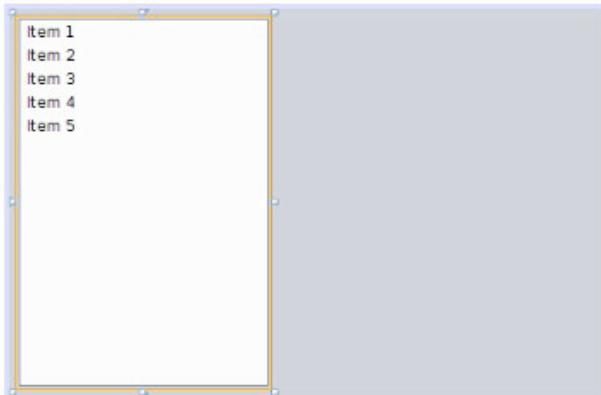
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent)
// TODO add your handling code here:
String jabatan = (String) JComboBox1.getSelectedItemAt();
System.out.println(jabatan);
}
```

- 1) Buatlah file baru dengan nama belajar_combobox.
- 2) Isikanlah ke dalam JFrame satu JComboBox dan satu JButton. Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.



- 3) Jalankan.
- 4) Ambil properties → model → ubah nilai model (dosen, Kabag TU, Kabag Umum, dan rektor).
- 5) Kemudian isikan kode di bawah ini pada JButton:

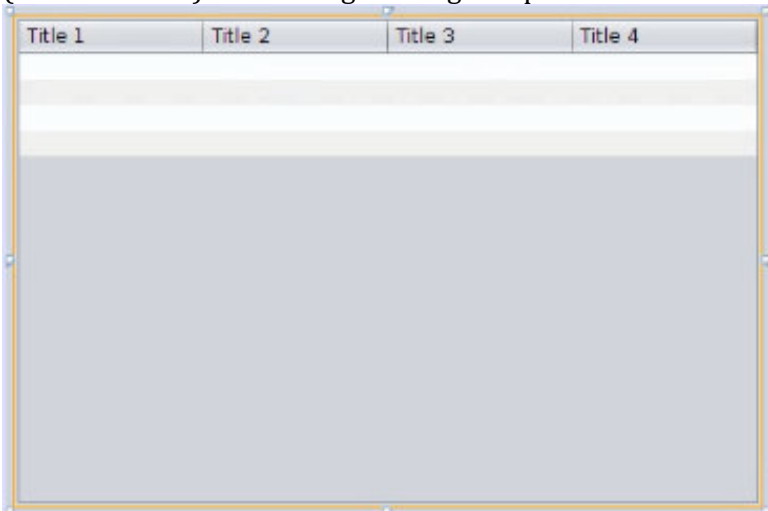
- 1) Buatlah file baru dengan nama belajar_listbox.
- 2) Isikanlah ke dalam JFrame satu JlistBox. Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.



- 3) Jalankan.
- 4) Ambil properties → model → ubah nilai model (dosen, Kabag TU, Kabag Umum, dan rektor).
- 5) Kemudian isikan kode di bawah ini pada JButton:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
// TODO add your handling code here:
List list = jList1.getSelectedValuesList();
for(Object objek : list){
    System.out.println(objek);
}
}
```

- 1) Buatlah file baru dengan nama belajar_tabel.
- 2) Isikanlah ke dalam JFrame satu JTable. Ubahlah nama (variabel name) sesuai dengan keinginan praktikan.



- 3) Jalankan.
- 4) Ambil properties → model → ubah nilai model (nim, nama, nama wali, no. telp).
- 5) Jalankan.



This page is intentionally left blank

BAB V

KONEKSI DATABASE

5.1 Mengetahui JDBC¹⁷

Java Database Connectivity adalah komponen API yang disediakan java untuk mengakses database. ada tiga hal yang dikelola oleh JDBC yakni, koneksi ke sumber data, mengirimkan query dan statement ke database, menerima dan mengolah resultset yang diperoleh dari database.

JDBC mempunyai empat komponen :

a. JDBC API

JDBC API menyediakan metode akses yang sederhana ke sumber data relational (RDBMS) menggunakan pemrograman Java. dengan menggunakan JDBC API, kita bisa membuat program yang dapat mengeksekusi SQL, menerima hasil ResultSet, dan mengubah data dalam database. JDBC API juga mempunyai kemampuan untuk berinteraksi dengan lingkungan terdistribusi dari jenis sumber data yang berbeda-beda. JDBC API adalah bagian dari Java Platform yang disertakan dalam library JDK maupun JRE. JDBC API sekarang ini sudah mencapai versi 4.0 yang disertakan dalam JDK 6.0. JDBC API 4.0 dibagi dalam dua package yaitu : `java.sql` dan `javax.sql`.

b. JDBC Driver Manager

Class DriverManager dari JDBC bertugas untuk mendefinisikan object-object yang dapat digunakan untuk melakukan koneksi ke sebuah sumber data. Secara tradisional DriverManager telah menjadi tulang punggung arsitektur JDBC.

c. JDBC Test Suite

JDBC Test Suite membantu kita untuk mencari driver mana yang cocok digunakan untuk melakukan sebuah koneksi ke sumber data tertentu. Tes yang dilakukan tidak memerlukan resource besar ataupun tes yang komprehensif, namun cukup tes-tes

¹⁷Sumber: Java Desktop – Ifnu Bima

sederhana yang memastikan fitur-fitur penting JDBC dapat berjalan dengan lancar.

d. JDBC-ODBC Bridge

Bridge ini menyediakan fasilitas JDBC untuk melakukan koneksi ke sumber data menggunakan ODBC (Open DataBase Connectivity) driver. Sebagai catatan, anda perlu memload driver ODBC di setiap komputer client untuk dapat menggunakan bridge ini. Sebagai konsekuensinya, cara ini hanya cocok dilakukan di lingkungan intranet dimana isu instalasi tidak menjadi masalah.

Sebagai catatan, tidak semua komponen JDBC tersebut dipakai ketika kita menghubungkan java dengan aplikasi database. Dalam modul ini, kita akan mencoba menggunakan JDBC API dan JDBC Driver Manager.

5.2 Database Driver

Untuk terhubung ke database JDBC memerlukan database driver, dalam modul ini kita akan mengkoneksikan pemrograman java dengan database mysql menggunakan mysql connector j (jconnector) yang dapat di download di situs resmi sql. Perlu diingat setiap aplikasi database memiliki database driver tersendiri oleh karena itu, kita perlu menyesuaikan database driver dengan aplikasi database yang akan kita gunakan.

5.3 Membuat Koneksi

Melakukan koneksi ke database melibatkan dua langkah: Memload driver dan membuat koneksi itu sendiri. Cara memload driver sangat mudah, pertama letakkan file jar database driver ke dalam classpath. Kemudian load driver dengan menambahkan kode berikut ini:

```
Class.forName("com.mysql.jdbc.Driver");
```

Nama class database driver untuk setiap DBMS berbeda, anda bisa menemukan nama class tersebut dalam dokumentasi driver database yang anda gunakan. Dalam contoh ini, nama class database driver dari MySQL adalah `com.mysql.jdbc.Driver`. Memanggil method `Class.forName` secara otomatis membuat instance dari database driver, class `DriverManager` secara otomatis juga dipanggil untuk mengelola class database driver ini. Jadi anda tidak perlu menggunakan statement `new` untuk membuat instance dari class

database driver tersebut. Langkah berikutnya adalah membuat koneksi ke database menggunakan database driver yang sudah di-load tadi. Class DriverManager bekerja sama dengan interface Driver untuk mengelola driver-driver yang di-load oleh aplikasi, jadi dalam satu sesi anda bisa me-load beberapa database driver yang berbeda. Ketika kita benar-benar melakukan koneksi, JDBC Test Suite akan melakukan serangkaian tes untuk menentukan driver mana yang akan digunakan. Parameter yang digunakan untuk menentukan driver yang sesuai adalah URL. Aplikasi yang akan melakukan koneksi ke database menyediakan URL pengenalan dari server database tersebut. Sebagai contoh adalah URL yang digunakan untuk melakukan koneksi ke MySQL :

```
jdbc:mysql://[host]:[port]/[schema]
```

contoh konkritnya :

```
jdbc:mysql://localhost:3306/latihan
```

Setiap vendor DBMS akan menyertakan cara untuk menentukan URL ini di dalam dokumentasi. Anda tinggal membaca dokumentasi tersebut tanpa harus khawatir tidak menemukan informasi yang anda perlukan.

Method DriverManager.getConnection bertugas untuk membuat koneksi:

```
Connection conn =  
    DriverManager.getConnection(  
        "jdbc:mysql://localhost:3306/latihan");
```

Dalam kebanyakan kasus anda juga harus memasukkan parameter username dan password untuk dapat melakukan koneksi ke dalam database. Method getConnection menerima Username sebagai parameter kedua dan password sebagai parameter ketiga, sehingga kode di atas dapat dirubah menjadi :

```
Connection conn =  
    DriverManager.getConnection(  
        "jdbc:mysql://localhost:3306/latihan",  
        "root", "");
```

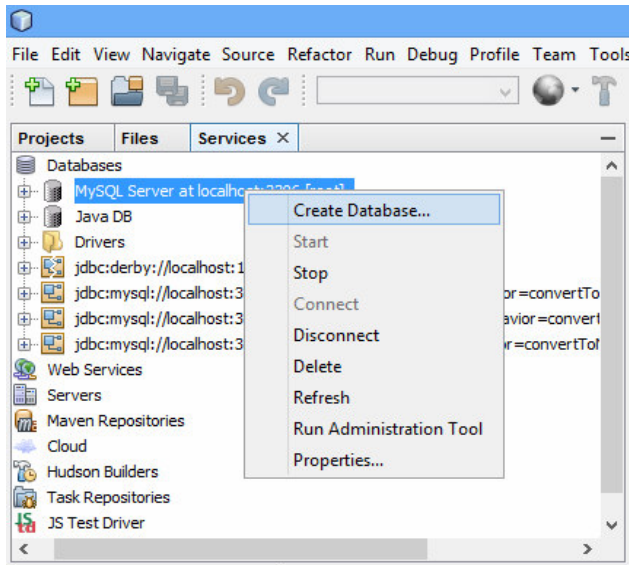

Jika salah satu dari driver yang diload berhasil digunakan untuk melakukan koneksi dengan URL tersebut, maka koneksi ke database berhasil dilaksanakan. Class Connection akan memegang informasi koneksi ke database yang didefinisikan oleh URL tersebut. Setelah sukses melakukan koneksi ke database, kita dapat mengambil data dari database menggunakan perintah query ataupun melakukan perubahan terhadap database.

5.4 Menambahkan pustaka MySql JDBC Driver

Seperti yang pembahasan di atas, kita akan mengkoneksikan pemrograman java yang kita buat dengan database MySql. Untuk itu kita harus mengeluarkan MySql JDBC Driver dengan langkah, klik kanan pada libraries di project → add library → MySql JDBC Driver → add library.

Untuk membuat database dan tabel kita dapat lakukan di netbeans dengan langkah-langkah sebagai berikut.

1. Ambil menu services → klik kanan pada MySQL Server at localhost:3306[root] → create database → nama database.



Gambar 30. Membuat database

2. Lihat pada bagian bawah Drivers, database yang kita buat sebelumnya sudah tersedia. Untuk membuat tabel database yang telah kita buat, langkahnya dengan memilih *create table*.

DAFTAR PUSTAKA

- [1] Ifnu Bima, Modul Java Desktop
- [2] Dr. Eng R.H. Sianipar, S.T, M.T, M.Eng. Soal, kasus, dan penyelesaian pemrograman java
- [3] Rosihan Ari Yuana, S.Si, M.Kom, E-book Pemrograman Java
- [4] <https://id.wikipedia.org/wiki/Java>

BIODATA PENULIS



Nama : Khairan AR
Jenis Kelamin : Laki-Laki
Tempat/Tanggal Lahir : Ambon/04 Juli 1986
NIP/NIDN : 198607042014031001 / 2004078602
Pangkat/Golongan : Penata Muda Tk.I/IIIb
Jabatan Fungsional : Asisten Ahli
Fakultas/Jurusan : Tarbiyah dan Keguruan/Pend. Teknologi Informasi
Perguruan Tinggi : UIN Ar-Raniry
Bidang Keahlian : Sistem Informasi
Alamat l. Kantor : Prodi. Pendidikan Teknologi Informasi
Rumah : Perum. Alue Batedong Desa Neuheun Aceh Besar

Pengalaman Pendidikan (SI ke atas)

No	Perguruan Tinggi	Kota/Negara	Bid. Studi	Thn Lulus
1.	UPI “YPTK”	Padang	Sistem Informasi	2008
2.	UPI “YPTK”	Padang	Sistem Informasi	2011

Java adalah bahasa pemrograman yang akan digunakan dalam proses pembelajaran (praktikum) mata kuliah pemrograman visual bagi mahasiswa Pendidikan Teknologi Informasi UIN Ar-Raniry tahun ajaran 2015/2016. Sedangkan untuk berkenalan dengan bahasa pemrograman java, mahasiswa dapat merujuk pada milis-milis java berbahasa indonesia seperti: <http://www.jug.or.id>, <http://groups.yahoo.com/group/jug-joglosemar/> (pengguna java di jogja dan jawa tengah), <http://groups.yahoo.com/group/jug-geulis/> (pengguna java bandung), <http://tech.groups.yahoo.com/group/jug-bonek/> (pengguna java di jawa timur) ataupun forum resmi java-oracle berbahasa inggris di <http://www.oracle.com/technetwork/java/community/index.html>, sebagai pengantar singkat pengenalan bahasa pemrograman java, dapat dilihat pada penjelasan singkat di bawah ini.

Java merupakan bahasa pemrograman yang sejatinya berbasis object oriented programming (OOP) atau dalam terjemahan bahasa indonesia-nya lebih dikenal dengan pemrograman berorientasi objek (PBO). Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems, dan pada tahun 2010 java resmi diambil alih oleh oracle.

Dalam pembuatan kode (sintak)-nya, aplikasi-aplikasi berbasis java umumnya akan dikompilasi ke dalam p-code (bytecode) agar dapat dijalankan pada berbagai sistem operasi melalui java virtual machine (JVM), hal ini tidaklah berlebihan karena java dikenal dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan,

UNIMAL PRESS

