



MENGENAL STRUKTUR BAHASA JAVA

Pertemuan ke-2

Sub-CPMK

- *Mahasiswa mampu mengenal struktur dasar pemrograman Bahasa Java (C2, A2).*
- *Mahasiswa mampu menentukan variabel yang dibutuhkan dengan tipe data yang sesuai dalam penyusunan program konsep Pemrograman Berorientasi Objek (PBO). (C3, A3).*

Materi

1. Sejarah Bahasa Pemrograman Java
2. Keyword Pada Java
3. Tipe Data Pada Java
4. Jenis-jenis Operator Pada Java
5. Fungsi Input dan Output Pada Java
6. Jenis-Jenis Exception Pada Java
7. Membuat Program Dalam Tampilan Teks & GUI



1. Sejarah Bahasa Pemrograman Java

- Java dikembangkan oleh Sun Microsystem, direlease sekitar tahun 1991 dengan nama Oak.
- Sekitar tahun 1995 nama Oak diganti dengan nama Java.
- Yang menjadi ide dasar dari bahasa pemrograman Java adalah untuk membuat bahasa pemrograman yang portable dan independent terhadap platform (platform independent).

Lanj...

- Bila menggunakan bahasa pemrograman Java dapat membuat perangkat lunak yang bisa ditanamkan (*embedded*) pada berbagai mesin dan peralatan seperti handphone, microwave, remote control, dan lainnya.
- Yang menjadi konsep bahasa pemrograman Java adalah: *write once run anywhere*.
- Sintaks pada bahasa pemrograman Java merupakan pengembangan dari bahasa pemrograman C/C++.

1.1 Karakteristik Java

- Bahasa pemrograman Java memiliki karakteristik, yaitu:
 - Berorientasi Objek.
 - Platform Independent.
 - Sederhana.
 - Aman.
 - Bersifat Architectural-neutral.

1.1 Karakteristik Java (Lanj...)

- Portabel.
- Kuat dan Powerful.
- Multithreaded.
- Terinterpretasi.
- Performa tinggi.
- Terdistribusi.
- Dinamis.

1.2 Arsitektur Java

- Sun membagi arsitektur bahasa pemrograman Java menjadi tiga kelompok, yaitu:
 - **Enterprise Java (J2EE)**, untuk aplikasi berbasis WEB.
 - **Standar Java (J2SE)**, J2SE yang biasa dikenal sebagai bahasa Java.
 - **Micro Java (J2ME)**, merupakan subset dari J2SE, banyak digunakan untuk mengembangkan aplikasi wireless device/mobile device.

1.3 IDE NetBeans

- IDE (*Integreted Development Environment*) adalah software yang digunakan untuk mengembangkan sebuah aplikasi.
- Ada banyak IDE yang dapat digunakan untuk membuat progam.
- Untuk bahasa JAVA dapat menggunakan Net Beans, jGRASP, atau JCreator, Eclipse.

1.3 IDE NetBeans (Lanj...)

- NetBeans dimulai sekitar tahun 1996 sebagai Xelfi yang merupakan proyek IDE pada Universitas Charles di Praha.
- Pada tahun 1997, Roman Stanek mengomersialkan NetBeans IDE dan kemudian dibeli oleh Sun Microsystems pada tahun 1999.
- Pada tahun 2000, Sun (dan NetBeans) diakuisisi oleh Oracle.

1.4 Struktur Java

Kelas: clsSegitiga

```
1. package segitiga;
2. import java.util.Scanner; — 2. Untuk menyertakan fungsi-fungsi library
3.
4. public class clsSegitiga {
5.     int Alas;
6.     int Tinggi; > 3. Deklarasi variabel
7.
8.     public double Luas() — 4. Method
9.     {
10.         return 0.5 * Alas * Tinggi; — 5. Statemen
11.     }
12. }
```

1.4 Struktur Java (Lanj...)

Main Program: segitiga

```

1.  package segitiga;
2.  import java.util.Scanner; — 4. Untuk menyertakan fungsi-fungsi library
3.
4.  public class Segitiga {      1. Fungsi main()
5.
6.      public static void main(String[] args) { — 5. Kurung kurawal buka
7.          // TODO code application logic here — 3. Komentar
8.          Scanner input = new Scanner (System.in);
9.          //membuat objek dari clsSegitiga
10.         clsSegitiga objSegitiga = new clsSegitiga(); — 2. Deklarasi variabel
11.
12.         //masukkan nilai alas dan tinggi
13.         System.out.printf("Masukkan Alas : ");
14.         objSegitiga.Alas = input.nextInt(); — 6. Statemen
15.         System.out.printf("Masukkan Tinggi : ");
16.         objSegitiga.Tinggi = input.nextInt();
17.         //Cetak luas segitiga
18.         System.out.printf("Luas segitiga : %.2f\n",objSegitiga.Luas());
19.     } — 5. Kurung kurawal tutup
20. }
  
```



2. Keyword Pada Java

- Kata kunci atau keyword merupakan istilah-istilah yang sudah dikenal dalam bahasa pemrograman.
- Sehingga keyword tersebut tidak boleh digunakan untuk mendeklarasikan nama variabel, dan lainnya.
- Berikut jenis-jenis keyword yang dikenal pada bahasa pemrograman java.

2.1 Jenis-Jenis Keyword

Keyword Pada Bahasa Pemrograman Java

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



3. Tipe Data Pada Java

3.1 Tipe Data Pada Java

Jenis Data	Deskripsi	Ukuran	Minimum s/d Maksimum
boolean	benar/salah	1-bit	nilai default adalah false
char	Karakter Unicode	16-bit	\u0000 s/d \uffff
byte	Bilangan bulat	8-bit	-127 s/d 128
short	Bilangan bulat	16-bit	-32768 s/d 32767
int	Bilangat bulat	32-bit	-2147483648 s/d 2147483647
long	Bilangan bulat	64-bit	-9223372036854775808 s/d 9223372036854775807
float	Bilangan riil	32-bit	1.40129846432481707e-45 s/d 3.40282346638528860e+38
double	Bilangan riil	64-bit	4.94065645841246544e-324 s/d 1.79769313486231570e+308



4. Jenis-jenis Operator Pada Java

- Jenis-jenis operator pada bahasa pemrograman java:
 - Operator Aritmetik
 - Operator Tunggal
 - Operator Pembanding
 - Operator Logika
 - Operator Assignment
 - Preseden Operator-operator Java

4.1 Aritmetik

- Operator aritmetika digunakan untuk melakukan operasi-operasi matematika.

Simbol	Nama Operator	Contoh
+	Operator penjumlahan	<code>x=x+1;</code>
-	Operator pengurangan	<code>x=x-1;</code>
*	Operator perkalian	<code>x=x*1;</code>
/	Operator pembagian	<code>x=x/1;</code>
%	Operator sisa pembagian	<code>x=x%1;</code>
+	Operator penyambung string	<code>x="Saya"+"makan";</code>

4.2 Tunggal

- Berikut tentang operator tunggal.

Simbol	Nama Operator	Operasi
+	Operator plus	Menyatakan nilai positif
-	Operator minus	Menyatakan nilai negatif
++	Operator kenaikan	Menambah satu bilangan dengan 1
--	Operator penurunan	Mengurangkan satu bilangan dengan 1
!	Operator lawan	Membalik nilai suatu boolean

4.3 Pembandingan

- Berikut tentang operator pembandingan.

Simbol	Nama Operator	Contoh
<code>==</code>	Sama dengan	<code>b=(1==2);</code>
<code>!=</code>	Tidak sama dengan	<code>b=(1!=2);</code>
<code>></code>	Lebih besar	<code>b=(1>2);</code>
<code>>=</code>	Lebih besar sama dengan	<code>b=(1>=2);</code>
<code><</code>	Lebih kecil	<code>b=(1<2);</code>
<code><=</code>	Lebih kecil sama dengan	<code>b=(1<=2);</code>
<code>?:</code>	Bentuk pendek dari if-then-else	

4.4 Logika

- Berikut tentang operator logika.

Simbol	Nama Operator	Contoh
&&	Conditional AND	b=true&&false;
	Conditional OR	b=true false;
!	NOT	b=true;

4.5 Assignment

- Berikut tentang operator assignment.

Operator	Deskripsi	Contoh
=	Mengirim nilai dari operand sebelah kanan ke operand sebelah kiri.	$C=A+B$
+=	Menjumlahkan nilai operan disebelah kiri dengan nilai operand disebelah kanan dan memasukan hasilnya ke dalam operand disebah kiri.	$B+=A$ sama dengan melakukan perintah $B=B+A$
-=	Mengurangkan nilai operan disebelah kiri dengan nilai operand disebelah kanan dan memasukan hasilnya ke dalam operand disebah kiri.	$B-=A$ sama dengan melakukan perintah $B=B-A$

4.5 Assignment (Lanj...)

Operator	Deskripsi	Contoh
$\ast =$	Mengalikan nilai operan disebelah kiri dengan nilai operand disebelah kanan dan memasukan hasilnya ke dalam operand disebah kiri.	$B \ast = A$ sama dengan melakukan perintah $B = B \ast A$
$/ =$	Membagi nilai operan disebelah kiri dengan nilai operand disebelah kanan dan memasukan hasilnya ke dalam operand disebah kiri.	$B / = A$ sama dengan melakukan perintah $B = B / A$
$\% =$	Menggunakan operasi modulus terhadap kedua operand dan memasukan hasilnya ke dalam operand di sebelah kiri.	$B \% = A$ sama dengan melakukan perintah $B = B \% A$

4.6 Preseden

- Untuk urutan preseden operator pada Java.

Katagori	Operator	Asosiasi
Postfix	(){}.(operator titik)	Kiri ke kanan
Unary	++--!~	Kanan ke kiri
Multiplicative	*/%	Kiri ke kanan
Additive	+-	Kiri ke kanan
Shift	>>>><<	Kiri ke kanan
Relasional	>>=<<=	Kiri ke kanan
Persamaan	==!=	Kiri ke kanan

4.6 Preseden (Lanj...)

Katagori	Operator	Asosiasi
Bitwise AND	&	Kiri ke kanan
Bitwise XOR	^	Kiri ke kanan
Bitwise OR		Kiri ke kanan
Logical AND	&&	Kiri ke kanan
Logical OR		Kiri ke kanan
Kondisional	?:	Kanan ke kiri
Assignment	= += -= *= /= %= >> << = & = ^ = =	Kanan ke kiri
Koma	,	Kiri ke kanan



5. Fungsi Input Dan Output Pada Java

- Fungsi output digunakan untuk menampilkan suatu nilai ke layar (screen).
- Pada bahasa pemrograman Java ada beberapa fungsi output yang dikenal, yaitu:
 - **print()**
 - **println()**
 - **printf()**

Lanj...

- **print()** digunakan untuk menampilkan informasi ke layar dalam satu baris, tanpa berpindah baris baru.
- Sintaks:
 - `System.out.print("string"+arguments);`
- Contoh:
 - `System.out.print("Luas segitiga: "+luas);`

Lanj...

- **println()** digunakan untuk menampilkan informasi ke layar, dan berpindah baris baru.
- Sintaks:
 - `System.out.println("String"+arguments);`
- Contoh:
 - `System.out.println("Luas segitiga: "+luas);`

Lanj...

- **printf()** digunakan untuk menampilkan informasi ke layar.
- Sintaks:
 - `System.out.printf("Luas segitiga: ",luas);`
- Contoh:
 - `System.out.printf("Luas segitiga: %d",luas);`

5.1 Conversion Specifier

- Berikut jenis-jensi conversion specifier.

Specifier	Untuk menampilkan variabel atau pernyataan dengan jenis tipe data.
%d	Bilangan bulat (integer)
%f	Bilangan berkoma (floating point)
%c	Karakter (character)
%s	String (kumpulan karakter)

5.2 Escape Sequence

Sequence	Untuk menampilkan variabel
\n	Membuat baris baru (Enter)
\t	Memberikan identasi (Tab)
\b	Menyisipkan backspace
\r	Menyisipkan carriage retron
\f	Menyisipkan feed dalam teks
\\	Menampilkan karakter backslash
\'	Menampilkan tanda kutip satu
\"	Menampilkan tanda kutip ganda
%%	Manampilkan tanda %

Contoh: `System.out.printf("Luas segitiga: \n%d",luas);`

5.3 Character Methods

- Character methods pada java.

Method	Keterangan.
isLetter()	Menentukan apakah nilai char yang ditentukan adalah huruf
isdigit()	Menentukan apakah nilai char yang ditentukan adalah digit angka
isWhitespace()	Menentukan apakah nilai char yang ditentukan adaah spasi

5.3 Character Methods(Lanj..)

Method	Keterangan.
isUpperCase()	Menentukan apakah nilai char yang ditentukan adalah huruf besar
isLowerCase()	Menentukan apakah nilai char yang ditentukan adalah huruf kecil
toUpperCase()	Mengembalikan bentuk huruf kecil dari nilai char yang ditentukan
toStringt()	Mengembalikan sebuah objek String yang mewakili character tertentu

5.4 Fungsi Input Dasar

- Fungsi input digunakan untuk menerima masukan dari user ke dalam program.
- Kita harus membuat scanner terlebih dahulu sebelum menerima input dari user.
- Scanner disimpan di dalam library file (packge) yang bernama `java.util.Scanner`. Bila akan menggunakannya maka lakukan **import** **java.util.Scanner**;

5.4 Fungsi Input Dasar (Lanj...)

- Selanjutnya lakukan deklarasi scanner bila dalam program akan menerima input dari user dengan sintaks.
 - `Scanner namaScanner = new Scanner(System.in);`
- Contoh:
 - `Scanner input = new Scanner (System.in);`

5.4 Fungsi Input Dasar (Lanj...)

No	Perintah	Keterangan
1	next()	Fungsi next() digunakan untuk menerima masukan dengan tipe data karakter
2	nextLine()	Fungsi nextLine() digunakan untuk menerima masukan dengan tipe data string
3	nextInt	Fungsi nextInt() digunakan untuk menerima masukan dengan tipe data bulat (integer)
4	nextFloat()	Fungsi nextFloat() digunakan untuk menerima masukan dengan tipe data berkoma (float)

5.5 Manipulasi String

No	Perintah	Keterangan
1	toUpperCase()	Mengubah huruf kecil menjadi huruf kapital
2	toLowerCase()	Mengubah huruf kapital menjadi huruf kecil
3	substring()	Mengambil sebagian isi String
4	compareTo()	Membandingkan 2 buah String (return integer)
5	indexOf()	Mencari Index dari sebuah karakter
6	contains()	Membandingkan 2 buah String (return boolean)

5.5 Manipulasi String (Lanj...)

No	Perintah	Keterangan
7	concat()	Menggabungkan 2 buah String
8	endsWith()	Pengecekan apakah value String diakhiri karakter tertentu
9	startsWith()	Pengecekan apakah value String diawali karakter tertentu
10	length()	Mengetahui panjang karakter dari value string
11	equals()	Membandingkan isi string
12	equalsIgnoreCase()	Membandingkan isi string tanpa membedakan huruf kapital dan bukan kapital.

5.6 Fungsi Matematik

No	Perintah	Keterangan
1	abs()	Untuk menghasilkan nilai asbolut (nilai positif)
2	ceil()	Untuk pembulatan ke atas
3	floor()	Untuk pembulatan ke bawah
4	floorDiv()	Pembulatan ke bawah dengan membagi paramter pertama dengan parameter kedua
5	min()	Untuk memberikan nilai minimal dari dua atau lebih parameter
6	max()	Untuk memberikan nilai maximum dari dua atau lebih parameter

5.6 Fungsi Matematik (Lanj...)

No	Perintah	Keterangan
7	round()	Untuk pembulatan nilai sesuai dengan besarnya suatu parameter
8	random()	Untuk menghasilkan nilai acak dari 0 hingga 1, menjadikan nilai ini tidak terduga
9	sqrt()	Untuk menghasilkan akar dari suatu parameter/variabel
10	pow()	Untuk melakukan pangkat terhadap bilangan
11	sin()	Untuk menghasilkan nilai sinus
12	cos()	Untuk menghasilkan nilai cosinus
13	tan()	Untuk menghasilkan nilai tangen

5.7 Jenis-Jenis Library

Package	Nama Package	Keterangan
Language	java.lang	Berisi library utama/inti dari bahasa Java
Utilities	java.util	Berisi library struktur Java
I/O	java.io	Berisi library input/output
Text	java.text	Berisi library untuk mendukung penanganan teks, tanggal, bilangan, dan message

5.7 Jenis-Jenis Libray (Lanj...)

Package	Nama Package	Keterangan
Math	java.math	Berisi library matematik
AWT	java.awt	Berisi library untuk perancangan user-interface dan event-handling
Swing	java.swing	Untuk membuat berbagai komponen dalam Java yang sama dengan berbagai platform
Javax	javax	Perluasan dari bahasa Java
Applet	java.applet	Class untuk membuat applet

5.7 Jenis-Jenis Libray (Lanj...)

Package	Nama Package	Keterangan
Beans	java.beans	Class untuk membuat Java Beans
Reflection	java.lang.reflect	Class untuk memperoleh informasi run-time
SQL	java.sql	Class untuk mendukung akses dan pengelolaan data dalam database
RMI	java.rmi	Class untuk mendukung distributed programming
Network	java.net	Class untuk mendukung keamanan kriptografi



6. Jenis-Jenis Exception Pada Java

- Jenis kesalahan dalam membuat program dibagi dalam tiga katagori yaitu:
 - syntax error
 - run time error
 - logic error
- Exception dipicu oleh run time error dan digunakan sebagai sarana melaporkan kondisi-kondisi kesalahan.

Lanj...

No	Exception	Keterangan
1	ClassNotFoundException	Kelas tidak ditemukan
2	CloneNotSupportedException	Melakukan clone objek yang tidak mengimplementasikan interface Cloneable
3	IllegalAccessException	Pengaksesan ke kelas ditolak
4	InstantiationException	Menciptakan objek dari kelas abstract ataupun dari interface
5	InterruptedException	Thread telah diinterupsi oleh thread lain
6	NoSuchFieldException	Field yang diminta tidak ada
7	NoSuchMethodException	Method yang diminta tidak ada

Lanj...

No	Exception	Keterangan
1	AritmaticException	Kesalahan aritmetik seperti pembagian dengan nol
2	ArrayIndexOutOfBoundsException	Indeks array di luar batas
3	ArrayStoreException	Pemberian nilai ke elemen array tidak sesuai dengan tipenya
4	ClassCastException	Pemberian nilai ke elemen array tidak sesuai dengan tipenya
5	IllegalArgumentException	Argument illegal

Lanj...

No	Exception	Keterangan
6	IllegalMonitorException	Operasi monitor illegal
7	IllegalStateException	Lingkungan atau aplikasi state yang tidak benar
8	IllegalThreadStateException	Operasi yang diminta tidak kompatibel dengan state thread saat itu
9	IndexOutOfBoundsException	Indeks diluar batas
10	NegativeArraySizeException	Array diciptakan dengan ukuran negatif
11	NullPointerException	Penggunaan null yang tidak sah

Lanj...

No	Exception	Keterangan
12	NumberFormatException	Konversi yang tidak sah dari string ke format numerik
13	SecurityException	Melanggar aturan security
14	StringIndexOutOfBoundsException	Index di luar batas string
15	UnsupportedOperationException	Detemukan operasi yang tidak didukung

Lanj...

No	Method Pesan Exepction	Keterangan
1	getMessage()	Mengembalikan nilai string yang berisi pesan rinci tentang objek Throwable yang mengalami exception
2	toString()	Mengembalikan nilai string yng berisi pesan singkat tentang objek yang mengalami exception
3	getLocalizedMessage()	Menampilkan pesan exception lokal.
4	printStackTrace	Method ini bersifat void, dan hanya mencetak informasi tentang objek Throwable



7. Membuat Program Dalam Tampilan Teks dan GUI

- Bahasa pemrograman java menyediakan tiga class untuk membaca nilai yang diinputkan, yaitu: **Scanner**, **BufferedReader**, **Console**.
- Penggunaan class tersebut digunakan untuk membaca input berbasis teks atau yang dikenal dengan console.
- Bila berbasis GUI maka dapat menggunakan class yang lain seperti: **JOptionPane** dan **inputbox** pada form.

7.1 Dalam Tampilan Teks

- **Penggunaan Scanner**
- Pada latihan ini akan mengimplementasikan input berbasis teks pada program java.
- Berikut program yang mengimplementasikan class Scanner pada java.

7.1 Dalam Tampilan Teks (Lanj...)

Nama Class: clsDokter

```
1. package dokter;
2. public class clsDokter {
3.     //mendeklarasikan atribut/variabel
4.     String IdDokter;
5.     String Nama;
6.     int Gaji;
7.     //membuat method dengan fungsi
8.     public float Tunjangan()
9.     {
10.         return Gaji/100*10;
11.     }
12.     //membuat method dengan fungsi
13.     public float TotalGaji()
14.     {
15.         return Gaji+Tunjangan();
16.     }
17. }
```

7.1 Dalam Tampilan Teks (Lanj...)

Main Program: Dokter

```
1.  package dokter;
2.
3.  import java.util.Scanner;
4.
5.  public class Dokter {
6.      /**
7.       * @param args the command line arguments
8.       */
9.      public static void main(String[] args) {
10.         // TODO code application logic here
11.
12.         //membuat scanner baru
13.         Scanner input = new Scanner(System.in);
14.         //membuat objek
15.         clsDokter objDokter = new clsDokter();
16.
```

7.1 Dalam Tampilan Teks (Lanj...)

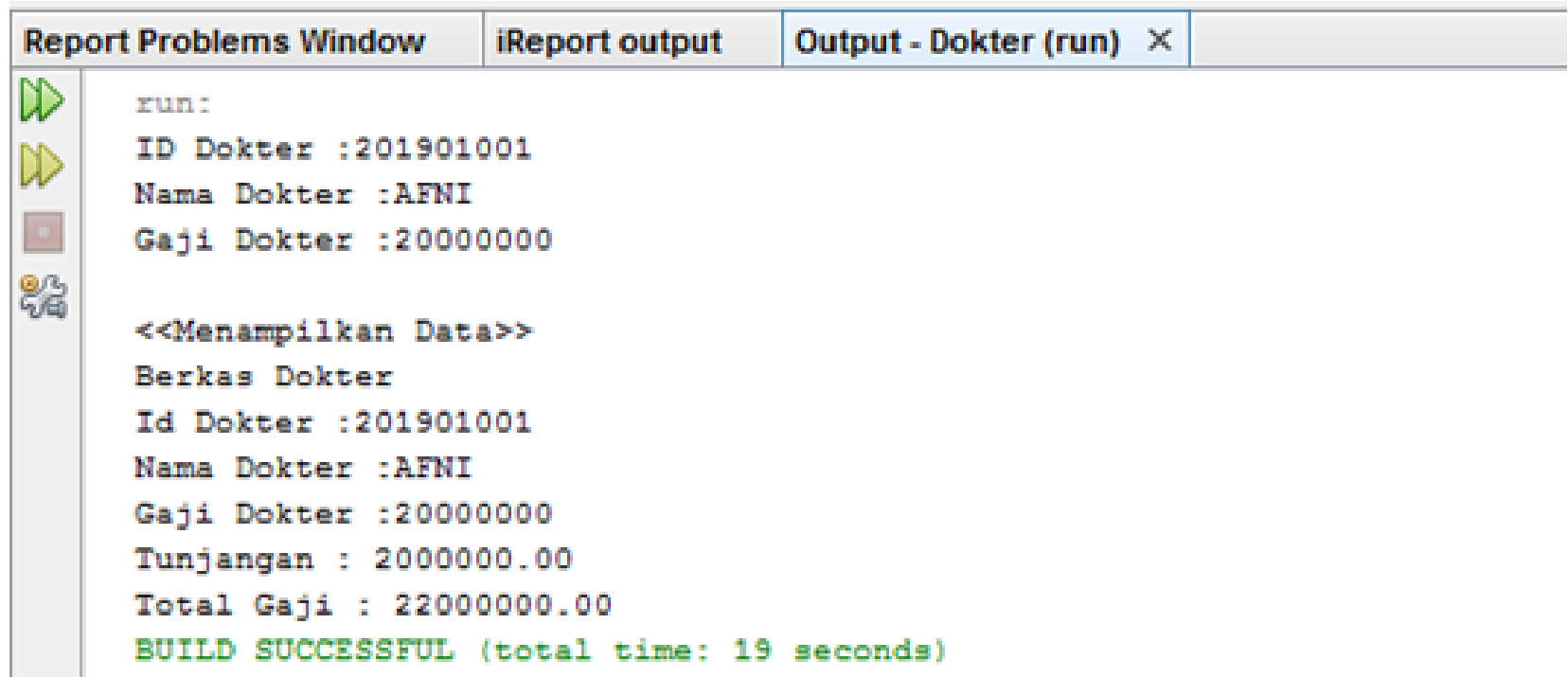
```
17.      //mencatak ke layar
18.      System.out.print("ID Dokter :");
19.      //membaca nilai
20.      objDokter.IdDokter = input.nextLine();
21.      System.out.print("Nama Dokter :");
22.      objDokter>Nama = input.nextLine();
23.      System.out.print("Gaji Dokter :");
24.      objDokter.Gaji = input.nextInt();
25.
26.      //menampilkan data
27.      System.out.println();
28.      System.out.println("<<Menampilkan Data>>");
29.      System.out.println("Berkas Dokter");
30.      System.out.println("Id Dokter
      :"+objDokter.IdDokter);
```

7.1 Dalam Tampilan Teks (Lanj...)

```
31.         System.out.println("Nama Dokter  
        :"+objDokter.Nama);  
32.         System.out.println("Gaji Dokter  
        :"+objDokter.Gaji);  
33.         System.out.printf("Tunjangan : %.2f\n",  
        +objDokter.Tunjangan());  
34.         System.out.printf("Total Gaji : %.2f\n",  
        +objDokter.TotalGaji());  
35.     }  
36. }
```

7.1 Dalam Tampilan Teks (Lanj...)

- Contoh keluaran program.



The screenshot shows a software development environment's output window. It has three tabs: 'Report Problems Window', 'iReport output', and 'Output - Dokter (run)'. The 'Output - Dokter (run)' tab is active. On the left side of the output window, there is a vertical toolbar with icons for running (green play button), stepping through (yellow play button), stopping (red square), and debugging (bug icon). The output text is as follows:

```
run:
ID Dokter :201901001
Nama Dokter :AFNI
Gaji Dokter :20000000

<<Menampilkan Data>>
Berkas Dokter
Id Dokter :201901001
Nama Dokter :AFNI
Gaji Dokter :20000000
Tunjangan : 2000000.00
Total Gaji : 22000000.00
BUILD SUCCESSFUL (total time: 19 seconds)
```

7.1 Dalam Tampilan Teks (Lanj...)

- **Class BufferedReader** tidak hanya untuk membaca nilai dari keyboard.
- Class **BufferedReader** dapat juga digunakan untuk membaca input dari file dan jaringan.
- Bila menggunakan class **BufferedReader** terdapat pada paket `java.io`.
- **Catatan:** silahkan buat sendiri latihannya

7.1 Dalam Tampilan Teks (Lanj...)

- **Class Console** hampir sama dengan menggunakan `BufferedReader`. Pada class `Console` juga menggunakan fungsi `readLine()` untuk membaca input.
- Class `Console` hanya bisa digunakan di lingkungan console seperti terminal. Class `Console` tidak bisa langsung digunakan di Netbeans. Untuk itu dilakukan kompilasi secara manual.

7.1 Dalam Tampilan Teks (Lanj...)

- Cara mengkompilasi secara manual perintahnya: `javac namafile.java`, untuk menjalankannya `java namafile`. Program dapat dibuat menggunakan Notepad++, simpan filenya dengan sintaks: `nama_file.java`.
- **Catatan:** silahkan buat sendiri latihannya

7.2 Dalam Tampilan GUI

- Kali ini akan mengimplementasikan input berbasis GUI pada program java.
- Kita dapat membuat input dialog box untuk memasukkan nilai.
- Memasukkan nilai dari keyboard lewat **JOptionPane.showInputDialog()**
- Contoh program lengkapnya.

7.2 Dalam Tampilan GUI (Lanj...)

Nama Class: clsDokter

```
1. package dokter;  
2.  
3. public class clsDokter {  
4.     //mendeklarasikan atribut  
5.     String IdDokter;  
6.     String Nama;  
7.     int Gaji;  
8.
```

7.2 Dalam Tampilan GUI (Lanj...)

```
9.      //membuat method dengan fungsi
10.     public float Tunjangan()
11.     {
12.         return Gaji/100*10;
13.     }
14.
15.     //membuat method dengan fungsi
16.     public float TotalGaji()
17.     {
18.         return Gaji+Tunjangan();
19.     }
20. }
```

7.2 Dalam Tampilan GUI (Lanj...)

Main Program: Dokter

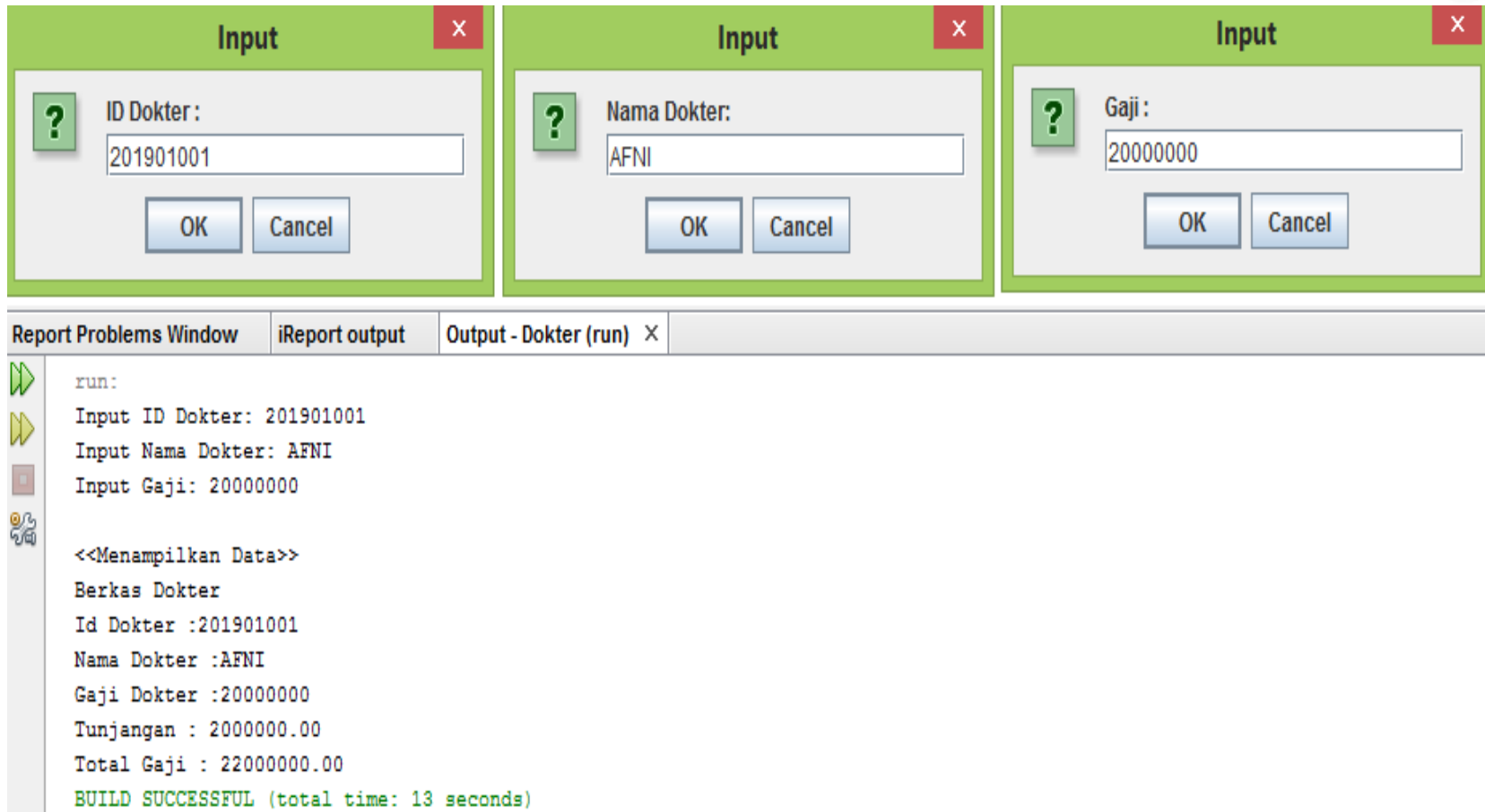
```
1.  package dokter;
2.
3.  import javax.swing.JOptionPane;
4.
5.  public class Dokter {
6.
7.      public static void main(String[] args) {
8.          //membuat objek
9.          clsDokter objDokter = new clsDokter();
10.
11.          //Membuat kotak dialog dengan showInputDialog()
12.          objDokter.IdDokter =
            JOptionPane.showInputDialog("ID Dokter :");
```

7.2 Dalam Tampilan GUI (Lanj...)

```
13.         System.out.println("Input ID Dokter: "+
    objDokter.IdDokter);
14.         objDokter>Nama =
    JOptionPane.showInputDialog("Nama Dokter:");
15.         System.out.println("Input Nama Dokter: "+
    objDokter>Nama);
16.         String mGaji =
    JOptionPane.showInputDialog("Gaji :");
17.         System.out.println("Input Gaji: "+ mGaji);
18.         objDokter.Gaji = Integer.parseInt(mGaji);
19.
20.         //menampilkan data
21.         System.out.println();
22.         System.out.println("<<Menampilkan Data>>");
```

7.2 Dalam Tampilan GUI (Lanj...)

- Contoh keluaran program seperti berikut.



7.2 Dalam Tampilan GUI (Lanj...)

- **showInputDialog()**
 - Memasukkan nilai dari keyboard lewat **JOptionPane.showInputDialog()**
- **showMessageDialog()**
 - Menampilkan data dengan **JOptionPane.showMessageDialog()**

7.2 Dalam Tampilan GUI (Lanj...)

- **showInputDialog():** digunakan untuk menampilkan kotak dialog input.
- **showMessageDialog():** digunakan untuk menampilkan kotak dialog pesan/informasi.
showConfirmDialog(): digunakan untuk menampilkan kotak dialog konfirmasi.
- **showOptionDialog():** digunakan untuk menampilkan kotak dialog pilihan.

Ringkasan:

- Untuk menguasai suatu bahasa pemrograman ada baiknya menguasai struktur bahasa pemrogramannya, tipe data, operator, dan yang lainnya. Pada bahasa pemrograman java juga banyak menggunakan library (pustaka) yang dibutuhkan saat membuat program. Dengan mengetahui struktur bahasa pemrograman Java dan library-nya dapat mempermudah dalam membuat program pada bahasa pemrograman Java.

Latihan Mandiri 1

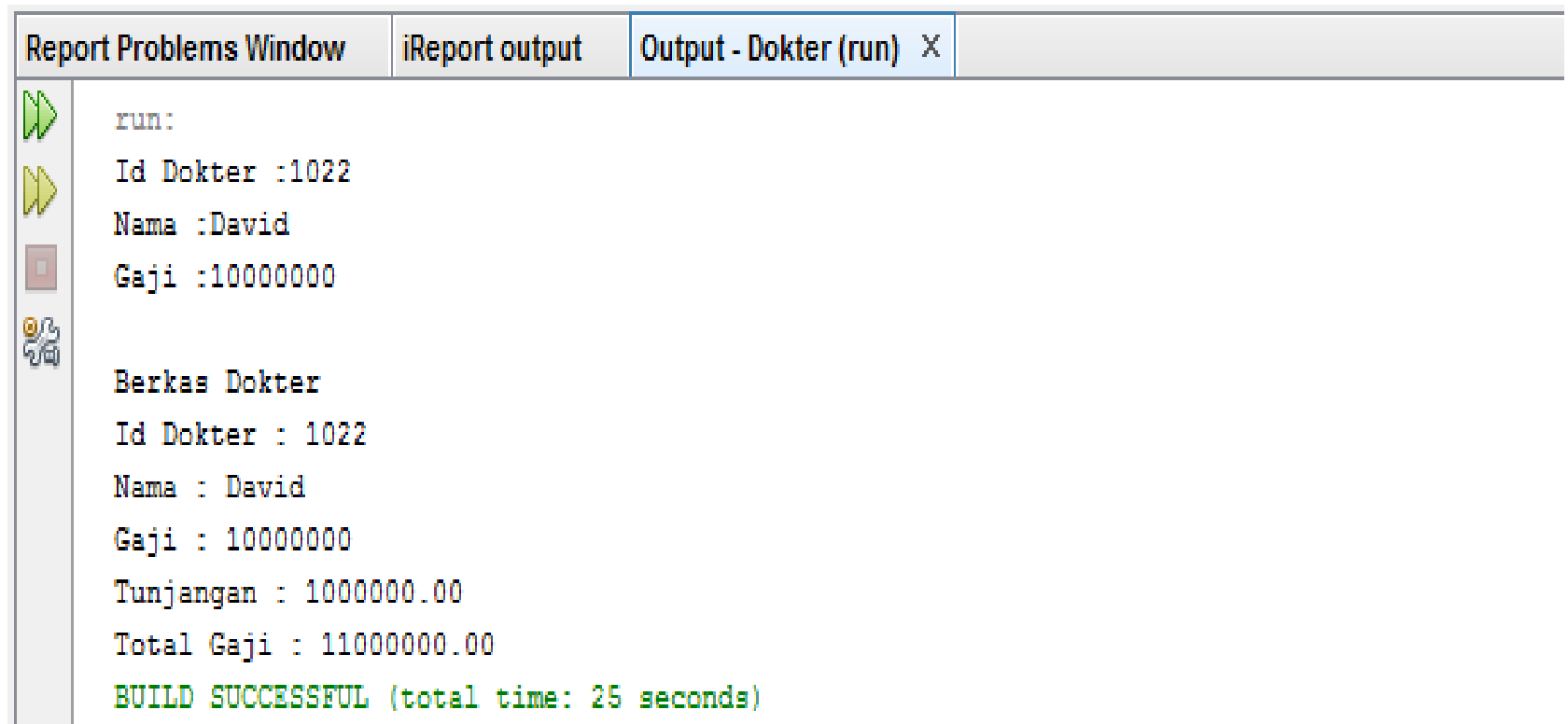
clsDokter
+IdDokter : String +Nama : String +Gaji : int
+ Tunjangan() : float + TotalGaji() : float

- Buatlah program berikut dengan konsep berorientasi objek.
- Kelas / Objek dokter memiliki atribut IdDokter, Nama, Gaji.
- Methodnya adalah
Tunjangan = 10% dari gaji,
Total Gaji = Gaji + Tunjangan.

Id Dokter, Nama dan Gaji dimasukkan dari keyboard.

Latihan Mandiri 1 (Lanj..)

- Contoh keluarah program.

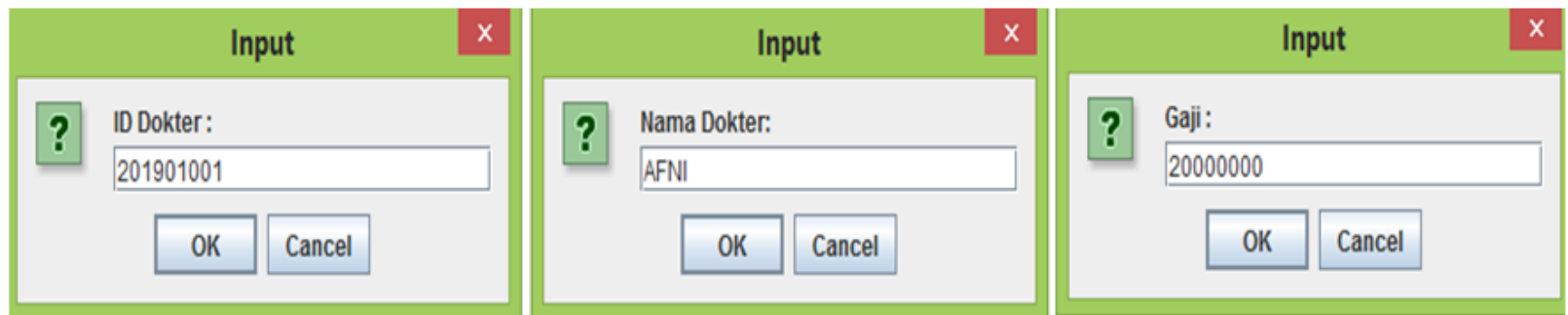


The screenshot shows a window titled "Report Problems Window" with three tabs: "iReport output", "Output - Dokter (run)", and a close button "X". The "Output - Dokter (run)" tab is active, displaying the following text:

```
run:  
Id Dokter :1022  
Nama :David  
Gaji :10000000  
  
Berkas Dokter  
Id Dokter : 1022  
Nama : David  
Gaji : 10000000  
Tunjangan : 1000000.00  
Total Gaji : 11000000.00  
BUILD SUCCESSFUL (total time: 25 seconds)
```

Latihan Mandiri 2

- Analisis tampilan berikut, dan buat programnya dengan menggunakan konsep berorientasi objek. Input data dilakukan dengan kotak dialog.

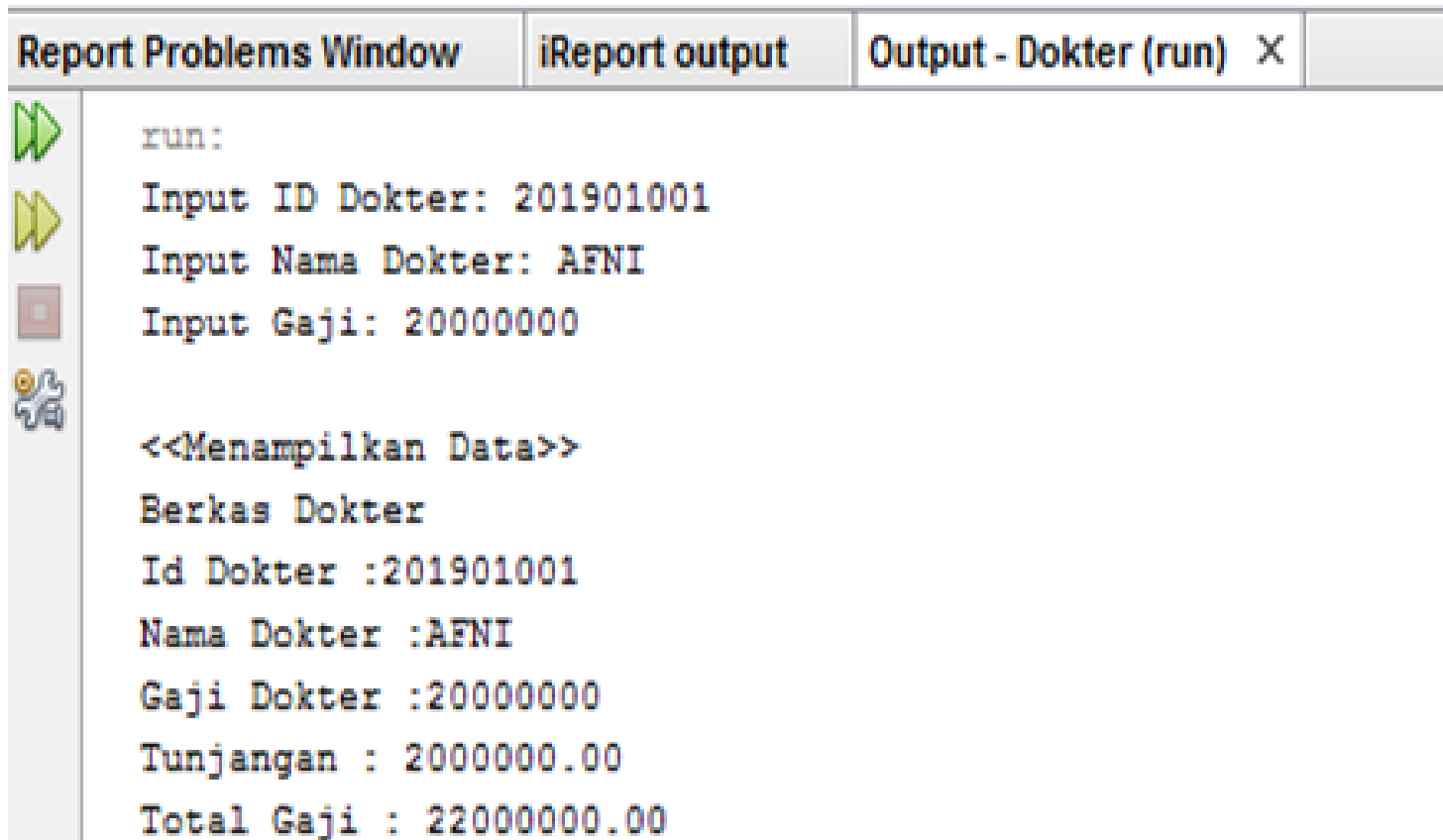


The image displays three separate 'Input' dialog boxes, each with a green title bar and a red close button. Each dialog contains a green question mark icon, a label, a text input field, and 'OK' and 'Cancel' buttons.

- Dialog 1:** Label 'ID Dokter:', input field contains '201901001'.
- Dialog 2:** Label 'Nama Dokter:', input field contains 'AFNI'.
- Dialog 3:** Label 'Gaji:', input field contains '20000000'.

Latihan Mandiri 2 (Lanj..)

- Contoh keluaran program.



The screenshot shows a Java IDE window with three tabs: "Report Problems Window", "iReport output", and "Output - Dokter (run) X". The "Output - Dokter (run)" tab is active, displaying the following text:

```
run:
Input ID Dokter: 201901001
Input Nama Dokter: AFNI
Input Gaji: 20000000

<<Menampilkan Data>>
Berkas Dokter
Id Dokter :201901001
Nama Dokter :AFNI
Gaji Dokter :20000000
Tunjangan : 2000000.00
Total Gaji : 22000000.00
```

Latihan Mandiri 3

- Buatlah sebuah program dengan menerapkan konsep pemrograman berorientasi objek. Dengan input data dilakukan dari kotak dialog (dialog box), contoh kasusnya bebas minimal ada 4 atribut dan ada 3 method.



TERIMA KASIH

U N I V E R S I T A S B U N D A M U L I A