KIT BOOK

GENERATION
TEKNIK INFORMATIKA

PEMROGRAMAN BERORIENTASI OBYEK MENGGUNAKAN

JAVA & NETBEANS

Intan Nur Farida
Daniel Swanjaya

PEMROGRAMAN BERORIENTASI OBYEK MENGGUNAKAN JAVA DAN NETBEANS

Daniel Swanjaya, M.Kom.
Intan Nur Farida, M.Kom.

CV. KASIH INOVASI TEKNOLOGI

PEMROGRAMAN BERORIENTASI OBYEK MENGGUNAKAN JAVA DAN NETBEANS

Oleh :
Daniel Swanjaya, M.Kom.
Intan Nur Farida, M.Kom.

ISBN : 978-602-51918-9-3

Editor & Penyunting
Juli Sulaksono

Desain Cover
Ahmad Bagus Setiawan

Penerbit
CV. Kasih Inovasi Teknologi

Redaksi
Jl. KH. Hasyim Asyari Gg.1 Nusa Indah No.74,
Kota Kediri
Telp. +628563533234
Email : kasihinovasiteknologi@gmail.com

Cetakan Pertama,

Kata Pengantar

Puji syukur kehadirat Allah SWT atas limpahan rahmat dan karunianya sehingga Buku Pemrograman Berorientasi Obyek menggunakan Java dan NetBeans telah dapat diselesaikan. Buku ini merupakan bahan ajar dalam mata kuliah Pemrograman Berorientasi Obyek untuk mahasiswa semester III pada program studi Teknik Informatika Universitas Nusantara PGRI Kediri.

Terimakasih disampaikan kepada Ahmad Bagus Setiawan, ST., MM., M.Kom. selaku Ketua Program Studi Teknik Informatika beserta 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 maanfaat 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.

Kediri, Februari 2019

Tim Penulis

Daftar Isi

BAB 9 Java Database Connectivity

BAB 10 Java Report

BAB 1
PENDAHULUAN

1.1 Mengenal Java

Program Komputer atau yang biasa disebut dengan Program adalah kumpulan perintah yang ditulis untuk melakukan suatu fungsi spesifik pada komputer sesuai maksud dari pembuat perintah. Sebuah program biasanya memiliki suatu bentuk model pengeksekusian tertentu agar dapat secara langsung dieksekusi oleh komputer. Program yang sama dalam format kode yang dapat dibaca oleh manusia disebut sebagai kode sumber (source code), bentuk program yang memungkinkan programmer menganalisis serta melakukan penelaahan Algoritma yang digunakan pada program tersebut. Kode sumber tersebut pada akhirnya dikompilasi oleh utilitas bahasa pemrograman tertentu sehingga membentuk sebuah program. bentuk alternatif lain model pengeksekusian sebuah program adalah dengan menggunakan bantuan interpreter, kode sumber tersebut langsung dijalankan oleh utilitas interpreter suatu bahasa pemrograman yang digunakan.

Java merupakan sebuah bahasa pemrograman yang berorientasi objek dan dapat dijalankan (Run) pada segala jenis sistem operasi komputer(OS) karena perkembangan Java ini tiak hanya terfokus pada satu sistem operasi saja tetapi dikembangkan untuk segala macam jenis sistem operasi dan bersifat open source

Java memiliki dua komponen yaitu :

A. JVM (*Java Virtual Machine*)

Java Virtual Machine (JVM), sebuah mesin abstrak yang dapat mengeksekusi Java *Bytcode* menjadi bahasa mesin kemudian menjalankannya. Serta tida tergantung dengan platform apapun. Tidak seperti bahasa pemrograman lain yang kodenya langsung dikompilasi sesuai

dengan mesin target dan sistem operasinya. JVM adalah mesin dalam mesin yang meniru mesin inangnya dan dapat menjalankan kode Java yang Anda tulis dimanapun. Setiap mesin yang menginstal JVM, dapat menjalankan kode program yang telah Anda buat. JVM dibangun oleh James Gosling. JVM memiliki dua komponen yaitu :

1. *Java Runtime Environment* (JRE) atau yang biasa disebut Java, merupakan bagian yang memungkinkan program Java dapat berjalan di Komputer.
2. *Java Development Kit* (JDK) berisi sekumpulan tools perintah (*command line tool*) untuk menciptakan program java. Dua komponen utama JDK adalah :
    a. Kompilator, program "javac" untuk mengompilasi file kode sumber Java menjadi kelas *bytecode*.
    b. Interpreter, untuk menjalankan program *bytecode* Java.

B. IDE (Integrated Development Environment)

Program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Dengan menggunakan Java IDE (Integrated Development Environment) tertentu, semua kebutuhan pemrograman akan dijadikan menjadi satu tempat. Mulai dari text editor, compiler/interpreter, system help dan terkadang juga terdapat fitur lain yang sangat bermanfaat dalam penulisan kode (seperti: code auto-complete dan syntax highlight). Contoh Java IDE : Netbeans dan Eclipse

1.2 Java Standard Edition & NetBeans IDE

Untuk dapat mengkompilasi dan menjalankan program-program Java, Anda terlebih dahulu harus memasang (*install*) dan mengkonfigurasi *Java Development Kit* (JDK). Proses instalasi dan konfigurasi *Java Development Kit* sangat mudah. Java edisi standar (J2SE) versi terbaru dapat ditemukan dan diunduh secara gratis dari situs resmi atau situs-situs lain di internet. Namun sangat disarankan Anda mengunduh JDK dari situs resminya di https://www.oracle.com/technetwork/java/javase/downloads/.

IDE (*Integrated Development Environment*) adalah program 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. Pada buku pemrograman berorientasi obyek ini digunakan IDE NetBeans. NetBeans

adalah sebuah aplikasi dari java oracle dalam membuat aplikasi yang berbasis java. NetBeans dapat diunduh di https://netbeans.org/downloads/ . Oracle juga menyediakan paket bundling JDK dan NetBeans IDE, yang memungkinkan kita memasang JDK sekaligus dengan NetBeans IDE.

1.3 Bekerja dengan NetBeans

Pada NetBeans semua perancangan dan pemrograman dilakukan di dalam kerangka sebuah proyek. Proyek NetBeans merupakan sekumpulan file yang dikelompokkan di dalam satu kesatuan.

Membuat Project Baru

Untuk project baru & membuat program dengan NetBeans IDE ikuti langkah-langkah berikut :

1. Buka NetBeans IDE
2. Pilih File > New Project
3. Pilih Java > Java Application > Next
4. Setelah muncul tampilan “New Java Application”, lakukan :
   a. Ubah nama project menjadi “PRAKTIKUM_PBO_2018”
   b. Tentukan lokasi penyimpanan, misal “D:\NetBeansProject”. Klik tombol Browse untuk mengganti lokasi penyimpanan.
   c. Klik Finish
5. Ketikkan baris perintah seperti pada gambar 1. 1.
6. Untuk proses compile & interpreter (Run) tekan tombol keyboard “Shift” dan “F6” bersama-sama. Bila program yang akan dijalankan adalah program utama (main program) dari project, maka untuk menjalankan main program cukup tekan tombol keyboard “F6” atau tekan tombol Run Project di toolbar.
7. Jika tidak ada kesalahan perintah atau runtime error, maka NetBeans akan memunculkan window output dan tampilan seperti pada gambar 1.2.

Membuat Package baru

Untuk membuat Package (direktori) baru, ikuti langkah berikut :

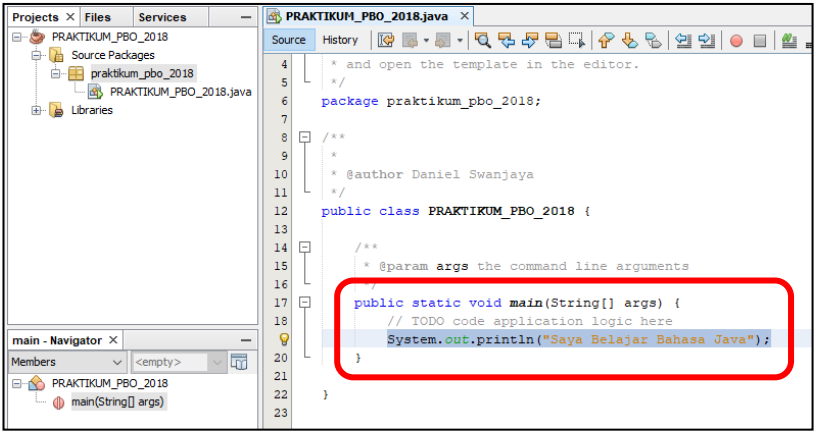
1. Klik kanan Project tempat package baru akan dibuat.
2. Pilih “New“ kemudian pilih “Java Package...”.
3. Ketikkan nama package yang baru “pertemuan_ke_01”, kemudian klik Finish.

4. Setelah berhasil dibuat maka package baru akan muncul seperti pada gambar 1.3.

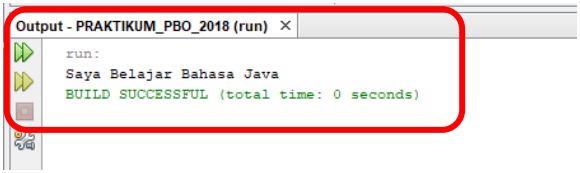
Membuat File Baru

Untuk membuat file class baru pada Package yang sudah ada, ikuti langkah berikut :

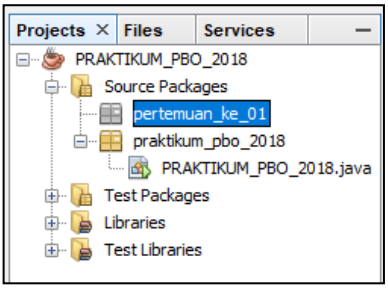
1. Klik kanan package tempat file class baru akan dibuat.
2. Pilih “New” kemudian pilih “Java Class...”.
3. Ketikkan nama file class baru “BelajarJava”, kemudian klik Finish.



Gambar 1. 1



Gambar 1. 2



Gambar 1. 3

1.4 Catatan Penting

Beberapa hal yang perlu diperhatikan ketika kita menggunakan Bahasa Java yaitu :

a. Nama Project, Package dan File hanya boleh menggunakan huruf, angka dan underscore, tetapi tidak boleh diawali dengan angka.

b. Nama Project, Package dan File bersifat case sensitive, dimana upper case dan lower case dibedakan.

c. Jika nama Project, Package dan File menggunakan lower case maka namanya tidak boleh menggunakan keyword Bahasa Java (gambar 1.4). Tetapi jika namanya mengandung keyword diperbolehkan. Misal : breakAndContinue.

d. Penamaan Project, Package dan File sesuaikan dengan tujuannya, untuk mempermudah pelacakan.

e. Syarat utama source code dapat di-Run adalah harus mempunyai main method dengan peletakan dan penulisan yang benar (gambar 1.5), dimana main method harus berada dalam public class. Dalam satu class hanya boleh terdapat satu main method.

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

Gambar 1. 4

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

    }
}
```

Gambar 1. 5

1.5 Latihan

1. Buatlah project baru dengan nama "**Latihan_PBO_2018**"
2. Pada file **Latihan_PBO_2018.java** Ketikkan baris perintah seperti pada gambar 1.6.
3. Run **Latihan_PBO_2018.java** tersebut (output 1)!
4. Ubah file **Latihan_PBO_2018.java** seperti gambar 1.7.
5. Run **Latihan_PBO_2018.java** yang telah diubah (output 2)!
6. Dari output 1 dan output 2 apa yang dapat Anda simpulkan ?
7. Buat Package baru dengan nama "**latihan_01**", kemudian buat file class baru dengan nama **latihanCetakString.java**. Kemudian buat baris program untuk menampilkan output seperti gambar 1.8.

```
public static void main(String[] args) {
    System.out.println("Saya sedang mengikuti Praktikum");
    System.out.println("Pemrograman Berorientasi Obyek");
}
```

Gambar 1. 6

```
public static void main(String[] args) {
    System.out.print("Saya sedang mengikuti Praktikum");
    System.out.print("Pemrograman Berorientasi Obyek");
}
```

Gambar 1. 7

```
run:
Kami Belajar
Kami Berlatih
Kami Mahir
BUILD SUCCESSFUL (total time: 0 seconds)
```

Gambar 1. 8

BAB 2
DASAR JAVA

2.1 Identifier

Identifier (Pengenal) adalah suatu nama yang digunakan dalam program untuk menyatakan variabel, fungsi, dll. Aturan umum yang berlaku bagi pengenal :

a. Diawali dengan huruf, underscore atau simbol dollar
b. Selanjutnya dapat berupa huruf, angka atau underscore.
c. Bersifat case sensitive, dimana huruf kecil (lower case) dan kapital (upper case) dianggap berbeda.

2.2 Java Literals

Literals adalah tanda bahwa tidak terjadi perubahan atau konstan. Macam-macam literals dalam Java adalah : *Integer Literals*, *Floating-Point Literals*, *Boolean Literals*, *Character Literals* dan *String Literals*. Literals adalah tanda bahwa tidak terjadi perubahan atau konstan. Macam-macam literals dalam Java adalah : *Integer Literals*, *Floating-Point Literals*, *Boolean Literals*, *Character Literals* dan *String Literals*.

a. Integer Literals

   Integer literals dibedakan dalam beberapa format yang berbeda: **desimal** (berbasis 10), **heksadesimal** (berbasis 16), and **oktal** (berbasis 8). Dalam penggunaan tipe data integer pada program, kita harus mengikuti aturan penggunaan beberapa notasi khusus. Untuk angka desimal, kita tidak memerlukan notasi khusus. Kita hanya menulis angka desimal seperti apa adanya. untuk angka heksadesimal, hal itu harus ditandai oleh “0x” atau “0X”. untuk oktal, ditandai oleh“0”. Sebagai contoh, mewakili angka **12**.

Penulisan dalam bentuk desimalnya adalah **12**, Sementara dalam heksadesimal, menjadi **0xC**, dan dalam oktal, nilai tersebut sama dengan
014.

b. Floating-Point Literals
Floating point literals mewakili bentuk desimal dengan bagian yang terpisah. Sebagai contoh adalah 3.1415. *Floating point literals* dapat dinyatakan dalam notasi standard atau scientific. Sebagai contoh, 583.45 dinyatakan dalam notasi standard, Sementara 5.8345e2 dinyatakan dalam notasi scientific. Default *Floating point literals* mempunyai tipe data **double** yang dinyatakan dalam 64-bit. Untuk menggunakan ketelitian yang lebih kecil (32-bit) **float**, hanya dengan menambahkan karakter "f" atau "F".

c. Boolean Literals
Boolean literals hanya memiliki dua nilai, true atau false.

d. Character Literals
Character Literals diwakili oleh karakter single Unicode.Karakter Unicode adalah 16-bitcharacter set yang menggantikan 8-bit ASCII character set. Unicode memungkinkanpengunaan simbol dan karakter khusus dari bahasa lain. Untuk menggunakan *character literals*, karakter tersebut di dalam tanda *single pute* (' ') (single quote delimiters). Sebagai contoh huruf a, diwakili sebagai **'a'**.

e. String Literals
String literals mewakili beberapa karakter dan dinyatakan dalam tanda double pute(" ")(double quotes). Sebagai contoh *string literal* adalah, **"Belajar Java".**

2.3 Tipe Data

Tipe Data Java dibagi menjadi dau kategori yaitu Sederhana (Primitif) dan Komposit (Reference). Tipe Data Sederhana merupakan tipe inti, tipe data sederhana tidak diturunkan dari tipe lain. Tipe ini juga disebut tipe data primitif karena hampir sama seperti pada bahasa pemograman lainnya. Tipe Data Komposit adalah tipe yang disusun dari tiper sederhana atau tipe komposit lain yang sudah ada. Beberapa tipe komposit antara lain : string, array, class, interface.

Terdapat delapan tipe primitif di Java :

a. Tipe bilangan bulat : byte, short, int, long
b. Tipe bilangan pecahan : float, double
c. Tipe karakter : char (bisa juga untuk bilangan bulat)
d. Tipe logika : boolean

Masing- masing tipe data memliki kebutuhan memori yang berbeda, besarnya memori yang dipakai menentukan jangkauan nilai dari tipe data tersebut. Untuk mengetahui jangkauan nilai dari masing-masing tipe, eksekusi (Run) source code **jangkauanNilaiTipeData.java**. Sebelum mengetikan source code, buat terlebih dahulu Package baru dengan nama "pertemuan_ke_02" pada Project "PRAKTIKUM_PBO_2018", kemudian buat fileclass baru dengan nama **jangkauanNilaiTipeData.java**di Package "pertemuan_ke_02".

jangkauanNilaiTipeData.java

```
package pertemuan_ke_02;
public class jangkauanNilaiTipeData {
public static void main(String[] args) {
System.out.println("Jangkauan Nilai Tipe Data
Sederhana.");
System.out.println("byte : "+Byte.MIN_VALUE+" s.d "
+Byte.MAX_VALUE);
System.out.println("short : "+Short.MIN_VALUE+" s.d "
+Short.MAX_VALUE);
System.out.println("int : "+Integer.MIN_VALUE+" s.d "
+Integer.MAX_VALUE);
System.out.println("long : "+Long.MIN_VALUE+" s.d "
+Long.MAX_VALUE);
System.out.println("float : "+Float.MIN_VALUE+" s.d "
+Float.MAX_VALUE);
System.out.println("double : "+Double.MIN_VALUE+" s.d "
+Double.MAX_VALUE);
}
}
```

2.4 Variabel

Variabel adalah unit dasar penyimpan di program Java. Variabel memiliki tipe data dan nama. Tipe data menandakan tipe nilai yang dapat dibentuk oleh variabel itu sendiri. Nama variabel harus mengikuti aturan untuk identifier.

a. Deklarasi dan Inisialisasi Variabel

<data tipe><name> [=initial value];

Hal yang berada diantara **<>** adalah hal yang harus ada, sementara hal yang di dalamtanda **[]** bersifat optional. Misal :

```
int alpha;
char hurufAwal;
int betha = 10;
char hurufAkhir = 'Z';
```

Untuk mendeklarasikan dan menginisialisasi beberapa variabel dengan tipe yang sama dapat pula dilakukan dengan cara berikut :

```
<data tipe><name1> [=initial value],
<name2> [=initial value], ...;
```

Misal :

```
int alpha, betha = 10;
char hurufAwal, hurufAkhir = 'Z';
```

Untuk lebih memahami Deklarasi dan Inisialisasi Variabel dan cara mengaksesnya eksekusi (Run) source code **tesVariabel.java**, tempatkan **tesVariabel.java** di Package "pertemuan_ke_02".

b. Variabel Reference dan Variabel Primitif

Variabel primitif adalah variabel dengan tipe data primitif.Variabel ini menyimpan data dalamlokasi memori yang sebenarnya dimana variabel tersebut berada.

Variabel Reference adalah variabel yang menyimpan alamat dalam lokasi memori. Yangmenunjuk ke lokasi memori dimana data sebenarnya berada. Ketika Anda mendeklarasikanvariabel pada class tertentu, Anda sebenarnya mendeklarasikan reference variable dalambentuk objek dalam classnya tersebut.

c. Konversi Tipe Data

Dalam kasus-kasus pemrograman tertentu, ada kalanya perlu mengubah tipe data dari satu tipe ke tipe lainnya. Apabila kedua tipe tersebut kompatibel atau cocok, maka Java akan melakukan konversi tipe secara otomatis. Sebagai contoh ketika memerlukan nilai ASCII atau UNICODE dari variabel bertipe char, maka kita perlu mengubah tipe char menjadi int, seperti file **tipeCasting.java**. contoh lainnya, jika kita hendak meminta masukkan nilai dari keyboard menggunakan JOptionPane, dimana masukkan dari keyboard disimpan dengan tipe String, tetapi data

yang kita butuhkan bertipe bilangan bulat (int), maka kita mengkonversi tipe String ke tipe int (**StringToInt.java**).

tesVariabel.java

```
package pertemuan_ke_02;
public class tesVariabel {
public static void main(String[] args) {
byte byteA = 127, byteB = -128;
short dd = 17, mm = 8, yy = 1945;
    int Desimal = 17, Heksa = 0xABC, Oktal = 017;
    long banyakPenduduk = 256231897;
char hurufAwal = 'A', hurufAkhir = 'Z';
    float luasTanah = 564231.789F, volumeTandon = 9.78E5F;
    double totalDebitAir = 987.654E10;
    String labelVolume = "Total Debit Air",
labelDebit = "\nTotal Debit Air";

    System.out.println("byteA = "+byteA+"\nbyteB = "+byteB);
    System.out.println("Tanggal : "+dd+"/"+mm+"/"+yy);
    System.out.println("Angka Desimal = "+Desimal);
    System.out.println("Angka Heksa = "+Heksa+
"\nAngka Oktal = "+Oktal);
System.out.println("Banyak Penduduk = "+banyakPenduduk);
System.out.println("huruf Awal = "+hurufAwal+
" ; Huruf Akhir = "+hurufAkhir);
System.out.println("Luas Tanah = "+luasTanah+
"\n"+labelVolume+" = "+
volumeTandon+labelDebit+totalDebitAir);
}
}
```

tipeCasting.java

```
package pertemuan_ke_02;
public class tipeCasting {
public static void main(String[] args) {
char huruf1 = 'A', huruf2;
    int kodeASCII_1 = (int) huruf1, kodeASCII_2 = 75;
    huruf2 = (char) kodeASCII_2;
    System.out.println("Huruf ke-1 = "+huruf1);
    System.out.println("Kode ASCII Huruf ke-1 = "+
kodeASCII_1);
System.out.println("Kode ASCII Huruf ke-2 = "+
kodeASCII_2);
System.out.println("Huruf ke-2 = "+huruf2);
}
}
```

tipeCasting.java

```
package pertemuan_ke_02;
import javax.swing.JOptionPane;
public class StringToInt {
public static void main(String[] args) {
String masukkan;
    masukkan = JOptionPane.showInputDialog(null,
"Berapakah umur Anda ?");
System.out.println("Umur Anda = "+
Integer.parseInt(masukkan));
}
}
```

2.5 Operator

Dalam Java, ada beberapa tipe operator. Ada operator aritmatika, operator relasi, dan operator logika.

a. Operator Aritmatika

Operator	Penggunaan	Keterangan
+	op1 + op2	Penjumlahan
-	op1 - op2	Pengurangan
*	op1 * op2	Perkalian
/	op1 / op2	Pembagian
%	op1 % op2	Modulus

b. Operator Increment dan Decrement

Operator	Penggunaan	Keterangan
++	op++	Menambahkan nilai 1 pada op; mengevaluasi nilai op sebelum ditambahkan
++	++op	Menambahkan nilai 1 pada op; mengevaluasi nilai op setelah ditambahkan
--	op--	Mengurangkan nilai 1 pada op; mengevaluasi nilai op sebelum ditambahkan
--	--op	Mengurangkan nilai 1 pada op; mengevaluasi nilai op setelah ditambahkan

c. Operator Relasi

Operator	Penggunaan	Keterangan
>	op1 > op2	op1 lebih besar dari op2
>=	op1 >= op2	op1 lebih besar dari atau sama dengan op2
<	op1 < op2	op1 kurang dari op2
<=	op1 <= op2	op1 kurang dari atau sama dengan op2
==	op1 == op2	op1 sama dengan op2
!=	op1 != op2	op1 tidak sama dengan op2

d. Operator Logika

Operator	Penggunaan	Keterangan
&&	op1 && op2	Operator AND • Mengembalikan nilai true jika op1 dan op2 bernilai true • nilai false jika salah satunya atau keduanya bernilai false
||	op1 || op2	Operator OR • Mengembalikan nilai false jika op1 dan op2 bernilai false • nilai true jika salah satunya atau keduanya bernilai true
!	!op1	Operator NOT • Mengembalikan nilai true jika op1 bernilai false • nilai false jika op1 bernilai true

e. Operator Bitwise

Operator	Penggunaan	Keterangan
&	op1 & op2	Operator AND • Mengembalikan nilai true jika op1 dan op2 bernilai true • nilai false jika salah satunya atau keduanya bernilai false
|	op1 | op2	Operator OR • Mengembalikan nilai false jika

		op1 dan op2 bernilai false • nilai true jika salah satunya atau keduanya bernilai true
^	op1 ^ op2	Operator XOR (Exclusive OR) • Mengembalikan nilai false jika op1 dan op2 nilainya sama • nilai true nilai keduanya tidak sama
~	~op1	Operator NOT Menberikan hasil dengan masing-masing bit berupa kebalikan dari bit op1.
>>	op1 >> op2	Menggeser bit-bit op1 ke kanan sebanyak op2
>>>	op1 >>> op2	Menggeser bit-bit op1 ke kanan sebanyak op2, tetapi bit terkanan diisi dengan nol. Sehingga hasilnya selalu bilangan positif.
<<	op1 << op2	Menggeser bit-bit op1 ke kiri sebanyak op2

2.6 Operasi Masukkan

Untuk mengisi variable melalui input keyboard Java menggunakan Scanner (komponen siap pakai untuk input output) untuk melakukan proses ini. Untuk penggunaanya bisa dilihat pada **inputKeyboard.java**.

inputKeyboard.java

```
package pertemuan_ke_02;
import java.util.Scanner;
public class inputKeyboard {
public static void main(String[] args) {
String NPM, Nama; int tglLahir, blnLahir, thnLahir;
    Scanner input = new Scanner(System.in);
    System.out.print("NPM  : "); NPM = input.next();
    System.out.print("Nama : "); Nama = input.next();
    System.out.print("Tanggal Lahir : ");
tglLahir = input.nextInt();
    System.out.print("Bulan Lahir   : ");
blnLahir = input.nextInt();
System.out.print("Tahun Lahir   : ");
thnLahir = input.nextInt();
```

```
    System.out.println("Resume Data Diri:");
System.out.println("NPM Anda "+NPM+", Nama Anda "+Nama);
     System.out.println("Tanggal Lahir Anda "+
tglLahir+"/"+blnLahir+"/"+thnLahir);
}
}
```

2.7 Catatan Penting

a. Di Java terdapat kode escape, kode karakter yang penulisannya diawali dengan simbol "\". Tabel 2.1 menunjukkan beberapa karakter escape. codeEscape.java menunjukkan penggunaan kode escape.

Tabel 2. 1

Kode	Keterangan
\b	Backspace
\f	Formfeed
\n	Newline
\r	Carriage Return
\t	Tab
\'	Petik Tunggal
\"	Petik Ganda
\\	Backslash
\ddd	Oktal (ddd = 0 s.d 377)
\u00dd	Heksadesimal (dd = 0 s.d FF)

codeEscape.java

```
package pertemuan_ke_02;
public class codeEscape {
public static void main(String[] args) {
System.out.println("BACKS\bPACE");
    System.out.println("FORMFEED\fFORMFEED");
    System.out.println("GANTI\nBARIS");
    System.out.println("CARRIAGE\rRETURN");
    System.out.println("TAB\tTAB");
    System.out.println("HARI JUM\'AT");
    System.out.println("LAGU \"INDONESIA RAYA\"");
    System.out.println("C:\\Program Files\\Java\\");
    System.out.println("OKTAL \101 \102 \103");
    System.out.println("HEKSA \u0051 \u0052 \u0053");
}
}
```

b. Untuk penulisan rumus yang komplek, gunakan tanda kurung untuk memastikan hirarki pengerjaan. Misal : Jika hendak menuliskan $\frac{Alpha}{Betha \times Gamma}$ jangan mengetik tanpa tanda kurung, Alpha/Betha*Gamma karena yang akan dikerjakan terlebih dahulu adalah Alpha/Betha bukan Betha*Gamma, jadi penulisan yang benar adalah Alpha/(Betha*Gamma).

c. Penamaan Variabel dalam satu blok pernyataan tidak boleh duplikat, kecuali berbeda blok. Seperti pada Gambar 2.1.

d. Bilangan bulat jika dibagi bilanga bulat akan menghasilkan bilangan bulat juga, jika kita membutuhkan hasil dalam bilangan pecahan maka kita bisa melakukan 2 hal yaitu Konversi Tipe atau Perkalian dengan bilangan 1.0 seperti pada gambar 2.2.

e. Penyederhanaan operator aritmatika, Jika pada increment dan decrement kita bisa menyingkat penulisan A = A+1 menjadi A++ dan A = A-1 menjadi A--, maka untuk menyingkat A = A+2 kita bisa menggunakan A += 2, A = A-3 menjadi A-=3, A=A*4 menjadi A*=4, A=A/5 menjadi A/=5, dan A = A%6 menjadi A%=5.

```
public static void main(String[] args) {
    int Alpha = 1;
    {
        Alpha = 5;
        int Betha = 2, Gamma = 3;
        System.out.println("Alpha = "+Alpha);
        System.out.println("Betha = "+Betha);
        System.out.println("Gamma = "+Gamma);
    }

    {
        Alpha = 7;
        int Betha = 4, Gamma = 5;
        System.out.println("Alpha = "+Alpha);
        System.out.println("Betha = "+Betha);
        System.out.println("Gamma = "+Gamma);
    }
}
```

Gambar 2. 1

```
public static void main(String[] args) {
    int Alpha = 1, Betha = 3;
    System.out.println("Alpha / Betha = "+((double)Alpha/Betha));
    System.out.println("Alpha / Betha = "+(1.0*Alpha/Betha));
}
```

Gambar 2. 2

2.8 Latihan

1. Buat Package baru pada "**Latihan_PBO_2018**" dengan nama "**latihan_02**".
2. Buat File Class baru untuk mengkonversi suhu Celcius ke Fahrenheit, dengan nama **Celcius_Fahrenheit.java**.
3. Run **Celcius_Fahrenheit.java** tersebut!
4. Apakah hasil dari konversi dapat berupa bilangan pecahan? Jika belum ubah **Celcius_Fahrenheit.java** sehingga dapat memberikan hasil berupa bilangan pecahan, dan berikan alasan Anda !
5. Buat File Class baru untuk mengkonversi suhu Fahrenheit ke Celcius, dengan nama **Fahrenheit_Celcius**.java. Suhu Fahrenheit diinputkan dari keyboard dan dapat berupa bilangan pecahan, serta hasil yang diberikan juga berupa bilangan pecahan.
6. Buat File Class baru (**ReamurToCFK.java**) untuk mengkonversi suhu Reamur ke Celcius, Fahrenheit dan Kelvin sekaligus, dimana suhu Reamur diinputkan dari keyboard. Input dan hasil dapat berupa bilangan pecahan.

Celcius_Fahrenheit.java

```
package latihan_02;
import java.util.Scanner;
public class Celcius_Fahrenheit {
public static void main(String[] args) {
int Celcius, Fahrenheit;
Scanner input = new Scanner(System.in);
    System.out.println("Konversi Suhu Celcius ke Fahrenheit");
    System.out.print("Masukkan suhu Celcius : ");
Celcius = input.nextInt();
    Fahrenheit = (Celcius * 9) / 5 + 32;
    System.out.println(Celcius+"\u00B0C = "+
Fahrenheit+"\u00B0F");
}
}
```

BAB 3
STRUKTUR KONTROL, ARRAY & STRING

3.1 Struktur Kontrol Pemilihan

Kontrol seleksi digunakan untuk membuat pemilihan terhadap aksi yang akan dilakukan.

1. Pernyataan if

   Pernyataan *if* akan menentukan sebuah pernyataan (atau blok kode) yang akan eksekusi jika dan hanya jika persyaratan bernilai benar *(true)*. Apabila statement yang akan dijalankan hanya satu maka pernyataan if tidak memerlukan tanda kurung kurawal.
2. Pernyataan if..else..

   Pernyataan *if-else* digunakan apabila kita ingin mengeksekusi beberapa pernyataan dengan kondisi *true* dan pernyataan yang lain dengan kondisi *false*. Pernyataan pada bagian kondisi *else* dari blok *if-else* dapat menjadi struktur *if-else* yang lain. Kondisi struktur seperti ini mengijinkan kita untuk membuat seleksi persyaratan yang lebih kompleks.
3. Pernyataan switch..case

   Switch digunakan pada saat kita melakukan pemilihan yang kondisinya pasti. Setiap bagian case diakhiri pernyataan break supaya hanya statemen-statement pada case tersebut saja yang dijalankan. Bagian Default dijalankan jika semua PILIHAN tidak terpenuhi, bagian default harus dibagian akhir switch.
4. Operator ternary

Operator ternary digunakan untuk menggantikan pernyataan if-else yang masing-masing bagian hanya mengerjakan satu statement saja. Biasanya operator ternary digunakan untuk pemberian nilai pada variabel. Bentuk penggunaan operator ternary adalah :

```
Value = Kondisi ? val_1 : val_2;
```

Apabila Kondisi bernilai benar maka Value akan bernilai val_1 dan jika Kondisi salah maka Value akan bernilai val_2.

3.2 Struktur Kontrol Pengulangan

Struktur kontrol pengulangan adalah berupa pernyataan dari Java yang mengijinkan kita untuk mengeksekusi blok code berulang-ulang sesuai dengan jumlah tertentu yang diinginkan. Ada tiga macam jenis dari struktur kontrol pengulangan yaitu *while, do- while, dan for*.

1. Pernyataan while

   Pernyataan while adalah jenis pengulangan yang mendefinisikan kondisi di awal blok. Sehingga apabila kondisi tidak terpenuhi (bernilai false) maka proses pengulangan pun tidak akan pernah dilakukan, apabila statement yang akan dijalankan hanya satu boleh tidak menggunakan tanda kurung kurawal.
2. Pernyataan for

   Pernyataan for pada umumnya digunakan untuk melakukan pengulangan yang banyaknya sudah pasti atau sudah diketahui sebelumnya. Pertama kita harus mendefinisikan inisiasi dan kondisi untuk keluar dari perulangan, kemudian kita juga perlu menambahkan iterasi, yaitu variabel pengontrol untuk melakukan proses increment atau decrement.
3. Pernyataan do..while

   Pernyataan do..while sebenarnya mirip dengan pernyataan while. Perbedaannya hanya terletak pada penempatan kondisinya saja. Pada pernyataan while kondisi ditempatkan di awal blok pengulangan sedangkan pada pernyataan do..while kondisinya berada di akhir blok. Sehingga proses pengulangan akan dilakukan minimal sekali meskipun ternyata kondisinya tidak terpenuhi(bernilai false). Pada pernyataan do..while tidak disarankan untuk menghilangkan tanda kurung kurawal, sekalipun statement yang akan dijalankan hanya satu.
4. Pernyataan break

Dengan menggunakan pernyataan break, kita dapat menghentikan proses pengulangan tertentu tanpa memperdulikan lagii kondisi yang didefinisikan ataupun sisa statement-statement yang terdapat pada blok pengulangan tersebut. Sehingga ketika pernyataan break dijalankan maka proses perulangan akan langsung dihentikan dan eksekusi program akan berlanjut ke bagian kode setelah blok perulangan tersebut.

5. Pernyataan continue

   Pernyataan continue digunakan untuk memaksa program agar melanjutkan proses pengulangan. Berbeda dengan pernyataan break yang ketika dieksekusi menghentikan proses perulangan, pernyataan continue hanya mengabaikan sisa statement-statement yang terdapat pada blok pengulangan tersebut, dan melanjutkan kembali pada awal blok perulangan. Pernyataan continue ini tidak dianjurkan digunakan pada pernyataan while dan do..while, karena dapat menyebabkan perulangan tidak berakhir.

3.3 Array

Array merupakan sekumpulan obyek yang memiliki tipe data yang sama dan dapat di akses secara random dengan menggunakan index. Array mempunyai panjang yang tetap, artinya ketika kita mendeklarasikan suatu array dengan panjang 10, maka array tersebut panjangnya akan tetap 10 walaupun kita hanya memakai 5 elemen.

1. Deklarasi array

   <type data> nama_variabel[ ]; atau

   <type data>[ ] nama_variabel;

2. Pembuatan dan alokasi array

   nama_variabel=new <type data>[jumlah elemen];

   atau (sekaligus deklarasi)

   <type data> nama_variabel[ ] =
   new <type data>[jumlah elemen];

3. Inisialisasi elemen array

   <type data>nama_variabel[ ]=
   {nilai1,nilai2,nilai3,…,nilai_n};

4. Pengaksesan array

   nama_variabel[indeks]

5. Panjang Array
   `nama_variabel.length`
6. Array multidimensi
   Array multidimensi merupakan array yang mengandung array. Yang membedakan dengan array satu dimensi adalah penggunaan kurung kurawal, dimana banyaknya pasangan kurung kurawal menentukan besarnya dimensinya. Keistimewaan array multi dimensi pada Java adalah panjang array fleksibel.

3.4 Operasi String

Operasi String adalah segala manipulasi yang dapat dilakukan terhadap isi variable bertipe string antara lain, menggabungkan, memotong, mencari sebuah karakter dalam sekumpulan string, mengubah huruf besar ke kecil dan sebaliknya, dan berbagai macam fungsi string lainnya.

1. Menggabung string
   Untuk menggabung string kita bisa menggunakan operator + atau fungsi **concat**.
2. Memotong string
   Untuk mengambil sebagian karakter dari suatu string kita menggunakan fungsi **substring**.
3. Membandingkan string
   Untuk membandingkan string kita menggunakan fungsi equals, **equalsIngnoreCase**, **compareTo**, **compareToIgnoreCase**. Kita tidak bisa menggunakan operator == untuk operasi string, ataupun operator relasi lainnya seperti >, >=, <, <=, !=.
4. Mengubah huruf besar ke kecil dan sebaliknya
   Untuk mengubah huruf besar ke kecil kita menggunakan fungsi **toLowerCase**, untuk sebaliknya kita menggunakan **toUpperCase**.
5. Panjang String
   Untuk mengetahui panjang string kita menggunakan fungsi **length()**.
6. Mengganti karakter dalam string
   Untuk mengganti satu karakter menjadi karakter lain dalam sebuah string kita menggunakan fungsi **replace**

3.5 Contoh Program

Buat Package baru "**pertemuan_ke_03**" pada Project **PRAKTIKUM_PBO_2018**, kemudian coba beberapa contoh file class berikut.

tesPemilihan.java

```
package pertemuan_ke_03;
import java.util.Scanner;
public class tesPemilihan {
public static void main(String[] args) {
int absen, persenTarget, gajiPokok, bonus;
    Scanner input = new Scanner(System.in);
System.out.print("Dalam sebulan anda tidak masuk berapa
kali ? ");
absen = input.nextInt();
    System.out.print("Pencapaian target penjualan bulan
ini (%)? ");
persenTarget = input.nextInt();
    System.out.print("Gaji Pokok ? ");
gajiPokok = input.nextInt();

    if(absen > 5)
System.out.println("Siap-siap mendapat surat
teguran !");

if(persenTarget > 80){
System.out.println("Anda mendapat bonus sebesar 10%");
      bonus = (10/100) * gajiPokok;
    } else if(persenTarget > 60) {
System.out.println("Anda mendapat bonus sebesar 5%");
      bonus = (5/100) * gajiPokok;
    } else {
System.out.println("Maaf Anda tidak mendapat bonus");
      bonus = 0;
    }

    System.out.println("Gaji yang Anda terima = "
+(gajiPokok+bonus));
}
}
```

operatorTernary.java

```
package pertemuan_ke_03;
import java.util.Scanner;
public class operatorTernary {
public static void main(String[] args) {
int pinjaman, persenBunga;
Scanner input = new Scanner(System.in);

    System.out.print("Besar Pinjaman Rp. ? ");
pinjaman = input.nextInt();
    persenBunga = pinjaman > 10000000 ? 10 : 5;
```

```
    System.out.println("Anda dibebani bunga Rp."
+(persenBunga/100)*pinjaman);
System.out.println("Dana yang harus Anda kembalikan Rp."
+(pinjaman+(persenBunga/100)*pinjaman));
}
}
```

whileDuaPangkat.java

```
package pertemuan_ke_03;
import java.util.Scanner;
public class whileDuaPangkat {
public static void main(String[] args) {
int pangkat, angka = 1, count = 1;
Scanner input = new Scanner(System.in);
    System.out.print("masukkan pangkat (pangkat \u2265 0)= ");
pangkat = input.nextInt();

    while(count <= pangkat){ angka *= 2; count++; }

    System.out.println("2 pangkat "+pangkat+" = "+angka);
}
}
```

forDuaPangkat.java

```
package pertemuan_ke_03;
import java.util.Scanner;
public class forDuaPangkat {
public static void main(String[] args) {
int pangkat, angka = 1, count;
Scanner input = new Scanner(System.in);
    System.out.print("masukkan pangkat (pangkat \u2265 0)= ");
pangkat = input.nextInt();
    for (count = 0; count < pangkat; count++)
angka*=2;
    System.out.println("2 pangkat "+pangkat+" = "+angka);
}
}
```

doWhileMenu.java

```
package pertemuan_ke_03;
import java.util.Scanner;
public class doWhileMenu {
public static void main(String[] args) {
int channel; Scanner input = new Scanner(System.in);
    do{
System.out.println("Pilihan Channel TV :\n1. RCTI\n2.
SCTV\n3. NET TV");
      System.out.println("lainnya untuk matikan TV\n
Masukkan pilihan Anda : ");
      channel = input.nextInt();
```

```
      switch(channel){
case 1 :
System.out.println("Selamat menyaksikan RCTI OK.");
break;
        case 2 :
System.out.println("SCTV satu untuk semua...");
          break;
        case 3 :
System.out.println("NET TV, Televisi masa kini...");
          break;
default:
System.out.println("Anda mematikan TV.");
}
} while (channel > 0 && channel < 4);
    System.out.println("TV sudah mati.");
}
}
```

CariNomorRumah.java

```
package pertemuan_ke_03;
import java.util.Scanner;
public class CariNomorRumah {
public static void main(String[] args) {
int nomorYangDicari, nomorUrut;
Scanner input = new Scanner(System.in);

    System.out.print("Masukkan Nomor rumah yang Anda cari
(1 s.d 50): ");
nomorYangDicari = input.nextInt();
    for (nomorUrut = 0; nomorUrut < 50; nomorUrut++) {
System.out.print(nomorUrut+" --> ");
      if (nomorUrut == nomorYangDicari) {
System.out.println("KETEMU"); break;
} else  System.out.println("bukan");
}
}
}
```

dataStatistik.java

```
package pertemuan_ke_03;
public class dataStatistik {
public static void main(String[] args) {
int DATA[] = {11,13,17,19,23,29,31,37};
int jumlahData = 0, indeks;

for (indeks = 0; indeks < DATA.length; indeks++) {
System.out.println("DATA indeks ke-"+indeks+
" = "+DATA[indeks]);
```

```
      jumlahData += DATA[indeks];
    }

    System.out.println("Jumlah Data = "+jumlahData);
    System.out.println("Banyak Data = "+DATA.length);
}
}
```

titikKoordinat.java

```
package pertemuan_ke_03;
public class titikKoordinat {
public static void main(String[] args) {
int Koordinat[][] = {{11,13},{17,19},{23,29},{31,37}} ;
    int indeks, jumlahX = 0, jumlahY = 0;
double rataX,rataY;
    for (indeks = 0; indeks < Koordinat.length; indeks++) {
System.out.println("Koordinat indeks ke-1 : ("
+Koordinat[indeks][0]+","+Koordinat[indeks][1]+")");
      jumlahX += Koordinat[indeks][0];
      jumlahY += Koordinat[indeks][1];
    }

    rataX = (double) jumlahX/Koordinat.length;
    rataY = (double) jumlahY/Koordinat.length;
    System.out.println("Titik tengah ("+rataX+","+rataY+")");
}
}
```

arrayString.java

```
package pertemuan_ke_03;
import java.util.Scanner;
public class arrayString {
public static void main(String[] args) {
String NPM[], Nama[]; int banyakMHS, indeks;
    Scanner input = new Scanner(System.in);
    System.out.print("Banyak Mahasiswa = ");
banyakMHS = input.nextInt();
    NPM = new String[banyakMHS];
Nama = new String[banyakMHS];

    for (indeks = 0; indeks < banyakMHS; indeks++) {
System.out.println("Data Mahasiswa indeks ke-"+indeks);
      System.out.print("NPM  : ");
NPM[indeks] = input.next();
      System.out.print("Nama : ");
Nama[indeks] = input.next();
}
```

```
    System.out.println("Rekap Data MAHASISWA\nNPM\t\tNama");
    for (indeks = 0; indeks < banyakMHS; indeks++)
      System.out.println(NPM[indeks]+"\t"+Nama[indeks]);
}
}
```

3.5 Latihan

1. Buat Package baru pada "**Latihan_PBO_2018**" dengan nama "**latihan_03**".
2. Buat File Class baru untuk mendata NPM, Nama dan nilai Mahasiswa, dengan nama **nilaiAkademik.java**.
3. Run **nilaiAkademik.java** tersebut! Masukkan data seperti tabel 3.1.
4. Tambahkan mata kuliah IMK, Basis Data, Sistem Operasi & Struktur Data, kemudian tambahkan rata-rata nilai tiap mata kuliah dan rata-rata nilai per Mahasiswa.

Catatan : jika kita ingin menampilkan bilangan pecahan 1.3333333 menjadi 1.3, kita bisa menggunakan String.format("%,.1f",variabel), jika 2 angka dibelakang koma String.format("%,.2f",variabel). Contoh penggunaan :

```
System.out.println("Rata-rata Mata Kuliah = "+
String.format("%,.2f",rataMataKuliah));
```

nilaiAkademik.java

```
package latihan_03;
import java.util.Scanner;
public class nilaiAkademik {
public static void main(String[] args) {
String NPM[], Nama[]; int PBO[], banyakMhs, indeks;
    Scanner input = new Scanner(System.in);
    System.out.print("Banyak Mahasiswa = ");
banyakMhs = input.nextInt();

    NPM = new String[banyakMhs];
Nama = new String[banyakMhs];
    PBO = new int[banyakMhs];

    for (indeks = 0; indeks < banyakMhs; indeks++) {
System.out.println("Data Mahasiswa ke-"+(indeks+1));
      System.out.print("NPM  : ");
NPM[indeks] = input.next();
      System.out.print("Nama : ");
```

```
Nama[indeks] = input.next();
      do {
System.out.print("Nilai PBO : ");
PBO[indeks] = input.nextInt();
        if(PBO[indeks] < 0 || PBO[indeks] > 100)
System.out.println("Nilai antara 0 - 100.
Silakan input ulang.");
}while(PBO[indeks] < 0 || PBO[indeks] > 100);
}

System.out.println("Rekap data nilai
Mahasiswa\nNPM\tNama\tPBO");
for (indeks = 0; indeks < banyakMhs; indeks++)
System.out.println(NPM[indeks]+"\t"+Nama[indeks]+
"\t"+PBO[indeks]);
}
}
```

Tabel 3. 1

NPM	Nama	PBO
14.1.03.02.0009	Supriadi	95
14.1.03.02.0163	Nurin nakmah	60
14.1.03.02.0180	Davit Hartono	65
14.1.03.02.0188	Sonia Goretty	75
14.1.03.02.0247	Indra Pradana	90
14.1.03.02.0249	Ryo Rangga	85

BAB 4
Method

4.1 Method

Method adalah sub program yang memungkinkan programmer untuk membagi program dengan membagi masalah ke dalam beberapa sub masalah yang bisa diselesaikan secara modular. Method ditujukan untuk menangani suatu tindakan dengan tujuan untuk mempermudah pembuatan program mengingat method bisa dipanggil berkali-kali dalam suatu program. Umumnya method dibedakan menjadi 2, method yang mengembalikan nilai dan tidak mengembalikan nilai.

Pada method nilai yang dikirim disebut argumen dan yang menerima disebut parameter.Urutan pendeklarasian method adalah bebeas, tidak ada aturan khusus yang mengatur.

4.2 Method yang mengembalikan nilai

Method yang mengembalikan nilai atau biasa disebut fungsi, memiliki bentuk deklarasi sabagai berikut :

```
public static <tipe_data><nama_method>
([parameter_1,parameter_2, ...]){
  Statement_1;
  Statement_2;
  ...
  return variabel;
}
```

4.3 Method yang tidak mengembalikan nilai

Method yang tidak mengembalikan nilai atau biasa disebut prosedur, memiliki bentuk deklarasi sebagai berikut :

```
        public static void <nama_method>
        ([parameter_1,parameter_2, ...]){
          Statement_1;
          Statement_2;
          ...
}
```

4.4 Penamaan method

Penamaan method mengikuti aturan penamaan identifier yaitu diawali dengan huruf, kemudian diikuti dengan huruf, angka atau underline.

4.5 Variabel lokal dan global

Variabel global adalah variabel yang dikenal dan dapat digunakan dalam lingkungan satu kelas, dan variabel lokal hanya dikenal dan dapat digunakan di satu blok yang dibatasi kurung kurawal { dan }.

4.6 Overloading Method

Dalam lingkup satu kelas diperbolehkan terdapat beberapa method yang memiliki nama yang sama, dengan batasan kombinasi tipe data method, nama method dan parameter tidak boleh ada yang sama.

4.7 Rekursif

Rekursif adalah method yang memanggil dirinya sendiri berulang-ulang hingga tercapai suatu kondisi yagn membuatnya berhenti.

4.8 Contoh Program

Buat Package baru "**pertemuan_ke_04**" pada Project **PRAKTIKUM_PBO_2018**, kemudian coba beberapa contoh file class berikut.

cariNamaByNPM.java

```
package pertemuan_ke_04;
import java.util.Scanner;
public class cariNamaByNPM {
  private static StringNPM[] =
{"14.1.03.02.0009","14.1.03.02.0163","14.1.03.02.0180",
    "14.1.03.02.0188","14.1.03.02.0247","14.1.03.02.0249"},
    Nama[] = {"SUPRIADI","NURIN NAKMAH","DAVIT HARTONO",
"SONIA GORETTY","INDRA PRADANA","RYO RANGGA"};
public static void siapaNamanya(String npm){
boolean ada = false;
    for (int indeks = 0; indeks < NPM.length; indeks++) {
if(NPM[indeks].equals(npm)){
ada = true;
        System.out.println("Namanya : "+Nama[indeks]);
```

```
        break;
      }
    }
    if(!ada) System.out.println("NPM tidak ditemukan.");
}

public static void main(String[] args) {
String npm; Scanner input = new Scanner(System.in);
    System.out.print("Masukkan NPM yang akan dicari : ");
npm = input.next();
    siapaNamanya(npm);
}
}
```

cekDuplikat.java

```
package pertemuan_ke_04;
import java.util.Scanner;
public class cekDuplikat {
  private static int[] DATA;
public static boolean duplikat
(int bilangan, int lastIndeks){
    boolean ada = false;
    for (int i = 0; i < lastIndeks; i++)
if(DATA[i]==bilangan) { ada = true; break; }
      return ada;
}

  public static void main(String[] args) {
Scanner input = new Scanner(System.in);
    int banyakData, indeks, tmp;
    System.out.print("Banyak Data = ");
banyakData = input.nextInt();
    DATA = new int[banyakData];
    for (indeks = 0; indeks < banyakData; indeks++)
do {
System.out.print("DATA ke-"+(indeks+1)+" = ");
        tmp = input.nextInt();
        if(duplikat(tmp, indeks))
System.out.println("Data sudah ada.
Silakan masukkan lagi !");
        else DATA[indeks] = tmp;
      } while (duplikat(tmp, indeks));

System.out.println("Rekap DATA :");
    for (int i = 0; i < DATA.length; i++)
System.out.print(DATA[i]+", ");
System.out.println("\b\b");
```

```
}
}
```

konversiNilaiHuruf.java

```
package pertemuan_ke_04;
import java.util.Scanner;
public class konversiNilaiHuruf {
public static String nilaiHuruf(int nilai){
String huruf;
    if(nilai > 90) huruf = "A";
    else if(nilai > 80) huruf = "B+";
    else if(nilai > 70) huruf = "B";
    else if(nilai > 60) huruf = "C+";
    else if(nilai > 54) huruf = "C";
    else if(nilai > 38) huruf = "D";
    else huruf = "E";
    return huruf;
}

  public static void main(String[] args) {
String NPM[], Nama[]; int PBO[], banyakMhs, indeks;
Scanner input = new Scanner(System.in);

    System.out.print("Banyak Mahasiswa = ");
banyakMhs = input.nextInt();

    NPM = new String[banyakMhs];
    Nama = new String[banyakMhs];
PBO = new int[banyakMhs];

    for (indeks = 0; indeks < banyakMhs; indeks++) {
System.out.println("Data Mahasiswa ke-"+(indeks+1));
      System.out.print("NPM  : ");
NPM[indeks] = input.next();
      System.out.print("Nama : ");
Nama[indeks] = input.next();
      do {
System.out.print("Nilai PBO : ");
PBO[indeks] = input.nextInt();
        if(PBO[indeks] < 0 || PBO[indeks] > 100)
System.out.println("Nilai antara 0 - 100.
Silakan input ulang.");
}while(PBO[indeks] < 0 || PBO[indeks] > 100);
}
    System.out.println("Rekap data nilai Mahasiswa");
    System.out.println("NPM\tNama\tPBO\tHuruf");
    for (indeks = 0; indeks < banyakMhs; indeks++) {
System.out.println(NPM[indeks]+"\t"+Nama[indeks]+"\t"+
```

```
PBO[indeks]+"\t"+nilaiHuruf(PBO[indeks]));
}
}
}
```

pengurutanData.java

```
package pertemuan_ke_04;
public class pengurutanData {
private static int DATA[] = {95,60,65,75,90,85};

  public static void main(String[] args) {
System.out.println("DATA sebelum diurutkan :");
cetakDATA();

    for (int ind_1 = 0; ind_1 < DATA.length-1; ind_1++)
for (int ind_2 = ind_1+1; ind_2 < DATA.length; ind_2++)
tukar(ind_1, ind_2);

System.out.println("DATA setelah diurutkan :");
cetakDATA();
}

  public static void tukar(int indeks1, int indeks2){
if(DATA[indeks1] > DATA[indeks2]){
int tmp = DATA[indeks1];
      DATA[indeks1] = DATA[indeks2]; DATA[indeks2] = tmp;
}
}

  public static void cetakDATA(){
for (int indeks = 0; indeks < DATA.length; indeks++)
System.out.print(DATA[indeks]+", ");
    System.out.println("\b\b");
}
}
```

rekursifDuaPangkat.java

```
package pertemuan_ke_04;
public class rekursifDuaPangkat {
public static void main(String[] args) {
int pangkat = 5;
    System.out.println("2 pangkat "+pangkat+
" = "+duaPangkat(pangkat));
}

  public static int duaPangkat(int pangkat) {
if(pangkat == 0) return 1;
    else return 2*duaPangkat(pangkat-1);
```

```
}
}
```

4.9 Latihan

1. Buat Package baru pada "**Latihan_PBO_2018**" dengan nama "**latihan_04**".
2. Buat File Class baru untuk penerapan metode Fuzzy Tsukamoto, dengan nama **FuzzyTsukamoto.java.**
3. **FuzzyTsukamoto.java.** di atas nilai keanggotaannya didapat menggunakan fungsi linear, ubah menjadi fungsi sigmoid dengan persamaan :

$$NAIK = \begin{cases} 0 & ; crips \leq a \\ 2 \times \left(\frac{crips - a}{c - a}\right)^2 & ; a < crips < b \\ 0{,}5 & ; crips = b \\ 1 - 2 \times \left(\frac{c - crips}{c - a}\right)^2 & ; b < crips < c \\ 1 & ; crips \geq c \end{cases}$$

$$TURUN = \begin{cases} 1 & ; crips \leq a \\ 1 - 2 \times \left(\frac{crips - a}{c - a}\right)^2 & ; a < crips < b \\ 0{,}5 & ; crips = b \\ 2 \times \left(\frac{c - crips}{c - a}\right)^2 & ; b < crips < c \\ 0 & ; crips \geq c \end{cases}$$

Dimana :

a = nilai minimal

c = nilai maksimal

b = (a+c)/2

K_MeansClustering.java

```
package pertemuan_ke_04;
import java.util.Scanner;
public class FuzzyTsukamoto {
private static double NilaiKeanggotaanNaik(double min,
```

```
double max, double crips){
    double nilai = 0;
    if(crips >= max) nilai = 1;
    else if(crips <= min) nilai = 0;
    else nilai = (crips - min)/(max - min);
return nilai;
}

  private static double NilaiKeanggotaanTurun(double min,
double max, double crips){
    double nilai = 0;
    if(crips >= max) nilai = 0;
    else if(crips <= min) nilai = 1;
    else nilai = (max - crips)/(max - min);
    return nilai;
}

  private static double persediaanNaik(double min,
double max, double crips){
    return NilaiKeanggotaanNaik(min, max, crips);
}

  private static double persediaanTurun(double min,
double max, double crips){
    return NilaiKeanggotaanTurun(min, max, crips);
}

  private static double permintaanNaik(double min,
double max, double crips){
    return NilaiKeanggotaanNaik(min, max, crips);
}

  private static double permintaanTurun(double min,
double max, double crips){
    return NilaiKeanggotaanTurun(min, max, crips);
}

  private static double produksiNaik(double min,
double max, double alfa){
    return (alfa*(max - min))+min;
}

  private static double produksiTurun(double min,
double max, double alfa){
    return max-(alfa*(max - min));
}
```

```
  public static void main(String[] args) {
double minPersediaan = 800, maxPersediaan = 1500,
    minPermintaan = 500, maxPermintaan = 1200,
    minProduksi = 1000, maxProduksi = 2000,
persediaan, permintaan, produksi;
    Scanner input = new Scanner(System.in);
    System.out.println("Fuzzy Tsukamoto");
    System.out.println("Persediaan saat ini : ");
persediaan = input.nextDouble();
    System.out.println("Permintaan saat ini : ");
permintaan = input.nextDouble();

    //Fuzzifikasi
    double uPersediaanNaik, uPersediaanTurun,
uPermintaanNaik, uPermintaanTurun;
    uPersediaanNaik = persediaanNaik(minPersediaan,
maxPersediaan, persediaan);
    uPersediaanTurun = persediaanTurun(minPersediaan,
maxPersediaan, persediaan);
    uPermintaanNaik = permintaanNaik(minPermintaan,
maxPermintaan, permintaan);
    uPermintaanTurun = permintaanTurun(minPermintaan,
maxPermintaan, permintaan);

    //Pembentukan Rule & Mesin Inferensi
    double alpha[] = new double[4], z[] = new double[4];
/*Rule 1 : IF Persediaan turun AND permintaan turun THEN
produksi turun*/
    alpha[0] = Double.min(uPersediaanTurun, uPermintaanTurun);
    z[0] = produksiTurun(minProduksi, maxProduksi, alpha[0]);
/*Rule 2 : IF Persediaan turun AND permintaan naik THEN
produksi naik*/
    alpha[1] = Double.min(uPersediaanTurun, uPermintaanNaik);
    z[1] = produksiNaik(minProduksi, maxProduksi, alpha[1]);
    /*Rule 3 : IF Persediaan naik AND permintaan turun THEN
produksi turun*/
alpha[2] = Double.min(uPersediaanNaik, uPermintaanTurun);
    z[2] = produksiTurun(minProduksi, maxProduksi, alpha[2]);
    /*Rule 4 : IF Persediaan naik AND permintaan naik THEN
produksi naik*/
alpha[3] = Double.min(uPersediaanNaik, uPermintaanNaik);
    z[3] = produksiNaik(minProduksi, maxProduksi, alpha[3]);

    //defuzzifikasi
    double sumAlpha = 0, sumAlphaZ = 0;
    for (int indeks = 0; indeks < alpha.length; indeks++) {
```

```
sumAlpha = alpha[indeks];
sumAlphaZ = alpha[indeks] * z[indeks];
}


    produksi = sumAlphaZ/sumAlpha;
System.out.println("Produksi = "+(int)produksi);
}
}
```

BAB 5
Pemrograman Berorientasi Obyek

Desain berorientasi object adalah sebuah teknik yang memusatkan desain pada object dan class berdasarkan pada skenario dunia nyata. Hal ini menegaskan keadaan(state), behaviour dan interaksi dari object. Selain itu juga menyediakan manfaat akan kebebasan pengembangan, meningkatkan kualitas, mempermudah pemeliharaan, mempertinggi kemampuan dalam modifikasi dan meningkatkan penggunaan kembali software.

5.1 Class

Class merupakan blue print atau cetak biru dari sebuah objek. Sebuah Class dapat menggambarkan ciri dari sebuah objek secara umum. Sebagai contoh Toyota Yaris, Honda Jazz, Honda Brio, dan Suzuki Swift merupakan objek dari Class mobil. Toyota Yaris dan objek lainnya juga memiliki kesamaan atribut (merk, tipe, berat, kapasitas bensin, tipe mesin, warna, harga) dan method untuk mengakses data pada atributnya (misal fungsi untuk menginputkan data merk, tipe, berat, dsb serta fungsi untuk mencetak data merk, tipe, berat, dsb). **Mobil.java** adalah contoh pembuatan class.

Mobil.java

```
package pertemuan_ke_05;
public class Mobil {
private String merk, tipe;
  private int tangki;
  private long harga;
```

```
  public Mobil() {
merk = "";  tipe = ""; tangki = 0; harga = 0;
  }

  public Mobil(String merk, String tipe,
int tangki, long harga) {
    this.merk = merk; this.tipe = tipe;
this.tangki = tangki; this.harga = harga;
}

  public String getMerk() { return merk; }
  public String getTipe() { return tipe; }
public int getTangki() { return tangki; }
  public long getHarga() { return harga; }
  public void setMerk(String merk) { this.merk = merk; }
  public void setTipe(String tipe) { this.tipe = tipe; }
  public void setTangki(int tangki) { this.tangki = tangki; }
  public void setHarga(long harga) { this.harga = harga; }
}
```

5.2 Instant Of Class

a. Object

Objek merupakan segala sesuatu yang ada didunia ini dan memiliki bentuk fisik seperti manusia, hewan, tumbuhan, rumah, kendaraan, dan lain sebagainya. Di dalam pemrograman berorientasi objek, kita akan belajar tentang konsep objek dalam kehidupan nyata menjadi objek dalam dunia pemrograman.

Setiap objek dalam dunia nyata memiliki 2 elemen penyusunnya, yaitu keadaan (state) dan perilaku/sifat (behaviour). Saat sebuah objek diterjemahkan ke dalam konsep PBO, maka elemen penyusunnya juga terdiri atas 2 bagian, yaitu :

- Atribut, adalah ciri-ciri yang melekat pada suatu objek (*state*).
- Method, adalah fungsi-fungsi yang digunakan untuk memanipulasi nilai-nilai pada atribut untuk melakukan hal-hal yang dapat dilakukan suatu objek (*behaviour*).

Sintaks membuat objek dari suatu class :

```
namaClass namaObjek = new namaClass();
```

Class untuk memanggil class Mobil, **panggilMobil.java**. Dimana nama obyek (instant of class) dari Mobil adalah instanMobil.

panggilMobil.java

```
package pertemuan_ke_05;
```

```
public class panggilMobil {
public static void main(String[] args) {
Mobil instanMobil = new Mobil("Toyota Fortuner",
"4x2 2.4 G M/T Diesel", 80, 471850000);
    System.out.println("Merk   : "+instanMobil.getMerk());
    System.out.println("Tipe   : "+instanMobil.getTipe());
    System.out.println("Tangki : "+instanMobil.getTangki()+
" Liter");
System.out.println("Harga  : Rp. "+
instanMobil.getHarga());
}
}
```

b. Anonymous Object

Anonymous object merupakan objek yang tidak memiliki nama. Tipe object ini tidak akan memakan resource memory. Namun kelemahannya dari object ini tidak dapat digunakan lagi (hanya dapat digunakan satu kali saja) karena setelah digunakan akan langsung dihapus. Contoh :

```
System.out.println("Merk   :   "+   new   Mobil("Toyota
Fortuner",
"4x2 2.4 G M/T Diesel", 80, 471850000).getMerk());
```

5.3 Keyword "this"

Keyword this digunakan untuk membedakan variabel merek pada parameter dan atribut kelas (contoh class Mobil), atau dengan kata lain untuk membedakan variabel global dan lokal, this digunakan untuk merujuk variabel global.

5.4 Atribut & Method

Pada file Mobil.java, atribut dari class Mobil adalah merk, tipe, tangki, dan harga. Sedangkan methodnya adalah setMerk(String merk), setTipe(String tipe), setTangki(int tangki), setHarga(long harga), getMerk(), getTipe(), getTangki() dan getHarga().

Secara umum satu attribut pasti mempunyai hubungan dengan 2 method dalam satu class, yaitu getter dan setter. Getter adalah method untuk mendapatkan nilai dari attribut, Setter adalah method untuk memberikan nilai pada attribut tersebut. Misal : Attribut Merk berhubungan dengan getter getMerk() dan Setter setMerk(String Merk). panggilMobilLain.java adalah contoh pemanfaatan setter dan getter.

panggilMobilLain.java

```
package pertemuan_ke_05;
public class panggilMobilLain {
public static void main(String[] args) {
Mobil instanMobil =  new Mobil();
    instanMobil.setMerk("Toyota Innova");
    instanMobil.setTipe("All New V Diesel");
    instanMobil.setTangki(55);
    instanMobil.setHarga(375000000);

    System.out.println("Merk   : "+instanMobil.getMerk());
System.out.println("Tipe   : "+instanMobil.getTipe());
    System.out.println("Tangki : "+instanMobil.getTangki()+
" Liter");
    System.out.println("Harga  : Rp. "+
instanMobil.getHarga());
}
}
```

5.5 Access Modifier

Pada saat membuat, mengatur *properties* dan *class methods*, kita ingin untukmengimplementasikan beberapa macam larangan untuk mengakses data. Sebagai contoh,jika Anda ingin beberapa atribut hanya dapat diubah hanya dengan *method* tertentu, tentuAnda ingin menyembunyikannya dari *object* lain pada *class*. Di JAVA, implementasi tersebutdisebut dengan ***access modifiers.***Terdapat 4 macam *access modifiers* di JAVA, yaitu : *public, private, protected* dan *default*. 3tipe akses pertama tertulis secara ekplisit pada kode untuk mengindikasikan tipe akses,sedangkan yang keempat yang merupakan tipe *default*, tidak diperlukan penulisan *keyword*atas tipe.

5.6 Constructor

Constructor merupakan sebuah method yang secara otomatis dipanggil atau dijalankan pada saat sebuah class diinstansi. Java akan membuatkan sebuah default constructor jika di dalam sebuah kelas tidak memiliki constructor. Nama constructor harus sama dengan nama class dan tidak memiliki tipe return value. Sama halnya dengan method, constructor dapat memiliki satu atau banyak parameter maupun tanpa parameter.

Multiple constructor adalah adanya lebih dari satu contructor untuk sebuah class. Yang membedakan antara satu constructor dengan constructor lainnya adalah pada parameternya (nama constructornya sama). Seperti

Mobil.java terdapat constructor Mobil() dan constructor Mobil(String merk, String tipe, int tangki, long harga).

5.7 Information Hiding dan Encapsulation

Information Hiding adalah menyembunyikan attribute dan method suatu objek dari objek lain. Encapsulation adalah menyembunyikan atribute suatu objek dari objek lain. Attribute maupun method disembunyikan dengan cara memberikan modifier private.

5.8 Static Variable vs Instant Variable

Variabel static merupakan variabel yang dideklarasikan menggunakan keyword “static”, sedangkan variabel yang dideklarasikan dengan tidak menggunakan keyword “static” atau sering disebut dengan istilah instant variabel (atau variabel instant).

a. Jika sebuah variable merupakan variable instant, maka masing-masing objek dari class tersebut akan memiliki variable yang sama dengan variable instant tersebut, perubahan nilai yang terjadi pada variable instant di satu objek tidak akan berpengaruh pada variable instant di objek yang berbeda.
b. Jika sebuah variable merupakan variable static (pada suatu class), maka variabel static tersebut adalah variabel yang sama di semua objek dari class tersebut. Sehingga perubahan nilai pada variabel static tersebut di suatu objek akan berpengaruh juga terhadap objek yang lainnya.
c. Nilai suatu variabel static akan selalu sama untuk semua instant of class (atau objek) dari sebuah class.
d. Sebuah variable dideklarasikan static apabila variable tersebut bersifat global bagi semua objek (instant of class) dari suatu class.Contohnya adalah variable yang menyimpan nilai jumlah objek yang telah dibuat.

5.9 Inheritance

Inheritance merupakan proses pewarisan data dan method dari sebuah class yang telah ada kepada suatu class baru. Class yang mewariskan disebut dengan superclass / parent class / base class, sedangkan class yang mewarisi (class yang baru) disebut dengan subclass / child class / derived class. Subclass tidak dapat mewarisi anggota private dari superclass-nya. Dengan inheritance, class yang baru (subclass) akan mirip dengan class yang lama (superclass) namun memiliki karakteristik yang baru. Dalam

Java, subclass hanya bisa memiliki satu superclass (single inheritance) sedangkan superclass bisa memiliki satu subclass atau lebih. Untuk menerapkan inheritance, gunakan statement "extends".

```
namaSubclass extends namaSuperclass {
..... // definisi class
}
```

Keyword "super" digunakan oleh subclass untuk memanggil constructor atau method yang ada pada superclass-nya. Contoh untuk memanggil constructor milik superclass-nya :

```
super();
super(parameter);
```

Contoh untuk memanggil method milik superclass-nya :

```
super.namaMethod(parameter);
```

5.10 Polymorphism Method Overriding

Polymorphism mempunyai makna sesuatu yang memiliki banyak bentuk, yaitu memiliki nama sama, tetapi memiliki kelakuan (behaviour) yang berbeda.

Overriding method adalah kemampuan dari subclass untuk memodifikasi method dari superclass-nya, yaitu dengan cara menumpuk (mendefinisikan kembali) method superclass-nya.

5.11 Abstract Class

Abstract class adalah class yang terletak pada posisi tertinggi dari hierarki class dan digunakan sebagai acuan (superclass) bagi penurunan class-class lainnya. Biasanya class ini hanya berisi variable-variabel umum (atribut umum) dan header-header method saja tanpa isi methodnya atau bisa juga sebagian method ada isi methodnya dan sebagian method yang lain tidak memiliki body method. Oleh karena itu, class-class turunannyalah yang mendefinisikan secara detil isi method-method tersebut. Cara untuk membuat sebuah class abstrak adalah :

```
akses_modifier abstract class namaClassAbstrak {
..... // definisi class
}
```

Java memiliki aturan-aturan dalam penggunaan method abstrak dan class abstrak sebagai berikut :

a. Class yang di dalamnya terdapat abstract method harus dideklarasikan sebagai abstract class.
b. Abstract class tidak dapat diinstansi, tetapi harus di turunkan. Abstract class tidak dapat diinstansi (menjadi objek dari class abstract), tetapi kita dapat mendeklarasikan suatu variable yang bertipe abstract class dan membuat instansi dari variabel tersebut yang bertipe class turunan dari abstract class tersebut (teknik polymorphism).
c. Sebuah class dapat dideklarasikan sebagai abstract class meskipun class tersebut tidak memiliki abstract method.
d. Abstract method tidak boleh mempunyai body method dan demikian juga sebaliknya bahwa method yang tidak ditulis isi methodnya maka harus dideklarasikan sebagai abstract method.

5.12 Interface

Interface merupakan sekumpulan konstanta atau deklarasi method tanpa menyertakan/menuliskan isi methodnya. Interface biasa digunakan untuk mendeklarasikan koleksi method dan konstanta yang dapat digunakan oleh satu atau lebih class. Contoh syntax deklarasi interface :

```
interface namaInterface
{
......... //deklarasi
konstanta dan method
.........
}
```

Class mengimplementasikan sebuah interface yang telah ada dengan menggunakan kata kunciimplements. Class ini dibuat untuk mengimplementasikan semua method interface. Sebuah classboleh mengimplementasikan lebih dari satu interface.

5.13 Package

Package adalah sarana/cara pengelompokkan dan pengorganisasian kelas-kelas dan interface yang sekelompok menjadi suatu unit tunggal dalam library. Secara fisik package berupa folder yang berisi file-file class yang berfungsi khusus. Package juga mempengaruhi mekanisme hak akses

ke kelas-kelas di dalamnya. Deklarasi package harus diletakkan pada bagian paling awal (sebelum deklarasi import) dari source code setiap kelas yang dibungkus package tersebut, hal ini telah kita lakukan sejak awal, pada setiap awal file source code.

Jika kita membutuhkan interface, method dan atau attribut dari suatu kelas yang ada pada package lain dalam satu project, maka kita dapat memanggilnya dengan mengimport nama kelas tempat method atau attribut didefinisikan.

5.14 Catatan Penting

a. File source code tidak wajib memiliki method main.
b. File yang dapat dijalankan (Run) adalah file class yang memiliki method main.
c. Perbedaan antara abstract class & interface :
    1. Abstract class dapat memiliki tipe data dengan semua tipe data, sedangkan interface tipe datanya hanya berupa konstanta
    2. Method pada abstract class dapat berupa method kongkrit dan method abstract. Sedangkan pada interface semua method harus berupa method abstract.

5.15 Latihan

1. Buat Package baru pada "**Latihan_PBO_2018**" dengan nama "**latihan_05**".
2. Buat file baru Mahasiswa.java, interfaceSorting.java, BubbleSort.java, SelectionSort.java dan pengurutanDataMahasiswa.java
3. Run file pengurutanDataMahasiswa.java
4. Modifikasi program supaya menu "Pengurutan Berdasarkan Nama" dapat berfungsi.
5. Tambahkan menu "Pilih Urutan Tampilan" dimana pengguna diminta memilih tampilan Ascending atau Descending. Kemudian modifikasi program supaya pilihan tersebut berfungsi dengan baik.

Mahasiswa.java

```
package latihan_05;
public class Mahasiswa {
private String NPM, Nama;

  public Mahasiswa(String NPM, String Nama) {
```

```
this.NPM = NPM; this.Nama = Nama;
}
  public String getNPM() { return NPM; }
  public void setNPM(String NPM) { this.NPM = NPM; }
  public String getNama() { return Nama; }
  public void setNama(String Nama) { this.Nama = Nama; }
  public int bandingkanNPM(Mahasiswa mhs){
int hasil = NPM.compareTo(mhs.getNPM());
    if(hasil < 0) return -1;
    else if(hasil > 0) return 1;
    else return 0;
}

  public int bandingkanNama(Mahasiswa mhs){
int hasil = Nama.compareTo(mhs.getNama());
    if(hasil < 0) return -1;
    else if(hasil > 0) return 1;
    else return 0;
}
}
```

interfaceSorting.java

```
package latihan_05;
public interface interfaceSorting {
  public void sortByNPM();
  public void sortByNama();
  public void tukar(int indeks1, int indeks2);
}
```

BubbleSort.java

```
package latihan_05;
public class BubbleSort implements interfaceSorting{
  private Mahasiswa DataMahasiswa[];
  private int banyakData;

public BubbleSort(Mahasiswa[] DataMahasiswa) {
this.DataMahasiswa = DataMahasiswa;
banyakData = DataMahasiswa.length;
}

  public void sortByNPM() {
for (int ind1 = 0; ind1 < banyakData-1; ind1++)
for (int ind2 = ind1+1; ind2 < banyakData; ind2++)
if(DataMahasiswa[ind1].bandingkanNPM(
DataMahasiswa[ind2]) > 0) tukar(ind1, ind2);
    }

public void sortByNama(){ /*definisikan isinya*/}
```

```
  public void tukar(int indeks1, int indeks2) {
Mahasiswa mhs = DataMahasiswa[indeks1];
    DataMahasiswa[indeks1] = DataMahasiswa[indeks2];
    DataMahasiswa[indeks2] = mhs;
}

public Mahasiswa[] getDataMahasiswa() {
return DataMahasiswa;
}
}
```

SelectionSort.java

```
package latihan_05;
public class SelectionSort implements interfaceSorting {
private Mahasiswa DataMahasiswa[];
  private int banyakData;

  public SelectionSort(Mahasiswa[] DataMahasiswa) {
this.DataMahasiswa = DataMahasiswa;
banyakData = DataMahasiswa.length;
}

public void sortByNPM() {
int indMinimal;
    for (int ind1 = 0; ind1 < banyakData-1; ind1++) {
indMinimal = ind1;
      for (int ind2 = ind1+1; ind2 < banyakData; ind2++) {
if(DataMahasiswa[indMinimal].bandingkanNPM(
DataMahasiswa[ind2]) > 0) indMinimal = ind2;
}
      if(indMinimal != ind1) tukar(ind1, indMinimal);
}
}

public void sortByNama(){ /*definisikan isinya*/}
  public void tukar(int indeks1, int indeks2) {
Mahasiswa mhs = DataMahasiswa[indeks1];
    DataMahasiswa[indeks1] = DataMahasiswa[indeks2];
    DataMahasiswa[indeks2] = mhs;
}

  public Mahasiswa[] getDataMahasiswa() {
return DataMahasiswa;
}
}
```

pengurutanDataMahasiswa.java

```
package latihan_05;
import java.util.Scanner;
```

```
public class pengurutanDataMahasiswa {
private Scanner input = new Scanner(System.in);
  private int banyakData = -1, metodePengurutan = 1;
  private Mahasiswa[] DATA;
  private boolean adaData = false;

public pengurutanDataMahasiswa() {
int pilihan;
    do {
cetakMenu();
pilihan = input.nextInt();
aksi(pilihan);
} while (pilihan > 0 && pilihan < 7);
}

  public void cetakMenu(){
System.out.println("============================");
    System.out.print("Program Pengurutan Mahasiswa.\n"
+ "1. Input Banyak Mahasiswa\n"
+ "2. Input Data Mahasiswa\n"
      + "3. Pilih Metode Pengurutan\n"
+ "4. Pengurutan Berdasarkan NPM\n"
+ "5. Pengurutan Berdasarkan Nama\n"
+ "6. Cetak data Mahasiswa\n"
+ "lainnya untuk keluar\nMasukkan pilihan Anda :");
}

  public void aksi(int pilihan){
if(pilihan != 1 && banyakData < 0) {
System.out.println("Tentukan banyak data terlebih
dahulu.");
} else switch(pilihan){
case 1 :
if(banyakData < 0) setBanyakMahasiswa();
        else System.out.println("Banyak Mahasiswa tidak dapat
diganti.");
        break;
      case 2 : inputDataMahasiswa(); break;
      case 3 : setMetodePengurutan(); break;
      case 4 : pengurutanByNPM(); break;
      case 5 : pengurutanByNama(); break;
      case 6 : cetakData(); break;
      default:
System.out.println("Terima kasih telah menggunakan
Program ini.");
}
```

```
}

  public void setBanyakMahasiswa(){
System.out.print("Banyak Mahasiswa : ");
    banyakData = input.nextInt();
DATA = new Mahasiswa[banyakData];
}

  public void inputDataMahasiswa(){
String npm, nama;
    System.out.println("**********************");
    System.out.println("Input Data Mahasiswa.");
    for (int indeks = 0; indeks < banyakData; indeks++) {
System.out.println("Mahasiswa ke-"+(indeks+1)+
" dari "+banyakData);
      System.out.print("NPM  : "); npm = input.next();
      System.out.print("Nama : "); nama = input.next();
      DATA[indeks] = new Mahasiswa(npm, nama);
}
    adaData = true;
}

  public void setMetodePengurutan(){
System.out.print("Metode Pengurutan :\n"
+ "1. Bubble Sort.\n"
      + "2. Selection Sort.\n"
+ "Pilihan : ");
metodePengurutan = input.nextInt();
}

  public void pengurutanByNPM(){
if(!adaData){
System.out.println("Input data terlebih dahulu");
} else {
      System.out.print("Pengurutan Data Mahasiswa "
+ "berdasarkan NPM dengan metode");
if(metodePengurutan == 1){
BubbleSort bubbleSort = new BubbleSort(DATA);
        bubbleSort.sortByNPM();
DATA = bubbleSort.getDataMahasiswa();
        System.out.print(" Bubble Sort ");
      } else {
SelectionSort selectionSort = new SelectionSort(DATA);
selectionSort.sortByNPM();
DATA = selectionSort.getDataMahasiswa();
        System.out.print(" Selection Sort ");
}
```

```
      System.out.println("berhasil.");
}
}

public void pengurutanByNama(){ /*definisikan isinya*/}
public void cetakData(){
if(!adaData){
System.out.println("Input data terlebih dahulu");
} else {
System.out.println("Data Mahasiswa.\nNPM\t\tNama");
       for (int indeks = 0; indeks < banyakData; indeks++)
System.out.println(DATA[indeks].getNPM()+
"\t"+DATA[indeks].getNama());
}
}

  public static void main(String[] args) {
new pengurutanDataMahasiswa();
}
}
```

BAB 6
Exception Handling, Java Util, Operasi File, Penanganan Waktu, Java Math

6.1 Exception Handling

Error (kesalahan) dalam pemrograman dibagi dalam tiga katagori yaitu *syntax error* (muncul saat kompilasi), *run time error*, dan *logic error* (ketika output belum sesuai dengan yang diharapkan).

Exception digunakan sebagai sarana untuk melaporkan jenis kondisi kesalahan saat program dijalankan, dan mengendalikannya agar *run time error* tersebut tidak mengakibatkan eksekusi dihentikan (statement setelahnya tetap dieksekusi).

Dalam java, exception merupakan objek dari subkelas yang diturunkan dari kelas *Throwable*. Kelas *Throwable* ini terdapat dalam package *java.lang.object*. Kelas ini mengandung dua sub kelas berikut:

a. Kelompok Kelas Error
   Error ini bersifat fatal sehingga sistem tidak dapat dimanipulasi, contoh kelas: *LinkageError*, *VirtualMachineError*, dan *AWTError*.
b. Kelompok Kelas Exception
   Jenis error ini masih dapat diantisipasi dengan menyisipkan statement tambahan untuk mendeteksi statement penyebab dan jenisnya.

Ada kelompok *RuntimeException* yang diperiksa oleh interpreter, apakah akan ditangani atau dilempar, namun ada pula exception yang tidak

diperiksa interpreter. Disamping itu programmer dibolehkan membuat exception sendiri dengan cara *extends* atau *implements* kelas *Exception*.

Diperlukan tiga langkah berikut ini untuk mengantisipasi exception :

a. Mendeklarasikan Exception :

```
[modifier] returntype namaMethod() throws
tipeException{ ... }
```

b. Melempar Exception :

```
TipeException namaObjek = new TipeException();
throw namaObjek;
```

atau

```
throw namaObjek TipeException();
```

atau

```
throw new TipeException();
```

c. Menangkap Exception :

```
try {
// blok statement yg mungkin menghasilkan exception
}
catch(TipeException1 namaObjek) {
// penanganan jenis exceptionTipeException1
}
catch(TipeException2 namaObjek) {
// penanganan jenis exception TipeException2
}
catch(TipeExceptionN namaObjek) {
// penanganan jenis exception TipeExceptionN
}
finally {
// blok yang harus dieksekusi
}
```

Jika pada blok try tidak terjadi exception, maka blok catch tidak ada yang dieksekusi dan segera blok finally yang dieksekusi.

Jika terjadi exception pada blok try, maka salah satu blok catch dieksekusi, kemudian blok finally dieksekusi.

6.2 Array Dinamis

Kelas Vector adalah kelas yang memungkinkan pengimplementasian array yang ukurannya dapat diubah sewaktu-waktru (dinamis). Kelas ArrayList adalah suatu senarai (list) yang memungkinkan pembuatan obyek array yang ukurannya dapat diubah sewaktu. Secara default tipe data dari elemen Vector atau ArrayList adalah kelas Object, jika kita ingin

menspesifikkan tipe data elemennya maka saat pendeklarasian obyek kelas Vector ataupun ArrayList kita bisa menggunakan cara berikut :

```
    Vector<tipe Data> objVector = new Vector<>();
    ArrayList<tipe Data> objArrayList = new
ArrayList<>();
```

6.3 Tokenizing

Tokenizing adalah proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi token-token/bagian-bagian tertentu.

Kelas StringTokenizer dipakai untuk membagi string menjadi potongan-potongan sehingga informasi yang terkandung dapat diterima kembali dan diproses. Kelas StringTokenizer mengenali setiap kata dengan menentukan sekumpulan karakter sebagai delimiter / pembatas ketika membentuk sebuah object StringTokenizer. Delimiter yang akan membagi sebuah string menjadi potongan-potongan yang disebut tokens.

Method String.split() memiliki kegunaan yang sama dengan Kelas StringTokenizer akan tetapi hasil dari method split berupa array String.

6.4 Operasi File

Dengan operasi file, data yang digunakan dalam aplikasi bisa disimpan secara permanen di hard disk. Sehingga data tidak akan hilang ketika aplikasi di-close atau komputer dimatikan. Operasi file juga memungkinkan penyimpanan data dengan ukuran yang lebih besar.

Untuk membaca sebuah file, kita menggunakan class FileInputStream. Untuk menuliskan sebuah file, Anda dapat menggunakan class FileOutputStream.

OperasiFile.java adalah file bantu untuk menulis file dan memuat file. Sebelum menggunakannya pastikan file eksternal yang akan dimuat atau ditulis telah ada. Misal pada file penyimpananDinamisVector.java file eksternal yang kita butuhkan beralamat "D:\DATA\DataMahasiswa.txt".

operasiFile.java

```
package pertemuan_ke_06;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Vector;
```

```
public class operasiFile {
private Vector<String> isi;
  private String alamatFile, tampung;
  private FileWriter simpan;
  private FileReader muat;
  private BufferedWriter penulis;
  private BufferedReader pembaca;

public operasiFile(String alamatFile) {
setAlamatFile(alamatFile);
    try {
muat = new FileReader(alamatFile);
    } catch (Exception ex) {
System.out.println("File "+alamatFile+
" tidak ditemukan");
System.exit(0);
}
}

  public void muatFile(){
pembaca = new BufferedReader(muat);
    isi = new Vector<>();
    do {
try {
tampung=pembaca.readLine();
        if(tampung == null) break;
        isi.add(tampung);
      } catch (IOException ex) {
System.out.println("Kesalahan saat pembacaan isi
file");
}
} while (true);

    try {
pembaca.close();
} catch (IOException ex) {
System.out.println("Error : "+ex.getMessage());
}
}

  public void simpanFile(){
try {
sipan = new FileWriter(alamatFile);
} catch (IOException ex) {
System.out.println("File "+alamatFile+" tidak
ditemukan");
}
    penulis = new BufferedWriter(simpan);
    for (int i = 0; i < isi.size(); i++) {
try {
penulis.write(isi.elementAt(i));
        if(i < isi.size()-1) penulis.newLine();
} catch (IOException ex) {
System.out.println("Kesalahan saat penulisan isi
```

```
file");
}
    }
    try {
penulis.close();
} catch (IOException ex) {
System.out.println("Error : "+ex.getMessage());
}
}

  public Vector<String> getIsi() {return isi;}
  public void setIsi(Vector<String> isi) {this.isi = isi;}
  public String getAlamatFile() {return alamatFile;}
public void setAlamatFile(String alamatFile) {
this.alamatFile = alamatFile;
}
}
```

penyimpananDinamisVector.java (setelah modif)

```
package pertemuan_ke_06;
import java.util.Vector;
public class penyimpananDinamisVector {
private Vector<Mahasiswa> vMhs = new Vector<>();
private operasiFile opf;
  public penyimpananDinamisVector() {
muatDataMahasiswa();
    vMhs.add(new Mahasiswa("14.1.03.02.0009", "SUPRIADI"));
    vMhs.add(new Mahasiswa("14.1.03.02.0163",
"NURIN NAKMAH"));
    vMhs.add(new Mahasiswa("14.1.03.02.0180",
"DAVIT HARTONO"));
    System.out.println("Cetak 1 :"); cetak();
    vMhs.insertElementAt(new Mahasiswa("14.1.03.02.0188",
"SONIA GORETTY"), 1);
    System.out.println("Cetak 2 :"); cetak();
    vMhs.remove(0);
    System.out.println("Cetak 3 :"); cetak();
    vMhs.add(new Mahasiswa("14.1.03.02.0247",
"INDRA PRADANA"));
    System.out.println("Cetak 4 :"); cetak();
    simpanDataMahasiswa();
}

  private void cetak(){
for (int i = 0; i < vMhs.size(); i++)
System.out.println(vMhs.elementAt(i));
}

  public static void main(String[] args) {
new penyimpananDinamisVector();
}

  private void muatDataMahasiswa(){
opf = new operasiFile("D:/DATA/DataMahasiswa.txt");
```

```
    opf.muatFile();
    Vector<String> isi = opf.getIsi();
    if(isi.size() > 0){
for (int i = 0; i < isi.size(); i++) {
String[] elementAt = isi.elementAt(i).split(";");
        vMhs.add(new Mahasiswa(elementAt[0], elementAt[1]));
      }
    }
}

  private void simpanDataMahasiswa(){
Vector<String> isi = new Vector<>();
    for (int i = 0; i < vMhs.size(); i++) {
Mahasiswa mhs = vMhs.elementAt(i);
isi.add(mhs.getNPM()+";"+mhs.getNama());
}
    opf.setIsi(isi);
    opf.simpanFile();
}
}
```

6.5 Fungsi Matematika

Math adalah kelas yang terdapat pada paket bawaan Java yang berguna untuk melakukan berbagai perasi matematika sperti sinus, cosinus,akar kuadrat, logaritma dan sebagainya. Berikut beberapa method dalam kelas Math :

Fungsi	Keterangan
Sudut	
sin (double a)	Menghasilkan sinus sudut a dalam radian
cos (double a)	Menghasilkan cosinus sudut a dalam radian
tan (double a)	Menghasilkan tangent sudut a dalam radian
asin (double r)	Menghasilkan sudut yang sinusnya r
acos (double r)	Menghasilkan sudut yang cosinusnya r
atan (double r)	Menghasilkan sudut yang tangentnya r
atan2 (double a, double b)	Menghasilkan sudut yang tangentnya a / b
Exponensial	
pow(double y, double x)	Menghasilkan y pangkat x ; contoh pow(2,3) = 8
exp (double x)	Menghasilkan (exponen) e pangkat x
log (double x)	Menghasilkan logaritma natural dari x
sqrt (double x)	Menghasilkan akar kuadrat x
Pembulatan	
ceil (double a)	Menghasilkan pembulatan keatas , Contoh : 3.12 menjadi 4, 3.67 menjadi 4, 3.988 menjadi 4

floor (double a)	Menghasilkan pembulatan kebawah , Contoh : 3.12 menjadi 3, 3.67 menjadi 3, 3.988 menjadi 3
rint (double a)	Menghasilkan besaran double dari a yang dipotong
round (float a)	Menghasilkan pembulatan keatas ke int terdekat
round (double a)	Menghasilkan pembulatan keatas ke long terdekat
Agregat	
abs(a)	Menghasilkan nilai absolut dari a
max(a,b)	Menghasilkan nilai maximum dari a dan b
min(a,b)	Menghasilkan nilai minimum dari a dan b
Random	
random()	Menghasilkan bilangan random/acak
PI	
PI	Menghasilkan nilai PI

6.6 Penanganan Waktu

Java menyediakan kelas bawaan untuk menangani waktu yaitu, kelas Date, SimpleDateFormat, Calendar dan GregorianCalendar.

Kelas Date adalah kelas yang berhubungan dengan penanganan informasi tanggal dan jam. SimpleDateFormat adalah kelas yang berhubungan dengan format tampilan waktu sesuai kebutuhan pemrogram. Kelas Calendar adalah kelas abstrak yang digunakan untuk mengeset tanggal ataupun mendapatkan tanggal. Kelas GregorianCalendar adalah kelas turunan dari kelas Calendar yang ditujukan untuk menangani pemrosesan tanggal dan jam.

6.7 Latihan

Buat program untuk mengolah dan menampilikan data Mahasiswa, Mata Kuliah dan nilainya, dimana terdapat 3 file tempat data disimpan, yaitu

1. Mahasiswa.txt yang berisi NPM dan Nama Mahasiswa.
2. Mata Kuliah.txt yang berisi Kode Mata Kuliah, Nama Mata Kuliah dan SKSnya.
3. Nilai.txt yang berisi NPM, Kode Matakuliah dan nilai angka

Program tersebut memiliki menu untuk :

1. Input Mahasiswa Baru (NPM & Nama).
2. Input Nilai (per Mahasiswa per Mata Kuliah).
3. Cetak info Mata Kuliah (tampilkan kode, nama & SKS semua Mata Kuliah)
4. Cetak Nilai per Mata Kuliah (input : kode Mata Kuliah, output : NPM, Nama, Nilai Angka, Nilai Huruf).
5. Cetak Transkrip Mahasiswa (input : NPM, output : kode mata kuliah, nama mata kuliah, SKS, nilai huruf, bobot*sks dan nilai IPK)
6. Cetak IPK semua Mahasiswa (NPM, Nama, IPK).

Mahasiswa.txt

```
17.1.03.02.0099#Mifta Azizi
17.1.03.02.0060#Hari Satiawan
17.1.03.02.0015#Febry Randawan
17.1.03.02.0080#Ratna Nugraheni
17.1.03.02.0102#Yogik Se'bianto
17.1.03.02.0121#Fitria Nurlaili
17.1.03.02.0014#Andri Nur Hamzah
17.1.03.02.0017#Qoni` Abdul Wahid
17.1.03.02.0023#Haris Riza Valevi
17.1.03.02.0106#Cholilul Rosyidin
```

Mata Kuliah.txt

```
MKK2001#Algoritma Pemrograman I#3
MKK2003#Bahasa Inggris#2
MKK2007#Kalkulus #4
MKK2011#Logika Matematika#3
MPK1005#Pendidikan Agama #3
MKK2016#Pengantar Teknologi Informasi#3
```

Nilai.txt

```
17.1.03.02.0099#MKK2001#75
17.1.03.02.0060#MKK2001#65
17.1.03.02.0015#MKK2001#55
17.1.03.02.0099#MKK2003#80
17.1.03.02.0060#MKK2003#95
17.1.03.02.0015#MKK2003#85
17.1.03.02.0099#MKK2007#70
17.1.03.02.0060#MKK2007#90
17.1.03.02.0015#MKK2007#90
```

Konversi nilai Angka ke huruf

```
91 s.d 100 : A
81 s.d 90  : B+
71 s.d 80  : B
61 s.d 70  : C+
```

```
55 s.d 60  : C
39 s.d 54  : D
0  s.d 38  : E
```

BAB 7
Graphical User Interface

7.1 Graphical User Interface

Graphical User Interface (GUI) merupakan desain aplikasi dengan tampilan visual sehingga pengguna dapat dengan mudah menggunakan aplikasi. AWT dan Swing menyediakan komponen GUI yang dapat digunakan dalam membuat aplikasi Java dan applet. Nama dari komponen GUI milik Swing hampir sama persis dengan komponen GUI milik AWT. Sebagai contoh, komponen AWT, Button class. Pada Swing, nama komponen tersebut menjadi JButton class.

Berikut ini adalah daftar dari beberapa kelas penting pada kontainer yang telah disediakan oleh AWT.

- Component
- Container
- Panel
- Window
- Frame

Komponen GUI pada Swing terdapat dalam paket javax.swing. Berikut adalah daftar dari beberapa komponen Swing:

- JComponent
- JFrame
- JPanel
- JApplet
- JButton
- JLabel
- JTextField
- JtextArea
- JCheckBox
- JRadioButton

- JComboBox
- JFileChooser
- JColorChooser
- JTable
- JScrollPane
- JMenu
- JOptionPane
- JDialog
- JProgressBar
- JSlider
- JTabbedPane
- JDesktopPane
- JInternalFrame

7.2 Event Listener dan Event Handler

Untuk mendeteksi apa yang dilakukan user terhadap komponen-komponen tersebut dan menentukan prosesnya masing-masing, Anda perlu mempelajari bagaimana **event listener** dan **event handler** bekerja. **Event** adalah peristiwa yang di-*stimulasi* / di-*trigger* oleh user terhadap komponen-komponen GUI. Bila anda meng-klik suatu objek dari **Jbutton**, mengetikkan teks pada objek **TextField**, atau menentukan suatu pilihan dari objek **JComboBox**, maka event akan tercipta. Event ini kemudian akan "ditangkap" *event listener* karena setiap komponen Anda beri ID (identitas) seperti button1, button2, textField1, comboBox1, dan masing-masing anda tambahkan *event listener*, maka Java akan dapat mengenali komponen mana yang menstimulasi *event.*

Selanjutnya, Anda harus menentukan **event handler**, yaitu blok yang akan memproses bila terjadi suatu event. *Event handler* dapat anda analogikan seperti halnya fungsi pada tombol-tombol operasi x, /, +, -, = pada kalkulator. Bila Anda ingin mengalikan dua bilangan, maka tombol "x" ditekan, bila akan menjumlah bilangan, tombol "+" yang dipilih, dan sebagainya. Setiap tombol memiliki fungsi yang berbeda dan sebagai konsekuensinya, hasil pemrosesan juga akan berbeda.

Event listener dan Event handler ini terdapat pada package **java.awt.event,** dan memiliki bentuk sebagai interface, bukan kelas, sehingga cara penggunaannya berbeda dengan kelas. Berikut ini adalah User Action, Source Object, dan Tipe Event yang sering digunakan.

User Action	Source Object	Tipe Event
Mengklik suatu button	**JButton**	**ActionEvent**
Mengubah text	**JTextComponent**	**TextEvent**
Menekan tombol Enter pada suatu komponen text field	**JTextField**	**ActionEvent**

Memilih suatu item baru	**JComboBox**	**ItemEvent, ActionEvent**
Memilih satu atau banyak item	**JList**	**ListSelectionEvent**
Memilih suatu check box	**JCheckBox**	**ItemEvent, ActionEvent**
Memilih suatu radio button	**JRadioButton**	**ItemEvent, ActionEvent**
Memilih suatu item menu	**JMenuItem**	**ActionEvent**
Menggerakkan scroll bar	**JScrollBar**	**AdjustmentEvent**
Window terbuka, tertutup, diberi icon atau icon dilepaskan, atau window menutup.	**Window**	**WindowEvent**
Komponen ditambahkan atau dibuang dari container	**Container**	**ContainerEvent**
Komponen digerakkan, diresize, disembunyikan (hidden), atau ditampilkan.	**Component**	**ComponentEvent**
Komponen seketika menjadi focus atau kehilangan focus.	**Component**	**FocusEvent**
Key ditekan atau dilepaskan	**Component**	**KeyEvent**
Mouse ditekan, dilepaskan, diklik, atau ketika mouse memasuki atau keluar dari suatu komponen.	**Component**	**MouseEvent**
Mouse digerakkan atau didrag.	**Component**	**MouseEvent**

7.3 Layout Management

Agar komponen-komponen yang ditempelkan di window utama tertata dengan rapi, maka kita perlu mengatur layout window utama tersebut. Java menyediakan sejumlah class untuk mengatur layout dimana setiap class tersebut memiliki aturan tersendiri dan format layout yang berbeda.

Beberapa Layout Manager yang dikenal oleh Java adalah :

1. Border Layout
2. Flow Layout
3. Card Layout
4. Box Layout
5. Grid Layout
6. Grid Bag Layout
7. Group Layout (Free Design)
8. Null

7.4 GUI Builder

Untuk mendesain tampilan GUI kita juga dapat menggunakan fitur GUI Builder dari NetBeans, untuk mendesain tampilan GUI kita cukup drag-and-drop komponen yang kita inginkan. Dimana komponen-komponennya sudah tersedia di pallete. Pallete pada NetBeans dikelompokkan menjadi 7 yaitu : Swing Container, Swing Control, Swing Menus, Swing Windows, AWT, Beans dan Java Persistance. Secara default layout manager yang digunakan adalah Free Design. Pada pemrograman GUI Komponen Swing lebih sering dipakai dari pada AWT.

Untuk membuat JFrame menggunakan GUI Builder diawali dengan membuat JFrame baru dengan cara : 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 plih JFrame Form. Setelah berhasil dibuat maka akan muncul tampilan GUI Builder.

Untuk memberi Event pada komponen saat menggunakan GUI Builder kita dapat melakukannya dengan cara klik kanan pada komponen yang akan kita berikan aksi, kemudian pilih event, pilih jenis eventnya, kemudian methodnya.

Untuk penggunaan Layout Free Design dengan GUI Builder, sebaiknya hati-hati karena penataan komponennya saling terkait satu dengan yang lain. Kesalahan penempatan bisa menyebabkan penataan yang telah kita buat menjadi kacau, sebaiknya jika menggunakan Free Design kita mulai dari komponen yang menjadi patokan penempatan komponen berikutnya, atau kita juga bisa menggunakan beberapa Layout Manager sebagai bantuan.

DataStatistik.java

```
package pertemuan_ke_07;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Collections;
import java.util.Vector;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

public class DataStatistik extends JFrame implements
ActionListener{
  private String[] header = {"Angka", "Frekuensi"};
  private DefaultTableModel model;
  private JButton btnGenerate;
  private JScrollPane jsc1, jsc2, jsc3;
  private JLabel lbBanyakData, lbMean, lbMedian, lbModus;
  private JList<String> listAsli, listUrut;
  private JPanel pnlKiri;
  private JSpinner spnBanyakData;
  private JTable tbDistribusi;
  private JTextField tfMean, tfMedian, tfModus;

  public DataStatistik() {
super("DATA STATISTIK");  initComponents();
    setExtendedState(JFrame.MAXIMIZED_HORIZ);
}

  private void initComponents() {
pnlKiri = new JPanel();
    lbBanyakData = new JLabel("Banyak Data :");
    lbMean = new JLabel("Mean :");
    lbMedian = new JLabel("Median :");
    lbModus = new JLabel("Modus :");
    spnBanyakData = new JSpinner();
    tfMean = new JTextField(); tfMedian = new JTextField();
    tfModus = new JTextField();
    btnGenerate = new JButton("Generate");
    jsc1 = new JScrollPane(); jsc2 = new JScrollPane();
    jsc3 = new JScrollPane();
    listAsli = new JList<>(); listUrut = new JList<>();
    model = new DefaultTableModel(header, 0);
    tbDistribusi = new JTable();
setDefaultCloseOperation(WindowConstants.
EXIT_ON_CLOSE);
    spnBanyakData.setValue(1);
```

```
    lbBanyakData.setBounds(10, 10, 90, 30);
    lbMean.setBounds(10, 110, 90, 30);
    lbMedian.setBounds(10, 170, 90, 30);
    lbModus.setBounds(10, 230, 90, 30);
    tfMean.setBounds(10, 140, 90, 30);
    tfMedian.setBounds(10, 200, 90, 30);
    tfModus.setBounds(10, 260, 90, 30);
    spnBanyakData.setBounds(10, 40, 90, 30);
    btnGenerate.setBounds(10, 80, 90, 30);

pnlKiri.setLayout(null); pnlKiri.add(lbBanyakData);
    pnlKiri.add(spnBanyakData); pnlKiri.add(btnGenerate);
    pnlKiri.add(lbMean); pnlKiri.add(tfMean);
    pnlKiri.add(lbMedian); pnlKiri.add(tfMedian);
    pnlKiri.add(lbModus); pnlKiri.add(tfModus);

    btnGenerate.addActionListener(this);
    tbDistribusi.setModel(model);
    jsc1.setBorder(BorderFactory.createTitledBorder(
"Data Aslli"));
    jsc1.setViewportView(listAsli);
    jsc2.setBorder(BorderFactory.createTitledBorder(
"Data Terurut"));
    jsc2.setViewportView(listUrut);
jsc3.setBorder(BorderFactory.createTitledBorder(
"Distribusi Data"));
    jsc3.setViewportView(tbDistribusi);

    getContentPane().setLayout(new java.awt.GridLayout());
    getContentPane().add(pnlKiri);
getContentPane().add(jsc1);
    getContentPane().add(jsc2); getContentPane().add(jsc3);

pack();
}

public void actionPerformed(ActionEvent e) {
int n = (Integer) spnBanyakData.getValue(),
angka, sum = 0, idxMaks = 1,
    freq[] = new int[101];
    Vector<String> data = new Vector<>(), urut;
    for (int i = 0; i < 101; i++) freq[i] = 0;
for (int i = 0; i < n; i++) {
angka = 1+((int) (Math.random() * 10000)%100);
        data.add(String.format("%03d",angka));
        sum += angka; freq[angka]++;
}
```

```
      urut = (Vector<String>) data.clone();
      Collections.sort(urut);
      listAsli.setListData(data);
      listUrut.setListData(urut);
      model.setRowCount(0);
      for (int i = 1; i < freq.length; i++){
model.addRow(new String[] {String.format("%03d",i),
        String.valueOf(freq[i])});
        if(freq[idxMaks] < freq[i]) idxMaks = i;
}
    tfMean.setText(String.format("%.3f", (double)sum/n));
    tfModus.setText(String.format("%03d",idxMaks));
    tfMedian.setText(urut.get(n/2));
}

  public static void main(String args[]) {
new DataStatistik().setVisible(true);
}
}
```

7.5 Latihan

1. Pada Project “Latihan_PBO_2018” buat package baru “latihan_07”.
2. Modifikasi file DataStatistik.java untuk munculkan nilai Quartil (Q1, Q2, dan Q3) dan Standar Deviasi dari data yag ada.

BAB 8
Multi Document Interface

Banyak program aplikasi yang ada saat ini merupakan aplikasi yang dapat membuka lebih dari satu jendela kerja (multiple window) atau lebih dikenal dengan MDI (Multiple Document Interface). Kebalikan dari MDI adalah SDI (Single Document Interface). Multiple document interface memang cukup populer dan digunakan di banyak program aplikasi windows. Dengan Java, Anda dapat pula membuat program aplikasi semacam ini. Java menyediakan beberapa kelas untuk maksud tersebut yaitu kelas JDesktopPane dan kelas JinternalFrame, dan JScrollPane. Dalam pembuatan aplikasi multi form tidak disarankan untuk membuatnya dalam satu class saja, tetapi beberapa class sesuai form yang ada.

8.1 JDesktopPane & JinternalFrame

JDesktopPane adalah kelas wadah (container class) untuk membuat MDI. Di dalam membuat MDI, Anda tetap menggunakan kelas JFrame sebagai jendela luar (outer window), kemudian menambahkan obyek dari kelas JDesktopPane ke content pane dari kelas JFrame, selanjutnya menambahkan obyek dari kelas JInternalFrame ke obyek JDesktopPane. Sebelum meletakkan JDesktopPane pada JFrame, layout Container JFrame harus Flow Layout, GridLayout atau BorderLayout.

Kelas JInternalFrame merupakan kelas turunan dari kelas JComponent. JInternalFrame digunakan untuk membuat jendela (window) di dalam jendela yang lain. JInternalFrame mempunyai banyak fitur sebagaimana umumnya frame di perangkat lunak aplikasi window saat ini

seperti dragging, closing, resizing, menjadi icon, menampilkan title dan mendukung menu bar. Kita dapat menggunakan kelas JInternalFrame dengan cara yang hampir sama dengan kelas JFrame. Misalnya, untuk menambahkan komponen di content pane dari kelas JInternalFrame, Kita dapat menggunakan method add, sedangkan untuk menentukan ukuran frame dari obyek internal frame, kita dapat menggunakan method setSize.

8.2 JScrollPane

JSrollPane merupakan komponen kontainer yang berguna untuk menempatkan komponen lain tanpa khawatir komponen didalamnya lebih besar daripada JScrollPane. Ketika komponen yang lebih besar dari JScrollPane, maka secara otomatis akan ada scroll horizontal dan vertikal yang muncul. Untuk mengganti isi dari JSrollPane kita menggunakan method setViewPortView().

8.3 Contoh Penggunaan

1. Buat JPanel baru

FormSatu.java

```
package pertemuan_ke_08;
public class FormSatu extends javax.swing.JPanel {
public FormSatu() { initComponents(); }

  private void initComponents() {
btn1 = new javax.swing.JButton("btn1");
    btn2 = new javax.swing.JButton("btn2");
    btn3 = new javax.swing.JButton("btn3");
    btn4 = new javax.swing.JButton("btn4");
    btn5 = new javax.swing.JButton("btn5");

    setLayout(new java.awt.BorderLayout());
    add(btn1, java.awt.BorderLayout.CENTER);
    add(btn2, java.awt.BorderLayout.NORTH);
    add(btn3, java.awt.BorderLayout.EAST);
    add(btn4, java.awt.BorderLayout.SOUTH);
    add(btn5, java.awt.BorderLayout.WEST);
}

  private javax.swing.JButton btn1, btn2, btn3, btn4, btn5;
}
```

FormDua.java

```
package pertemuan_ke_08;
public class FormDua extends javax.swing.JPanel {
```

```
public FormDua() { initComponents(); }

  private void initComponents() {
btn1 = new javax.swing.JButton("btn1");
    btn2 = new javax.swing.JButton("btn2");
    btn3 = new javax.swing.JButton("btn3");
    btn4 = new javax.swing.JButton("btn4");
    btn5 = new javax.swing.JButton("btn5");

setLayout(new javax.swing.BoxLayout(this,
javax.swing.BoxLayout.Y_AXIS));
    add(btn1); add(btn2); add(btn3); add(btn4); add(btn5);
}

  private javax.swing.JButton btn1, btn2, btn3, btn4, btn5;
}
```

FormTiga.java

```
package pertemuan_ke_08;
public class FormTiga extends javax.swing.JPanel {
public FormTiga() { initComponents(); }

  private void initComponents() {
btn1 = new javax.swing.JButton("btn1");
    btn2 = new javax.swing.JButton("btn2");
    btn3 = new javax.swing.JButton("btn3");
    btn4 = new javax.swing.JButton("btn4");
    btn5 = new javax.swing.JButton("btn5");

    add(btn1); add(btn2); add(btn3); add(btn4); add(btn5);
}

  private javax.swing.JButton btn1, btn2, btn3, btn4, btn5;
}
```

2. BuatJInternalFrame Baru

InframeSatu.java

```
package pertemuan_ke_08;
public class InframeSatu extends javax.swing.JInternalFrame {
public InframeSatu() {
initComponents();
    getContentPane().add(new FormSatu());
    setSize(200,200); setLocation(10,10);
}

  public void tampil(){
if(isShowing()){
```

```
moveToFront();
try {
setSelected(true);
} catch (Exception e) {
System.out.println("error : "+e.getMessage());
}
} else show();
}

private void initComponents() {
setClosable(true);
setDefaultCloseOperation(javax.swing.WindowConstants.
HIDE_ON_CLOSE);
    setTitle("Inframe Satu");
    getContentPane().setLayout(new java.awt.GridLayout
(1, 0));
    pack();
}
}
```

InframeDua.java

```
package pertemuan_ke_08;
public class InframeDua extends javax.swing.JInternalFrame {
public InframeDua() {
initComponents();
    getContentPane().add(new FormDua());
    setSize(200,200); setLocation(220,10);
}

  public void tampil(){
if(isShowing()){
moveToFront();
try {
setSelected(true);
}catch (Exception e) {
System.out.println("error : "+e.getMessage());
}
    } else show();
  }

  private void initComponents() {
setClosable(true);
setDefaultCloseOperation(javax.swing.WindowConstants.
HIDE_ON_CLOSE);
    setMaximizable(true);
    setResizable(true);
    setTitle("Inframe Dua");
    getContentPane().setLayout(new java.awt.GridLayout
```

```
(1, 0));
    pack();
}
}
```

InframeTiga.java

```
package pertemuan_ke_08;
public class InframeTiga extends javax.swing.JInternalFrame {
public InframeTiga() {
initComponents();
    getContentPane().add(new FormTiga());
    setSize(200,200); setLocation(220,220);
}

  public void tampil(){
if(isShowing()){
moveToFront();
      try {
setSelected(true);
}catch (Exception e) {
System.out.println("error : "+e.getMessage());
}
} else show();
}

  private void initComponents() {
setClosable(true);
setDefaultCloseOperation(javax.swing.WindowConstants.
HIDE_ON_CLOSE);
    setIconifiable(true);
    setMaximizable(true);
    setResizable(true);
    setTitle("Inframe Tiga");
    getContentPane().setLayout(new java.awt.GridLayout
(1, 0));
    pack();
}
}
```

3. Buat JFrame Baru (JDesktopPane & JInternalFrame)

cara1.java

```
package pertemuan_ke_08;
import javax.swing.JFrame;
public class cara1 extends javax.swing.JFrame {
private InframeSatu inframeSatu;
  private InframeDua inframeDua;
  private InframeTiga inframeTiga;
```

```
  public cara1() {
initComponents();
    inframeSatu = new InframeSatu();
    inframeDua = new InframeDua();
    inframeTiga = new  InframeTiga();
    desktopPane.add(inframeSatu);
desktopPane.add(inframeDua);
    desktopPane.add(inframeTiga);
    setExtendedState(javax.swing.JFrame.MAXIMIZED_BOTH);
}

  private void initComponents() {
desktopPane = new javax.swing.JDesktopPane();
    menuBar = new javax.swing.JMenuBar();
    menu1 = new javax.swing.JMenu("File");
    menu2 = new javax.swing.JMenu("Edit");
    menuItemSatu = new javax.swing.JMenuItem("Satu");
    menuItemDua = new javax.swing.JMenuItem("Dua");
    menuItemTiga = new javax.swing.JMenuItem("Tiga");

setDefaultCloseOperation(javax.swing.WindowConstants.
EXIT_ON_CLOSE);
    getContentPane().add(desktopPane,
java.awt.BorderLayout.CENTER);
    menuItemSatu.addActionListener(new
java.awt.event.ActionListener() {
public void actionPerformed(
java.awt.event.ActionEvent evt) {
                inframeSatu.tampil();
}
});

    menuItemDua.addActionListener(new
java.awt.event.ActionListener() {
public void actionPerformed(
java.awt.event.ActionEvent evt) {
inframeDua.tampil();
}
});

    menuItemTiga.addActionListener(new
java.awt.event.ActionListener() {
public void actionPerformed(
java.awt.event.ActionEvent evt) {
inframeTiga.tampil();
}
```

```
});

    menu1.add(menuItemSatu); menu1.add(menuItemDua);
menu2.add(menuItemTiga);
    menuBar.add(menu2); menuBar.add(menu1);
setJMenuBar(menuBar);
pack();
}

  public static void main(String args[]) {
new cara1().setVisible(true);
}

  private javax.swing.JDesktopPane desktopPane;
  private javax.swing.JMenu menu1M menu2;
  private javax.swing.JMenuBar menuBar;
  private javax.swing.JMenuItem menuItemDua,
menuItemSatu, menuItemTiga;
}
```

4. Buat JFrame Baru (JScrollPane)

cara1.java

```
package pertemuan_ke_08;
public class cara2 extends javax.swing.JFrame {
public cara2() {
initComponents();
    setExtendedState(javax.swing.JFrame.MAXIMIZED_BOTH);
}

  private void initComponents() {
jsc = new javax.swing.JScrollPane();
    menuBar = new javax.swing.JMenuBar();
    menu1 = new javax.swing.JMenu("File");
    menu2 = new javax.swing.JMenu("Edit");
    menuItemSatu = new javax.swing.JMenuItem("Satu");
    menuItemDua = new javax.swing.JMenuItem("Dua");
    menuItemTiga = new javax.swing.JMenuItem("Tiga");

setDefaultCloseOperation(javax.swing.WindowConstants.
EXIT_ON_CLOSE);
    getContentPane().add(jsc, java.awt.BorderLayout.CENTER);

    menuItemSatu.addActionListener(new
java.awt.event.ActionListener() {
public void actionPerformed(
java.awt.event.ActionEvent evt) {
```

```
jsc.setViewportView(new FormSatu());
}
});

    menuItemDua.addActionListener(new
java.awt.event.ActionListener() {
public void actionPerformed(
java.awt.event.ActionEvent evt) {
jsc.setViewportView(new FormDua());
}
});

    menuItemTiga.addActionListener(new
java.awt.event.ActionListener() {
public void actionPerformed(
java.awt.event.ActionEvent evt) {
jsc.setViewportView(new FormTiga());
}
});
    menu1.add(menuItemSatu); menu1.add(menuItemDua);
menu2.add(menuItemTiga);
    menuBar.add(menu1); menuBar.add(menu2);
setJMenuBar(menuBar);
    pack();
}

public static void main(String args[]) {
new cara2().setVisible(true);
}

  private javax.swing.JScrollPane jsc;
  private javax.swing.JMenu menu1, menu2;
  private javax.swing.JMenuBar menuBar;
  private javax.swing.JMenuItem menuItemDua, menuItemSatu,
menuItemTiga;
}
```

8.4 Catatan Penting

1. Default Layout Manager JDesktopPane adalah Free Design, sebelum menggunakan ubah Layout Magernya menjadi null supaya dapat digunakan dengan mudah.
2. Untuk menggunakan JInternalFrame kita wajib :
   a. Mengatur ukuran menggunakan setSize()
   b. Menampilkannya mengunakan show() atau setVisible(true)
   c. Memasukkan JInternalFrame ke JDesktopPane menggunakan add()

3. Untuk menyembunyikan JInternalFrame kita menggunakan hide() atau setVisible(false)
4. Event close (defaultCloseOperation) dari JInternalFrame sebaiknya HIDE untuk menghemat penggunaan memori.
5. Sebuah JInternalFrame tidak harus memiliki fasilitas closeable, maximizable, resizeable atau iconifiable.

8.5 Latihan

Buat versi Multi Document Interface GUI dari soal latihan pada Bab 6.

BAB 9
Java Database Connectivity

9.1 Java Database Connectivity (JDBC)

Java Database Connectivity adalah komponen API yang disediakan java untuk mengakses database. ada tiga hal yang dikelola oleh JDBC yakni, konek 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 dalan 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 mencara 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 sederhana yang memastikan ftur-ftur 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 meload driver ODBC di setiap komputer client untuk dapat menggunakan bridge ini. Sebagaikonsekuensinya, 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.

9.2 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.

9.3 Membuat Koneksi

Melakukan koneksi ke database melibatkan dua langkah: Meload driver dan membuat koneksi itu sendiri. Cara meload driver sangat mudah, pertama letakkan fle jar database driver ke dalamclasspath. 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 classdatabase 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 diload tadi. Class DriverManager bekerja sama dengan interface Driver untuk mengelola driver-driver yang diload oleh aplikasi, jadi dalam satu sesi anda bisa memuatbeberapa 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 pengenal dari server databse tersebut. Sebagai contoh adalah URL yang digunakan untuk melakukan koneksi ke MySql :

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

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 pasword 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 didefnisikan oleh URL tersebut. Setelah sukses melakukan koneksi ke database, kita dapat mengambil data dari database menggunakan perintah queryataupun melakukan perubahan terhadap database.

9.4 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.

Library JDBC terdiri atas class-class yang berguna untuk setiap tasks di dalam pemrosesan database, misalnya class untuk:

a. Membuat koneksi ke database
b. Membuat statement menggunakan SQL
c. Mengeksekusi query SQL (statement) di dalam database
d. Menampilkan records yang dihasilkan

9.5 Penggunaan JDBC

1. Siapkan Database & tabel yang diperlukan. Database : kuliah

```
CREATE TABLE `tb_dosen` (
`nidn`  char(10) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
`nama`  varchar(100) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
`gender`  char(1) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
`tmp_lahir`  varchar(100) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
`tgl_lahir`  date NOT NULL ,
`alamat`  varchar(255) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
PRIMARY KEY (`nidn`)
)
```

```
CREATE TABLE `tb_mahasiswa` (
`npm`  char(15) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
`nama`  varchar(100) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
`gender`  char(1) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
`tmp_lahir`  varchar(100) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
```

```
`tgl_lahir`  date NOT NULL ,
`alamat`  varchar(255) CHARACTER SET latin1 COLLATE
latin1_swedish_ci NOT NULL ,
PRIMARY KEY (`npm`)
)
```

2. Buat package baru “pertemuan_ke_09”.
3. Tambahkan pustaka MySQL JDBC Driver.
4. Buat class baru Koneksi.java

Koneksi.java

```
package pertemuan_ke_09;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class Koneksi {
private Connection connection;
  private Statement statement;
  private ResultSet resultSet;
  private String url = "jdbc:mysql://localhost/kuliah",
user = "root", passwd = "";
public Koneksi() {
try {
      Class.forName("com.mysql.jdbc.Driver");
      connection = DriverManager.getConnection(url,
user, passwd);
      statement = connection.createStatement(
ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_UPDATABLE);
} catch (ClassNotFoundException ex) {
JOptionPane.showMessageDialog(null,
"Tambahkan library MySQL.");
      System.exit(0);
    } catch (SQLException ex) {
JOptionPane.showMessageDialog(null,
"Periksa Nama Database,Username dan
Password MySQL.");
System.exit(0);
}
}
  public void closeKoneksi(){
try {
connection.close();
} catch (SQLException ex) {
System.out.println("Error : "+ex.getMessage());
}
}
  public boolean executeQuery(String query){
try {
```

```
statement.execute(query);return true;
} catch (SQLException ex) {
System.out.println("Error : "+ex.getMessage());
      return false;
    }
}
public ResultSet showQuery(String query){
resultSet = null;
    try {
resultSet = statement.executeQuery(query);
}catch (SQLException ex) {
System.out.println("Error : "+ex.getMessage());
}
    return resultSet;
}
}
```

5. Buat JFrame baru FormDosen.java

FormMahasiswa.java

```
package pertemuan_ke_09;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
public class FormDosen extends JFrame{
private DefaultTableModel model;
  private String nidn, nama, gender, tmpLhr, tglLhr, alamat;
  private Koneksi koneksi;
  private String select = "SELECT * FROM TB_DOSEN",
header[] = {"NIDN", "Nama", "Gender", "Tempat Lahir",
"Tanggal Lahir", "Alamat"};
  private JButton btnBersih, btnSimpan;
  private ButtonGroup buttonGroup;
  private JScrollPane jsc;
  private JPopupMenu klikKanan;
  private JLabel lbAlamat, lbGender, lbJudul,
lbNIDN, lbNama, lbTglLhr, lbTmpLhr;
  private JMenuItem menuItemHapus;
  private JPanel pnlKiri;
  private JRadioButton rbPria, rbWanita;
  private JTable tbDosen;
  private JTextField tfAlamat, tfNIDN, tfNama,
tfTglLhr, tfTmpLhr;

  public FormDosen () {
super("DATA DOSEN");
    initComponents();
    model = (DefaultTableModel) tbDosen.getModel();
    koneksi = new Koneksi();
    isiTabel();
```

```
    setExtendedState(JFrame.MAXIMIZED_BOTH);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setVisible(true);
}

  private void getIsian(){
nidn = tfNIDN.getText(); nama = tfNama.getText();
    gender = rbPria.isSelected()?
rbPria.getText():rbWanita.getText();
    tmpLhr = tfTmpLhr.getText(); tglLhr = tfTglLhr.getText();
    alamat = tfAlamat.getText();
}

  private void setIsian(boolean bersih){
tfNIDN.setText(bersih?"":nidn);
tfNama.setText(bersih?"":nama);
    if(bersih || gender.equals(rbPria.getText())){
rbPria.setSelected(true); tbDosen.clearSelection();
    }else rbWanita.setSelected(true);
    tfTmpLhr.setText(bersih?"":tmpLhr);
tfTglLhr.setText(bersih?"":tglLhr);
    tfAlamat.setText(bersih?"":alamat);
}

  private Date getDate(){
try {
return new Date(new SimpleDateFormat
("dd/MM/yyyy").parse(tglLhr).getTime());
} catch (ParseException ex) {
System.out.println("Error : "+ex.getMessage());
      return null;
    }
}

  private void simpan(){
String query = select+" where nidn = '"+nidn+"'";
    ResultSet resultSet = koneksi.showQuery(query);
    try {
boolean b = resultSet.next();
      if(!b) resultSet.moveToInsertRow();

      resultSet.updateString("nidn", nidn);
      resultSet.updateString("nama", nama);
      resultSet.updateString("gender",
gender.substring(0, 1));
      resultSet.updateString("tmp_lahir", tmpLhr);
      resultSet.updateDate("tgl_lahir", getDate());
      resultSet.updateString("alamat", alamat);
      if(b){
resultSet.updateRow();
        JOptionPane.showMessageDialog(this,
"Pengubahan berhasil.");
      } else {
```

```
resultSet.insertRow(); resultSet.moveToCurrentRow();
        JOptionPane.showMessageDialog(this,
"Penambahan Data berhasil.");
      }
    } catch (SQLException ex) {
System.out.println("Error : "+ex.getMessage());
    }
}

  private void hapus(){
String query = select+" where nidn = '"+nidn+"'";
    ResultSet resultSet = koneksi.showQuery(query);
    try {
boolean b = resultSet.next();
      if(b){
resultSet.deleteRow();
JOptionPane.showMessageDialog(this,
"Penghapusan berhasil.");
} else {
JOptionPane.showMessageDialog(this,
"Penghapusan gagal.");
      }
    } catch (SQLException ex) {
System.out.println("Error : "+ex.getMessage());
    }
}

  private String[] getData(ResultSet resultSet){
String s, isi[] = new String[6]; Date d;
    try {
isi[0] = resultSet.getString("nidn");
      isi[1] = resultSet.getString("nama");
      s = resultSet.getString("gender");
      isi[2] = s.equals("P")?"Pria":"Wanita";
      isi[3] = resultSet.getString("tmp_lahir");
      d = resultSet.getDate("tgl_lahir");
      java.util.Date d2 = (java.util.Date) d;
      isi[4] = new SimpleDateFormat("dd/MM/yyyy").format(d2);
      isi[5] = resultSet.getString("alamat");
    } catch (SQLException ex) {
System.out.println("Error : "+ex.getMessage());
    }
    return isi;
}

  private void isiTabel(){
ResultSet resultSet = koneksi.showQuery(select);
    try {
model.setRowCount(0);
      while(resultSet.next())
model.addRow(getData(resultSet));
    } catch (SQLException ex) {
System.out.println("Error : "+ex.getMessage());
    }
```

```
}

  private void initComponents() {
buttonGroup = new ButtonGroup();
    klikKanan = new JPopupMenu();
    menuItemHapus = new JMenuItem("Hapus");
    pnlKiri = new JPanel();
    lbJudul = new JLabel("DATA DOSEN");
    lbNIDN = new JLabel("NIDN :");
    lbNama = new JLabel("Nama :");
    lbGender = new JLabel("Gender :");
    lbTmpLhr = new JLabel("Tempat Lahir :");
    lbTglLhr = new JLabel("Tanggal Lahir : (dd/mm/yyyy)");
    lbAlamat = new JLabel("Alamat :");
    tfNIDN = new JTextField();
    tfNama = new JTextField();
tfTmpLhr = new JTextField();
    tfTglLhr = new JTextField();
    tfAlamat = new JTextField();
    rbPria = new JRadioButton("Pria");
    rbWanita = new JRadioButton("Wanita");
    btnSimpan = new JButton("Simpan");
    btnBersih = new JButton("Bersihkan");
    jsc = new JScrollPane();
    tbDosen = new JTable();
    model = new DefaultTableModel(header, 0);

klikKanan.add(menuItemHapus);
    lbJudul.setFont(new java.awt.Font("Tahoma", 1, 14));
    buttonGroup.add(rbPria); buttonGroup.add(rbWanita);
    rbPria.setSelected(true);

int PS = GroupLayout.PREFERRED_SIZE,
DS = GroupLayout.DEFAULT_SIZE;
    GroupLayout.Alignment LEAD =
GroupLayout.Alignment.LEADING,
      TRAILING = GroupLayout.Alignment.TRAILING,
      BASELINE = GroupLayout.Alignment.BASELINE;
    LayoutStyle.ComponentPlacement RELATED =
LayoutStyle.ComponentPlacement.RELATED;
GroupLayout pnlKiriLayout = new GroupLayout(pnlKiri);
    pnlKiri.setLayout(pnlKiriLayout);
    pnlKiriLayout.setHorizontalGroup(
pnlKiriLayout.createParallelGroup(LEAD)
      .addGroup(pnlKiriLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(pnlKiriLayout.createParallelGroup(LEAD)
.addGroup(pnlKiriLayout.createSequentialGroup()
.addComponent(rbPria).addPreferredGap(RELATED)
.addComponent(rbWanita))
.addComponent(lbJudul).addComponent(lbNIDN)
.addComponent(tfNIDN, PS, 150,PS)
.addComponent(lbNama)
```

```
.addComponent(tfNama, PS, 300,PS)
.addComponent(lbGender)
.addComponent(lbTmpLhr).addComponent(lbTglLhr)
.addGroup(pnlKiriLayout
.createParallelGroup(TRAILING, false)
.addComponent(tfTglLhr, LEAD, DS, 150,
Short.MAX_VALUE)
.addComponent(tfTmpLhr, LEAD))
          .addComponent(lbAlamat)
.addComponent(tfAlamat, PS, 400, PS)
          .addComponent(btnSimpan).addComponent(btnBersih))
.addContainerGap(DS, Short.MAX_VALUE))
);
    pnlKiriLayout.setVerticalGroup(
pnlKiriLayout.createParallelGroup(LEAD)
.addGroup(pnlKiriLayout.createSequentialGroup()
.addContainerGap().addComponent(lbJudul)
.addPreferredGap(RELATED)
          .addComponent(lbNIDN).addPreferredGap(RELATED)
          .addComponent(tfNIDN, PS, DS, PS)
.addPreferredGap(RELATED)
          .addComponent(lbNama).addPreferredGap(RELATED)
          .addComponent(tfNama, PS, DS, PS)
.addPreferredGap(RELATED)
          .addComponent(lbGender).addPreferredGap(RELATED)
          .addGroup(pnlKiriLayout
.createParallelGroup(BASELINE)
            .addComponent(rbPria).addComponent(rbWanita))
.addPreferredGap(RELATED)
          .addComponent(lbTmpLhr).addPreferredGap(RELATED)
          .addComponent(tfTmpLhr, PS, DS, PS)
.addPreferredGap(RELATED)
          .addComponent(lbTglLhr).addPreferredGap(RELATED)
          .addComponent(tfTglLhr, PS, DS, PS)
.addPreferredGap(RELATED)
          .addComponent(lbAlamat).addPreferredGap(RELATED)
          .addComponent(tfAlamat, PS, DS, PS)
.addPreferredGap(RELATED)
          .addComponent(btnSimpan).addPreferredGap(RELATED)
          .addComponent(btnBersih).addContainerGap(DS,
Short.MAX_VALUE))
);

getContentPane().add(pnlKiri,
java.awt.BorderLayout.LINE_START);
    tbDosen.setModel(model);
    tbDosen.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);

    jsc.setViewportView(tbDosen);
    tbDosen.getColumnModel().getColumn(0)
.setPreferredWidth(90);
    tbDosen.getColumnModel().getColumn(1)
.setPreferredWidth(150);
    tbDosen.getColumnModel().getColumn(2)
```

```
.setPreferredWidth(75);
    tbDosen.getColumnModel().getColumn(3)
.setPreferredWidth(100);
    tbDosen.getColumnModel().getColumn(4)
.setPreferredWidth(150);
    tbDosen.getColumnModel().getColumn(5)
.setPreferredWidth(300);
    getContentPane().add(jsc, java.awt.BorderLayout.CENTER);

    menuItemHapus.addActionListener(new
java.awt.event.ActionListener() {
public void actionPerformed(
java.awt.event.ActionEvent evt) {
int idxRow = tbDosen.getSelectedRow();
            nidn = model.getValueAt(idxRow, 0).toString();
            hapus(); isiTabel();
}
});

    btnSimpan.addActionListener(new
java.awt.event.ActionListener() {
public void actionPerformed(
java.awt.event.ActionEvent evt) {
getIsian(); simpan(); isiTabel();
}
});

    btnBersih.addActionListener(new
java.awt.event.ActionListener() {
public void actionPerformed(
java.awt.event.ActionEvent evt) {
setIsian(true);
}
});

    tbDosen.addMouseListener(new
java.awt.event.MouseAdapter() {
public void mouseClicked(
java.awt.event.MouseEvent evt) {
if(evt.getButton() == 3){
int row = tbDosen.rowAtPoint(evt.getPoint());
              tbDosen.clearSelection();
              tbDosen.addRowSelectionInterval(row, row);
              klikKanan.show(evt.getComponent(),
evt.getX(), evt.getY());
            } else {
int idxRow = tbDosen.getSelectedRow();
              nidn = model.getValueAt(idxRow, 0).toString();
              nama = model.getValueAt(idxRow, 1).toString();
              gender = model.getValueAt(
idxRow, 2).toString();
              tmpLhr = model.getValueAt(
idxRow, 3).toString();
```

```
                tglLhr = model.getValueAt(
idxRow, 4).toString();
                alamat = model.getValueAt(
idxRow, 5).toString();
                setIsian(false);
              }
}
});
    pack();
}

public static void main(String args[]) {
new Form_Dosen();
}
}
```

9.6 Latihan

Lengkapi aplikasi dengan Form Mata Kuliah, Form Jadwal dan Form Rombel yang disesuaikan dengan tabel tb_mata_kuliah, tb_jadwal dan tb_rombel. Tambahkan juga fitur jika ada isian yang belum diisi, maka program memberikan peringatan bahwa isian tersebut harus diisi dan requestFocus() isian tersebut.

Kode SQL tb_mata_kuliah, tb_jadwal dan tb_rombel

```
CREATE TABLE `tb_mata_kuliah` (
  `kode_mk` char(7) NOT NULL PRIMARY KEY,
  `nama_mk` varchar(200) NOT NULL,
  `sks` tinyint(1) NOT NULL,
  `semester` tinyint(1) NOT NULL
)

CREATE TABLE `tb_rombel` (
  `tingkat1` tinyint(1) NOT NULL,
  `tingkat2` tinyint(1) NOT NULL,
  `tingkat3` tinyint(1) NOT NULL,
  `tingkat4` tinyint(1) NOT NULL
}

CREATE TABLE `tb_jadwal` (
  `kode_mk` char(7) NOT NULL,
  `kelas` char(1) NOT NULL,
  `nidn` char(10) NOT NULL,
  `hari` char(1) NOT NULL,
  `jam_mulai` char(5) NOT NULL,
  `jam_selesai` char(5) NOT NULL,
  `ruang` varchar(5) NOT NULL,
PRIMARY KEY(`kode_mk`,`kelas`)
)
```

BAB 10
Java Report

10.1 Report

Membuat laporan terkadang menjadi hal yang cukup melelahkan, apalagi dengan data yang begitu banyak dan berbeda. Kesalahan dalam melakukan Input ataupun kesalahan manusia terkadang mengurangi akurasi dalam pembuatan sebuah laporan. Namun seiring dengan berkembangnya zaman, beberapa solusi untuk mempermudah membuat laporan telah disediakan baik offline maupun online, dari sekian banyak salah satunya adalah iReport.

JasperReport merupakan library di lingkungan Java untuk pemroses laporan. Dengan library ini, kita dapat menampilkan laporan dalam bentuk print preview, melakukan export ke beberapa format dokumen lain (antara lain PDF, HTML, text, Excel), menampilkan gambar, grafik maupun tabel.

Laporan yang kita buat nantinya dapat dikaitkan ke database berdasarkan connection string dan sql yang kita inginkan. JasperReport mendasarkan format dokumen definisi laporan yang akan dikompilasi berbasis pada XML, sehingga nantinya dapat dengan mudah dapat dikonversi ke format dokumen lain dengan memanfaatkan XSLT ataupun FO (Format Object).

iReport adalah merupakan perangkat lunak bantu untuk perancangan laporan secara visual yang nantinya dapat di kompilasi dengan menggunakan JasperReport sehingga menjadi file *.jasper atau *.jrxml yang dapat langsung dipanggil oleh program Java.

10.2 Bidang Desain Report

1. Background.Background disini dapat diisi dengan gambar maupun text, yang nantinya akan menjadi background pada setiap halaman dalam report.
2. Title. Title akan dicetak sekali pada bagian paling atas report. title ini dapat diisi dengan judul report atau kop report.
3. Page Header. Page Header akan dicetak pada bagian atas di setiap report.
4. Column Header. Column Header akan dicetak pada bagian atas tabel / kolom pada report. biasanya digunakan untuk nama kolom.
5. Detail. detail adalah isi dari report itu sendiri. Biasanya komponen yang berada pada detail ini adalah field yang nantinya akan dicetak sebanyak data dari hasil query database.
6. Column Footer. Sama dengan column Header, hanya saja dicetak di bagian bawah.
7. Page Footer. Sama dengan page Header, hanya saja dicetak pada bagian bawah.
8. Last Page Footer. Last Page Footer akan dicetak sekali pada bagian bawah halaman report paling belakang
9. Summary. Summary akan dicetak pada halaman paling belakang dari report. biasanya diisi dengan grafik atau keterangan umum dari report.

10.3Membuat Report

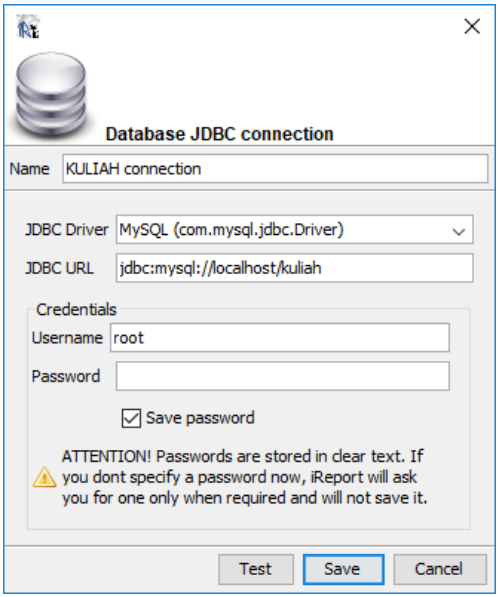
1. Buka iReport
2. Buka setting Datasource, buat koneksi baru dengan tipe Database JDBC connection.
3. Kemudian isi field yang disediakan seperti pada gambar 10.1. Kemudian klik Test, jika muncul pesan gagal cek isian dan koneksi database anda.
4. Setelah muncul pesan sukses, klik Save.
5. Buat File report baru (Empty Report)
6. Beri nama “cetakDosen”, Kemudian Finish.
7. Buka setting Report Query, ketikkan perintah SQL berikut :

```
SELECT nidn,
       nama,
       CASE gender WHEN 'P' THEN 'Pria‘
       ELSE 'Wanita' END AS Gender,
```

```
        tmp_lahir,
        DATE_FORMAT(tgl_lahir,'%m/%d/%Y') AS tgl_lahir,
        alamat
 FROM tb_dosen
 ORDER BY 2
```

Jika perintah SQL yagn diketikkan benar maka akan tampil nama field-field hasil dari perintah SQL. Jika sudah benar maka klik OK.

8. Pada cendela Report Inspector field-field hasil query akan muncul di bagian Fields.



Gambar 10. 1

9. Hapus bidang desain report yang tidak dipakai : Title, Column Footer, Page Footer, Last Page Footer, Summary, Background.
10. Atur tinggi Band (Band Height) :
    a. Page Header : 68
    b. Column Header : 21
    c. Detail 1 : 21
11. Drag & drop Static Text dari cendela Palette, letakkan pada bagian Page Header (posisi bebas). Kemudian atur properties tiap static text seperti tabel 10.1.

12. Drag & drop Static Text dari cendela Palette, letakkan pada bagian Column Header (posisi bebas). Kemudian atur properties tiap static text seperti tabel 10.2 & 10.3.
13. Beri garis tepi untuk tiap static teks yang berada pada Column Header.
14. Drag & drop Text Field dari cendela Palette, letakkan pada bagian Detail 1 (6 buah, posisi bebas). Kemudian atur properties tiap text field seperti tabel 10.4 & 10.5.
15. Beri Garis tepi untuk tiap Text Field yang berada pada Detail 1.
16. Klik Preview untuk melihat hasil Report.

Tabel 10. 1

Properties	**ST1**	**ST2**	**ST3**
Left	0	0	0
Top	0	20	40
Width	802	802	802
Text	DATA DOSEN	PROGRAM STUDI TEKNIK INFORMATIKA	UNIVERSITAS NUSANTARA PGRI KEDIRI
Size	14	14	14
Bold	TRUE	TRUE	TRUE
Horizontal Allignment	Center	Center	Center

Tabel 10. 2

Properties	**ST1**	**ST2**	**ST3**
Left	0	72	312
Top	0	0	0
Width	72	240	55
Opaque	TRUE	TRUE	TRUE
Backcolor	[204,255,255]	[204,255,255]	[204,255,255]
Text	NIDN	Nama	Gender
Size	11	11	11
Bold	TRUE	TRUE	TRUE
Vertical Alignment	MIDDLE	MIDDLE	MIDDLE

Tabel 10. 3

Properties	**ST4**	**ST5**	**ST6**
Left	367	455	535
Top	0	0	0
Width	88	80	267
Opaque	TRUE	TRUE	TRUE
Backcolor	[204,255,255]	[204,255,255]	[204,255,255]
Text	Tempat Lahir	Tanggal Lahir	Alamat
Size	11	11	11
Bold	TRUE	TRUE	TRUE
Vertical Alignment	MIDDLE	MIDDLE	MIDDLE

Tabel 10. 4

Properties	**TF1**	**TF2**	**TF3**
Left	0	72	312
Top	0	0	0
Width	72	240	55
Text Field Expression	" "+$F{nidn}	" "+$F{nama}	" "+$F{Gender}
Size	11	11	11
Vertical Alignment	MIDDLE	MIDDLE	MIDDLE

Tabel 10. 5

Properties	**TF4**	**TF5**	**TF6**
Left	367	455	535
Top	0	0	0
Width	88	80	267
Text Field Expression	" "+$F{tmp_lahir}	" "+$F{tgl_lahir}	" "+$F{alamat}
Size	11	11	11
Vertical Alignment	MIDDLE	MIDDLE	MIDDLE

10.4 Memanggil File Report menggunakan Java

Untuk mencetak report yang sudah kita buat menggunakan iReport, kita perlu menambahkan Library Jasper Report. File yang perlu kita tambahkan ke Library PROJECT kita adalah :

- commons-beanutils-1.8.2.jar
- commons-collections-3.2.1.jar
- commons-digester-1.7.jar
- commons-logging-1.1.jar
- jasperreports-4.6.0.jar

Buat File java untuk memanggil file Report.

cetakReportDosen.java

```
package pertemuan_ke_10;
import java.io.File;
import javax.swing.JOptionPane;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.util.JRLoader;
import net.sf.jasperreports.view.JasperViewer;
import pertemuan_ke_09.Koneksi;
public class cetakReportDosen extends javax.swing.JFrame {
public cetakReportDosen() {initComponents();}
  private void initComponents() {
btnCetak = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.
EXIT_ON_CLOSE);
    getContentPane().setLayout(new java.awt.GridLayout());
    btnCetak.setText("Cetak Dosen");
    btnCetak.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(
java.awt.event.ActionEvent evt) {
btnCetakActionPerformed(evt);
            }
});
    getContentPane().add(btnCetak);pack();
}

  private void btnCetakActionPerformed(
java.awt.event.ActionEvent evt) {
    try {
Koneksi con = new Koneksi();
String namaFile = "D:/PRAKTIKUM_PBO_2018/src/
pertemuan_ke_10/cetakDosen.jasper";
      JasperReport jasperReport =
(JasperReport)JRLoader.loadObject(new
File(namaFile));
```

```
      JasperPrint jasperPrint =
JasperFillManager.fillReport(jasperReport,
        null, con.getConnection());
JasperViewer.viewReport(jasperPrint,false);
    } catch (Exception e) {
      System.out.println(e);
      JOptionPane.showMessageDialog(null,
"Terjadi kesalahan system, Pencetakan gagal!");
}
}
  public static void main(String args[]) {
new cetakReportDosen().setVisible(true);
}
  private javax.swing.JButton btnCetak;
}
```

10.5 Catatan Penting

Software iReport tidak dapat berjalan menggunakan Java versi 8 atau sesudahnya, karena iReport versi terakhir dibuat saat Java versi 7, sehingga untuk menjalankannya kita memerluklan JDK/JRE versi 7 atau sebelumnya.

Apabila software iReport tidak muncul, maka perlu setting pada file ireport.conf, tambahkan statement : jdkhome="C:\Program Files\Java\jre7".

10.6Latihan

Buat report untuk mencetak Mata Kuliah, Mahasiswa per kelas, Jadwal, Nilai Mahasiswa per kelas per matakuliah.

DAFTAR PUSTAKA

Bambang, H. 2010. *Esensi-esensi Bahasa pemrograman JAVA*. Bandung: Informatika.

Hartati, Sri. 2007. *Pemrograman GUI Swing Java dengan Netbeans 5*. Yogyakarta: Andi.

Hendric, Spits Warnars Harco Leslie. 2006. *Pemrograman Grafik dengan Java*. Yogyakarta: Graha Ilmu.

Kadir, A. 2004. *Dasar Pemrograman Java 2*. Yogyakarta: Andi.

Nugroho, A. 2009. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Yogyakarta: Andi.

Nugroho, Adi. 2004. *Pemrograman Berorientasi Objek*. Bandung: Informatika.

Raharjo, Budi. Imam Heryanto. Arif Haryono. 2010. *Mudah Belajar Java – Edisi Revisi*. Bandung: Informatika.

Sutopo, A. H., & Masya, F. 2005. *Pemrograman Berorientasi Objek dengan Java*. Yogyakarta: Erlangga.

Wijono, G Sri Hartati. 2005. *Java 2 SE dengan JBuilder*. Yogyakarta: Andi.

Berbagai macam artikel yang diambil dari situs http://www.netbeans.org dan http://java.oracle.com.

Catatan

Catatan

Catatan