

BELAJAR PEMROGRAMAN JAVA DENGAN NETBEANS

Sebuah Pengantar

Oleh: Muhammad Rusli, I Komang Rinatha, Yohanes Priyo Atmojo.

Hak Cipta ©2016 pada Penulis.

Editor : Putri Christian
Setting : Yulius Basuki
Desain Cover : Dany
Korektor : Aditya Ari C

Hak Cipta dilindungi undang-undang.

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apa pun, baik secara elektronik maupun mekanis, termasuk memfotokopi, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari penulis.

Penerbit CV. ANDI OFFSET (Penerbit ANDI, Anggota IKAPI) Jl. Beo 38-40, telp (0274) 561881, Fax (0274) 588282 Yogyakarta 55281

Percetakan CV. ANDI OFFSET (Penerbit ANDI, Anggota IKAPI) Jl. Beo 38-40, telp (0274) 561881, Fax (0274) 588282 Yogyakarta 55281

Perpustakaan Nasional: Katalog dalam Terbitan

Rusli, Muhammad

BELAJAR PEMROGRAMAN JAVA DENGAN NETBEANS

Sebuah Pengantar / Muhammad Rusli, I Komang Rinatha, Yohanes Priyo Atmojo

- Ed. I - Yogyakarta: ANDI;

25 - 24 - 23 - 22 - 21 - 20 - 19 - 18 - 17 - 16

hlm viii + 120 ; 16 x 23 Cm.

10 9 8 7 6 5 4 3 2 1

ISBN: 978 - 979 - 29 - 5582 - 8

I. Judul

1. Programing Languages
2. Rinatha, I Komang
3. Atmojo, Yohanes Priyo

DDC'23 : 005.133

KATA PENGANTAR

Dengan mengucapkan syukur ke hadirat Tuhan Yang Maha Esa, materi "*Belajar Pemrograman Java dengan NetBeans: Sebuah pengantar*" telah dapat kami selesaikan.

Buku ini dimaksudkan bagi pemula yang ingin belajar Pemrograman Java. Memerhatikan tipe konten Pemrograman Java yang bersifat prinsip dan memiliki level abstraksi yang cukup tinggi, sebagai pemula tentunya belajar Pemrograman Java tidaklah mudah atau bahkan bagi sebagian pembelajar menganggapnya cukup sulit. Untuk itu, materi disajikan/didesain bersifat tutorial (dengan beberapa contoh aplikasi) sehingga diharapkan dapat meningkatkan motivasi pembelajar/mahasiswa dalam belajar Pemrograman Java menuju kompetensi/capaian pembelajaran yang diinginkan, yaitu agar pembelajar/mahasiswa mampu menerapkan konsep, prosedur, dan prinsip dasar Pemrograman Java.

Semoga buku ini bermanfaat bagi pembaca. Kritik dan saran yang membangun sangat kami harapkan. Terima kasih atas segala perhatiannya.

Denpasar, Juni 2016

Penyusun

KATA PENGANTAR.....III

DAFTAR ISI..... V

BAB 1 MENGENAL KONSEP PEMROGRAMAN JAVA... 1

- A. KONSEP PEMROGRAMAN KOMPUTER 1
- B. KONSEP PEMROGRAMAN JAVA..... 2
- C. INSTALASI PERANGKAT PENGEMBANGAN..... 3
- D. RINGKASAN 4
- E. SOAL/LATIHAN..... 5

BAB 2 BELAJAR PROGRAM PERTAMA KALI7

- A. MENULIS, MENYIMPAN, DAN MENJALANKAN PROGRAM.. 7
- B. PROGRAM JAVA BEKERJA..... 10
- C. RINGKASAN 16
- D. SOAL/LATIHAN..... 17

BAB 3 BELAJAR MENYIMPAN, MENGUBAH, DAN MENAMPILKAN INFORMASI 19

- A. MENYIMPAN DATA/INFORMASI 19
- B. MENGUBAH DATA/INFORMASI 23
- C. MENAMPILKAN DATA/INFORMASI 28
- D. MEMASUKKAN DATA/INFORMASI LEWAT KEYBOARD 33

E. RINGKASAN.....	36
F. SOAL/LATIHAN	37
BAB 4 MEMPROGRAM DENGAN STATEMEN	
KONDISIONAL	41
A. KONSEP-KONSEP YANG RELEVAN	41
B. STATEMEN IF	42
C. STATEMEN IF-ELSE	44
D. STATEMEN NESTED-IF	45
E. STATEMEN SWITCH	50
F. RINGKASAN	54
G. SOAL/LATIHAN	55
BAB 5 MEMPROGRAM DENGAN STATEMENT	
REPETISI (LOOP)	59
A. STATEMENT FOR.....	59
B. STATEMENT WHILE	62
C. STATEMENT DO-WHILE	65
D. RINGKASAN.....	69
E. SOAL/LATIHAN.....	70
BAB 6 MENYIMPAN INFORMASI DENGAN ARRAY	75
A. MEMBANGUN DAN MENGGUNAKAN VARIABEL	
ARRAY 1 DIMENSI	75
B. MEMBANGUN DAN MENGGUNAKAN VARIABEL ARRAY	
MULTIDIMENSI.....	79
C. MENGURUTKAN ARRAY.....	84
D. MENGGUNAKAN METODE STRING	86
E. RINGKASAN.....	89
F. SOAL/LATIHAN	90

BAB 7 MEMBANGUN METODE KLAS	93
A. MENGENAL KONSEP KLAS/OBJEK.....	93
B. MENGENAL METODE KLAS YANG SUDAH	
TERDEFINISIKAN	95
C. MEMBANGUN METODE KLAS	98
D. RINGKASAN.....	107
E. SOAL/LATIHAN.....	108
DAFTAR PUSTAKA	113
TENTANG PENULIS.....	115

BAB 1

MENGENAL KONSEP PEMROGRAMAN JAVA

A. KONSEP PEMROGRAMAN KOMPUTER

Dalam berkomunikasi dengan seseorang, tentunya kita perlu mengetahui beberapa hal berikut:

1. Apakah orang tersebut memiliki bahasa yang sama?
2. Apakah konten yang disampaikan dapat dimengerti?

Kedua hal tersebut tentunya penting kita perhatikan agar komunikasi kita berjalan efektif. Dengan syarat pertama, kemungkinan komunikasi masih bisa berjalan, tetapi pihak penerima pesan/materi/konten belum tentu dapat mengerti sepenuhnya maksud penyampaian pesan/materi/konten tersebut (belum efektif, karena masih dimungkinkan terjadi miskomunikasi). Untuk dapat memenuhi syarat kedua, tentunya harus ada aturan baku tentang bahasa tersebut yang dipenuhi oleh kedua belah pihak sehingga komunikasi berjalan efektif.

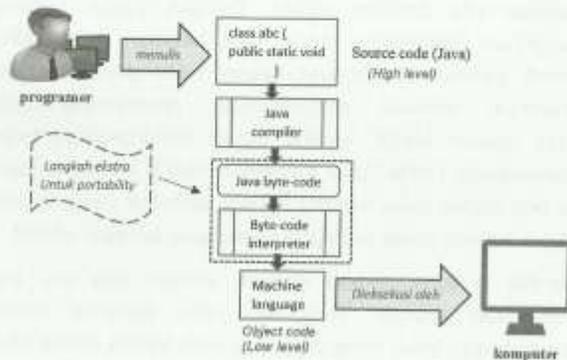
Kondisi di atas serupa dengan kondisi bila kita ingin berkomunikasi dengan komputer, yaitu perlunya bahasa yang sama dan baku yang dipahami oleh kedua belah pihak. Pihak pertama (penyampai) disebut pemrogram komputer (*programmer*), yang menyusun/menulis konten/instruksi/

statemen yang akan diberikan/dimasukkan pada komputer sebagai pihak kedua (penerima) untuk dilaksanakan/dieksekusi. Konten/kumpulan instruksi/statemen tersebut disebut program komputer. Aktivitas pemrogram komputer (*programmer*) membuat program komputer disebut pemrograman komputer.

B. KONSEP PEMROGRAMAN JAVA

Dalam konteks pemrograman komputer, terdapat sejumlah bahasa pemrograman komputer, seperti Pascal, C++, Visual Basic, dan Java. Buku ini hanya membahas tentang bahasa pemrograman Java.

Java adalah bahasa pemrograman yang tergolong pada *high level language* (mudah bagi manusia untuk memahami), mengingat kata-kata/statemennya menyerupai bahasa manusia (*english*). Namun demikian, dalam penulisannya memerlukan aturan (*syntax*) yang ketat.



Gambar 1. Skema Pemrograman Java

Gambar 1 menunjukkan aktivitas pemrograman mulai dari programmer menulis program melalui editor (salah satunya adalah NetBeans) dan kemudian disimpan berupa file (disebut sebagai *source code*, dengan format namafile.java), dan selanjutnya program dapat dijalankan (*run*). Dalam Java, agar bisa dijalankan, *source code* tersebut memerlukan verifikasi/kompilasi terlebih dahulu melalui *Java compiler* menjadi *Java byte-code*, kemudian diterjemahkan ke dalam bahasa mesin (*machine language/object code*) oleh *interpreter*. Bahasa mesin (*object code*) hasil verifikasi/kompilasi *source code* selanjutnya siap dieksekusi oleh komputer.

Sementara itu, Java memiliki kemampuan portabilitas (*portability*). Portabilitas mengacu pada kemampuan menjalankan (*run*) program pada mesin yang berbeda. Untuk itu, ketika kita mendengar seseorang menyatakan bahwa aplikasi Java dan *Applet portable*, maka maksudnya adalah aplikasi tersebut dapat berjalan pada tipe mesin yang berbeda dengan tanpa perubahan (misal tanpa rekompilasi, atau penyesuaian pada *source code*).

C. INSTALASI PERANGKAT PENGEMBANGAN

Sebelum kita melakukan instalasi perangkat pengembangan, kita memerlukan *download* file-file sistem yang relevan, yaitu Java installer, dan NetBeans installer. Netbeans merupakan software teks editor sebagai tempat menuliskan dan mengeksekusi program Java). File-file tersebut (berjalan di sistem operasi Windows, dan dipakai dalam contoh-contoh di buku ini) adalah: Java jdk.8.0_45 dan NetBeans 8.0.2. Namun Anda dapat juga men-*download* Java installer versi lain yang masih bisa mengakomodasi contoh-contoh program yang diberikan di buku ini, demikian juga untuk teks editor/NetBeans-nya).

BAB 2

BELAJAR PROGRAM PERTAMA KALI

A. MENULIS, MENYIMPAN, DAN MENJALANKAN PROGRAM

Untuk menulis sebuah program Java diperlukan sebuah perangkat pengembangan yang mendukung *Java Development Kit* (JDK). Salah satu perangkat pendukung tersebut disebut NetBeans IDE (*Integrated Development Environment*) dengan NetBeans (sebagai *text editor* atau *source code text editor*), kita dapat menulis/mengubah/menyimpan/membuka program, mengompilasi (*compiling*), dan menjalankan (*running*) program Java.

Program Java yang ditulis melalui NetBeans (disebut sebagai *source code*) merupakan file teks (*text files*) yang sederhana, tanpa adanya format khusus, seperti *centered text* atau *boldface text*. Namun demikian NetBeans mampu mengidentifikasi elemen-elemen yang berbeda dari Java yang ditulis melalui perbedaan warna, juga menakuk (*indent*) baris-baris statemen program dengan baik dan menyediakan fasilitas dokumentasi program yang berguna.

Membangun Program javaClass01

Bagaimana membangun program Java yang bila dieksekusi (*save* dan *run*) akan menampilkan "Selamat datang di pemrograman Java!"

Target program:

```
1 public class javaClass01 {
2     public static void main(String[] args) {
3         String selamat = "Selamat datang di Pemrograman Java!";
4         System.out.println(selamat);
5     }
6 }
```

Output program:

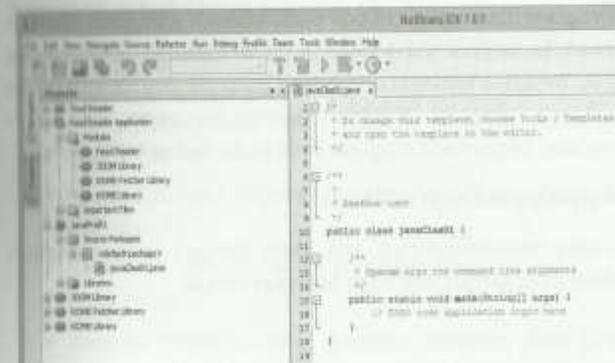
```
Output - JavaPro01 (run)
run:
Selamat datang di Pemrograman Java!
BUILD SUCCESSFUL (total time: 0 seconds)
```

Langkah-langkahnya sebagai berikut:

- Jalankan NetBeans (double click icon NetBeans)
- Pilih menu File, New Project.
- Pilih category project Java, tipe project Java Application, selanjutnya click Next.
- Masukkan Projek Name dengan javaPro01.
- Jika nama projek sudah ada, maka muncul error seperti berikut:

Project Folder already exists and is not empty.

- Pilih folder untuk Project Location dengan melakukan Browse.
- Masukkan Create Main Class dengan javaClass01.
- Click Finish.
- Project javaPro01 muncul di jendela Project bersama program Java javaClass01.java (Gambar 2).
- Selanjutnya pada program javaClass01.java, delete semua baris yang memuat komentar atau keterangan dimulai tanda */** dan berakhir **/*, baris dengan tanda *//* dan baris-baris kosong (*blank*).
- Insert 2 baris kosong setelah baris 2. Isi/ketiklah statemen baris 3 dan 4 sebagaimana statemen baris 3 dan 4 pada target program.
- Save Project dengan menekan tombol save all files () selanjutnya Run Project (tombol F6) dengan hasil sebagaimana output program.



Gambar 2. Jendela Project JavaPro1 dan Program javaClass01.java

- a. Kesalahan sintaksis.
 - b. Kesalahan logika.
 - c. Kesalahan debugging.
4. Metode utama (*main method*) dalam sebuah program Java dicirikan dengan adanya statement....
 - a. `public void main(String [] arguments).`
 - b. `public static void main(String [] arguments).`
 - c. `public void main(String [] arguments).`
 5. Proses memperbaiki *error* disebut....
 - a. *Decomposing.*
 - b. *Defrosting.*
 - c. *Debugging.*

II. Selesaikan soal berikut.

1. Sebutkan semua kata kunci yang telah dipelajari sampai dengan seksi ini. Deskripsikan masing-masing fungsinya.
2. Identifikasi dan perbaiki kesalahan dalam program sederhana berikut:

```
public class Mampu {
    Public void main(String[] args) {
        integer x = 5;
        System.out.println(Mampukah anda memprogram
        komputer?");
        Println("Bilangan integer = ", x)
    }
}
```

3. Buat sebuah program yang menampilkan nama dan id (nim/ktp) Anda.

BAB 3

BELAJAR MENYIMPAN, MENGUBAH, DAN MENAMPILKAN INFORMASI

A. MENYIMPAN DATA/INFORMASI

Pada pembelajaran sebelumnya, kita menjumpai istilah/konsep yang disebut "variabel". Variabel di sini berfungsi menyimpan data/informasi. Informasi yang disimpan dalam sebuah variabel dapat berubah ketika program dijalankan.

Sebelum kita sampai pada pembahasan bagaimana menyimpan data/informasi, marilah kita mengenal lebih dahulu tentang konsep-konsep yang berhubungan dengan variabel, yaitu aturan penamaan sebuah variabel dan tipe variabel.

Konsep-konsep yang berhubungan dengan variabel penamaan sebuah variabel

Nama variabel (atau *identifier*) dalam Java dapat dibentuk dari gabungan huruf alphabet (A-Z) atau (a-z), digit (0-9), dan karakter (_) dan (\$). Namun demikian digit pertama dari nama variabel harus berupa huruf alphabet. Nama-nama yang bertindak sebagai *reserved word* atau terpakai sebagai kata-kata standard (*class*) yang digunakan dalam pemrograman tidak diperbolehkan misal: `int`, `if`, `public`, `void`). Penamaan variabel dalam Java harus konsisten, mengingat Java bersifat *case-sensitive*.

Contoh-contoh nama variabel yang diperbolehkan:

```
num1, nomer, name_depan, namaAkhir, jumlahMahasiswa,  
noRekening, PROGRAMKU
```

Contoh-contoh nama variabel yang tidak diperbolehkan:

```
3rdValue, program ku, this, for
```

Tipe variabel (data)

Setiap variabel dapat menyimpan/menampung data. Terdapat 2 tipe data yang dapat disimpan/ditampung oleh variabel, yaitu tipe data *primitive* atau *reference (class)*. Pembahasan di sini dibatasi pada tipe data *primitive* (numeric, boolean, char, dan String).

Tipe data primitif numerik

Variabel dengan tipe data numerik digunakan untuk menyimpan bilangan. Tipe data numerik dideskripsikan sebagai berikut:

Tabel 1. Tipe Data

Tipe data	Tipe bilangan	Memori terpakai	Jangkauan nilai
Byte	integer	1 byte	-128 s.d 127
Short	integer	2 byte	-32768 s.d 32767
int	integer	4 byte	-2147483648 s.d 2147483647
Long	integer	8 byte	-9223372036854775808 s.d 9223372036854775807
Float	Rill	4 byte	$-\pm 1,40 \times 10^{-45}$ s.d $\pm 3,40 \times 10^{38}$
Double	Rill	8 byte	$-\pm 4,94,4 \times 10^{-324}$ s.d $\pm 1,76 \times 10^{308}$

Tipe data boolean

Variabel dengan tipe data *boolean* digunakan untuk menyimpan nilai *true* atau *false*. Umumnya tipe data ini digunakan untuk menyimpan/menampung jawaban atas pertanyaan atau

kuis benar/salah. Contoh berikut beberapa pertanyaan yang jawabannya *true/false* yang disimpan dalam variabel boolean.

- Sudahkah permainan selesai?
- Sudahkah user menekan tombol?

Tipe data char dan string

Variabel dengan tipe data *char* atau *string* untuk menyimpan data teks. Variabel dengan tipe *char* hanya dapat menyimpan sebuah huruf/alphabet, bilangan, tanda baca, atau simbol. Variabel dengan tipe *string* dapat menyimpan sekelompok karakter.

Bagaimana menyimpan data/informasi

Untuk mengetahui bagaimana menyimpan data/informasi, kita perlu juga mempelajari konsep-konsep yang berhubungan, antara lain *statement* (program/instruksi), deklarasi variabel, penugasan, operator, dan ekspresi. Contoh-contoh program (fragmen) berikut mendeskripsikan konsep-konsep tersebut dan relevansinya dengan penyimpanan data/informasi sebuah variabel.

Contoh 1

```
1 public static void main(String[] args) {  
2     int a, b, d;  
3     a = 5;  
4     b = 10;  
5     int c = 15;  
6     d = a * b * c;  
7 }
```

3. Buat sebuah program yang meminta *user* memasukkan dua buah bilangan integer, kemudian:
 - a. Dapat dan tampilkan nilai absolut kedua bilangan tersebut.
 - b. Tentukan/tampilkan nilai absolut terbesar di antaranya.
 - c. Dapat dan tampilkan akar dari hasil (b).
 - d. Dapat dan tampilkan sisa hasil bagi kedua bilangan hasil (a).

BAB 4

MEMPROGRAM DENGAN STATEMEN KONDISIONAL

Pembahasan berikut ini akan memfokuskan pada bagaimana komputer memproses program komputer yang memuat *statement* kondisional melalui statemen *if*, *if-else*, *nested-if*, dan *switch*. Namun demikian, sebelum membahas hal tersebut, akan diperkenalkan terlebih dahulu konsep-konsep yang relevan dengan *statement* kondisional.

A. KONSEP-KONSEP YANG RELEVAN

Eksresi boolean

- 1. Sebuah ekspresi atau *statement* yang bisa bernilai benar atau salah.
- 2. Biasanya menggunakan operator perbandingan (operator relasi).

Operator relasi

Bahasa Pemrograman	Nama	Notasi Java	Contoh Java
1	Sama dengan	==	Saldo == 0 Respons == 'Y'
2	Tidak sama dengan	!=	Honor != Gaji Respons != 'Y'
3	Lebih kecil dari	<	Honor < Gaji

<	Lebih kecil dari atau sama dengan	<=	Honor <= Gaji
>	Lebih besar dari	>	Honor > Insentif
>=	Lebih besar dari atau sama dengan	>=	Honor >= Insentif

Tabel kebenaran

Andaikan A dan B adalah ekspresi boolean, maka tabel kebenaran untuk operator logika and (&&), or (||), dan not (!) adalah sebagai berikut:

A	B	A && B	A B	!A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

B. STATEMEN IF

Statemen If digunakan untuk menyeleksi kondisi benar/*true* atau salah/*false*. Aksi akan diambil jika kondisi bernilai *true*. Perhatikan contoh program berikut:

Seleksi bilangan integer sebagai persyaratan untuk bisa menghitung akar sebuah bilangan.

Target program:

```

1 import java.util.*;
2 class seleksiIf {
3     public static void main(String[] args) {
4         Scanner masukan = new Scanner(System.in);
5         System.out.print("Masukan bilangan: ");
6         int bil = masukan.nextInt();
7         if (bil >= 0)
8             System.out.println("Akar * " + bil + " = " + Math.sqrt(bil));
9         System.out.println("Bilangan terseleksi..");
10    }
11 }

```

Output program jika kondisi *true* (bilangan positif atau nol)

```

Output X
JavaPro01 (run) X JavaPro01 (run) #2 X
run:
Enter an integer: 25
Akar 25 = 5.0
Bilangan terseleksi..
BUILD SUCCESSFUL (total time: 11 seconds)

```

Output program jika kondisi *false* (bilangan negatif)

```

Output X
JavaPro01 (run) X JavaPro01 (run) #2 X
run:
Enter an integer: -16
Bilangan terseleksi..
BUILD SUCCESSFUL (total time: 10 seconds)

```

Deskripsi:

- Baris 5-6 : menampilkan teks dan memasukkan nilai bilangan integer (bil).
- Baris 7-8 : menyeleksi apakah nilai bil positif atau nol. Jika *true*, baris 8 dieksekusi, menampilkan teks dan hasil akar bil (menggunakan metode *sqrt* dari class *Math*).
- Baris 9 : dieksekusi dengan menampilkan teks "Bilangan terseleksi.." baik baris 7 dieksekusi bernilai *true* maupun *false*.

berikan respons di layar 'tebakan Anda lebih besar', dan jika yang dimasukkan = x, maka berikan respons di layar 'tebakan Anda tepat'.

5. Buatlah sebuah program yang menggunakan *statement switch*. Data *input* yang dimasukkan oleh pengguna (via screen) adalah jumlah anak yang dilahirkan (1–3). Jika jumlah anak 1, tampilkan di screen 'Selamat'; jika jumlah anak 2, tampilkan 'Ou, kembar...selamat ya'; jika jumlah anak 3, tampilkan 'ou, kembar tiga, hebat'; jika selainnya, tampilkan 'wah, benarkah?'.

BAB 5

MEMPROGRAM DENGAN STATEMENT REPETISI (LOOP)

Pemahkah ketika Anda sedang mengikuti pelajaran di sekolah dasar, lalu Anda atau teman Anda melakukan kesalahan atau tidak mengerjakan sebuah tugas, kemudian bapak/ibu guru akan menghukum dengan mengharuskan Anda/teman Anda menulis sebuah kalimat/pernyataan beberapa kali (bisa 100 atau 200 kali) di buku/papan tulis? Jika pernah, tentunya hal itu akan membosankan. Namun tidaklah demikian jika tugas pengulangan/repetisi (*loop*) tersebut dilakukan oleh komputer (program komputer).

Pada pemrograman komputer, *loop* adalah sebuah *statement* atau blok *statement* yang eksekusinya diulang beberapa kali. Pengulangannya bisa tertentu atau tidak tertentu. Dalam pemrograman Java, terdapat tiga macam repetisi, yaitu *for*, *while*, dan *do-while* dengan masing-masing bekerja secara yang berbeda.

A. STATEMENT FOR

Statement for digunakan bila pengulangan/repetisi yang diinginkan atas *statement* atau blok *statement* jumlahnya tertentu (misal 10 kali). Beberapa contoh program berikut menggunakan *loop for*.

1. Menghitung total dan rerata dari sejumlah data

Target program:

```

1  import java.util.*;
2  class ProgramBerkas {
3  public static void main(String[] args) {
4  Scanner scanner = new Scanner(System.in);
5  System.out.println("Masukkan jumlah data yang akan diinput:");
6  System.out.println("-----");
7  int data = scanner.nextInt();
8  int total = 0;
9  System.out.println("Masukkan data:");
10 int i = 0;
11 for (i = 0; i < data; i++) {
12 System.out.print("Data ke- " + (i + 1) + " = ");
13 data = scanner.nextInt();
14 total += data;
15 }
16 System.out.println("-----");
17 System.out.println("Repetisi data : " + data);
18 System.out.println("Total : " + total);
19 System.out.println("Rerata : " + (total / data));
20 }
21 }

```

Output program:

```

Output
JavaPro01 (run) x JavaPro01 (run) #2 x
run:
** Program menghitung Jumlah & Rerata **

--> Banyaknya data? 4
Data ke- 1 = 10
Data ke- 2 = 11
Data ke- 3 = 12
Data ke- 4 = 13

Repetisi data : 4
Total : 46.0
Rerata : 11.50
BUILD SUCCESSFUL (total time: 20 seconds)

```

Deskripsi:

- Baris 7 : deklarasi variabel data dan inisiasi variabel total.
- Baris 8-9 : menampilkan teks dan memasukkan nilai variabel data.
- Baris 10 : statemen for (mulai [i = 1]; jangkauan [i < x]; kenaikan i [i = i + 1]).

Baris 11-13: blok repetisi sebanyak x kali → menampilkan teks dan input nilai data. Penjumlahan nilai total = total + data.

Baris 15 : menghitung nilai rerata.

Baris 17-19: menampilkan teks, jumlah repetisi, jumlah data, dan rerata. Baris 19 memuat format print untuk variabel rerata (tipe double/float) dengan 2 desimal.

2. Menghitung dan menampilkan kelipatan bilangan

Target program:

```

1 class Kelipatan {
2 public static void main(String[] args) {
3 int kelipatan;
4 for (int dex = 1; dex <= 100; dex++) {
5 kelipatan = 1000 + 5 * dex;
6 if ((dex%12) == 0)
7 System.out.println();
8 else
9 System.out.print(kelipatan + " ");
10 }
11 }

```

Output program:

```

Output
JavaPro01 (run) x JavaPro01 (run) #2 x
run:
1005 1010 1015 1020 1025 1030 1035 1040 1045 1050 1055
1060 1070 1075 1080 1085 1090 1095 1100 1105 1110 1115
1120 1130 1135 1140 1145 1150 1155 1160 1165 1170 1175
1180 1190 1195 1200 1205 1210 1215 1220 1225 1230 1235
1240 1250 1255 1260 1265 1270 1275 1280 1285 1290 1295
1300 1310 1315 1320 1325 1330 1335 1340 1345 1350 1355
1360 1370 1375 1380 1385 1390 1395 1400 1405 1410 1415
1420 1430 1435 1440 1445 1450 1455 1460 1465 1470 1475
1480 1490 1495 1500
BUILD SUCCESSFUL (total time: 0 seconds)

```

Deskripsi:

- Baris 3 : deklarasi variabel kelipatan.

```

double[] y;
for (int i=0; i < y.length; i++) {
    y[i] = Math.random * 100; System.out.print("** + y(i)); }
}

```

3. Buat sebuah program yang membaca skor mahasiswa, yang mencari skor terbaik, dan yang menugaskan nilai berdasarkan skema berikut:

Nilai A jika skor \geq Terbaik - 10;
 Nilai B jika skor \geq Terbaik - 20;
 Nilai C jika skor \geq Terbaik - 30;
 Nilai D jika skor \geq Terbaik - 40;
 Nilai F jika skor \geq Terbaik - 50;

Pengguna diminta memasukkan jumlah total mahasiswa, semua skor, dan kemudian menyimpulkan nilai berdasarkan skema di atas.

4. Buat sebuah program yang membaca sepuluh bilangan riil dan kemudian menampilkan sepuluh bilangan riil tersebut dengan urutan terbalik.
5. Buat sebuah program yang dapat menerima *input* sebuah nama dan menampilkan hasilnya secara terbalik.
6. Buat sebuah program guna menyimpan dan menampilkan nilai-nilai sebuah tabel dengan menggunakan array 2 dimensi. Catatan: nilai-nilai yang disimpan bertipe integer.
7. Buatlah sebuah program yang dapat meng-*input* 2 matriks bujur sangkar (tipe NxN), selanjutnya mengoperasikan kedua matriks tersebut (penjumlahan atau pengurangan), dan menampilkan hasilnya. Catatan: Gunakan sistem menu dengan *statement* switch.

BAB 7

MEMBANGUN METODE KLAS

Pada pembahasan materi sebelumnya, sebetulnya kita sudah sedikit mengenal istilah kelas dan metodenya, seperti pada konsep Kelas array dengan implementasi metode `length()`, begitu juga kelas String dengan metode-metodenya antara lain: `length()`, `toUpperCase()`, dan `toLowerCase()`. Pembahasan materi berikut ini akan diawali dengan mengenal konsep Objek/Klas, mengenal metode yang sudah terdefiniskan dari Klas, dan metode klas yang dapat dibangun sendiri.

A. MENGENAL KONSEP KLAS/OBJEK

Klas dapat dikatakan sebagai himpunan/wadah (atau *template*) objek-objek yang memiliki karakteristik yang sama dengan karakteristik Klasnya. Sebuah objek dapat dikatakan sebagai instan (*instance*) Klas. Sebagai contoh Klas Manusia memiliki karakteristik nama, tanggal lahir, jenis kelamin. Di samping itu Klas Manusia juga memiliki perilaku/kemampuan misalnya berjalan, bernapas, makan, minum, berbicara.

Terkait dengan konsep Klas, karakteristik nama, tanggal lahir, dan jenis kelamin disebut atribut, sementara perilaku/kemampuan berjalan, bernapas, makan, minum, berbicara disebut metode. Apabila atribut-atribut Klas manusia, yaitu nama diisi Agus,

tanggal lahir diisi 15-10-2000, dan jenis kelamin diisi pria, maka Agus (bersama dengan nilai atribut yang lain) tersebut disebut sebuah objek dari Kelas manusia. Tentunya metode/perilaku Kelas manusia (berjalan, bernapas, dan lain-lain) dimiliki juga oleh objek manusia yang bernama Agus.

Sebagai ilustrasi, berikut digambarkan klas manusia dan beberapa objeknya.



Tampak dalam gambar tersebut, kesamaan Kelas dan Objek adalah keduanya memiliki atribut dan metode. Namun Kelas berbeda dengan Objek, karena pada Objek, atributnya memiliki nilai (misal atribut nama memiliki nilai/isian, yaitu Budi).

Dalam konteks yang lebih riil/praktis, contoh Kelas/Objek lain yang sering kita jumpai di sekitar kita adalah keberadaan *form-form* isian yang terkadang suatu saat kita juga perlu/harus mengisinya (*form* biodata, pemesanan tiket, pajak, member sebuah organisasi/komunitas, dan lain-lain). Pada saat *form* tersebut masih kosong, dapatlah kita sebut sebagai kelas karena hanya memuat atribut-atribut yang nilainya masih kosong (yang nantinya perlu/harus diisi). Ketika *form* tersebut sudah diisi, maka *form* terisi tersebut dapat disebut objek. Kumpulan *form*

yang sudah terisi dapat disebut kumpulan objek (tentunya objek yang satu berbeda dengan yang lain). Adapun metodenya dapat berupa aksi, minimum kemampuan menghitung jumlah objek (*form* terisi) yang ada.

Dalam konteks pemrograman Java, sebuah objek dibangun melalui perintah `new` atas Klasnya, sebagai contoh di bawah ini:

`String nama = new String("Soebowo");` → membangun objek nama bertipe String bernilai Soebowo

`int[] Harga = new int[10];` → membangun objek Harga bertipe array integer dengan 10 elemen

`int ukuran = Harga.length();` → menggunakan metode `length` dari objek/klas Harga, ukuran = 10

Klas dalam Java, ada yang sudah terdefiniskan (*predefined*) dan ada yang belum, sehingga kita membuat/membangunnya (*create*) sendiri sesuai keperluan (dibahas pada sesi berikutnya).

B. MENGENAL METODE KLAS YANG SUDAH TERDEFINISIKAN

Klas/objek memiliki atribut dan metode. Metode Klas/Objek ada yang memiliki parameter dan ada yang tidak. Tabel berikut merupakan tabel yang memuat klas `Math` (matematika) dan metodenya (memiliki parameter/argumen) yang berhubungan dengan operasi aritmatika.

DAFTAR PUSTAKA

- Cadenhead, R. 2012. *Sams Teach Yourself Java™ in 24 Hours*. Sixth Edition. USA: Sams Publishing.
- Eck, D.J. 2011. *Introduction to Programming Using Java*. <http://math.hws.edu/javanotes>
- Johnson, R.A. 2007. *Introduction to Java Programming and Object-Oriented Application Development*. Boston: Thomson Course Technology.
- Kadir, A. 2012. *Algoritma & Pemrograman Menggunakan Java*. Yogyakarta: Penerbit Andi.
- Lim, S. 2009. *Java Fundamental*. Lecturing items of Dual-Degree Program. STMIK-STIKOM Bali Denpasar and HELP University College Kualalumpur.
- Sianipar, R. H. 2013. *Teori dan Implementasi Java*. Bandung: Penerbit Informatika.