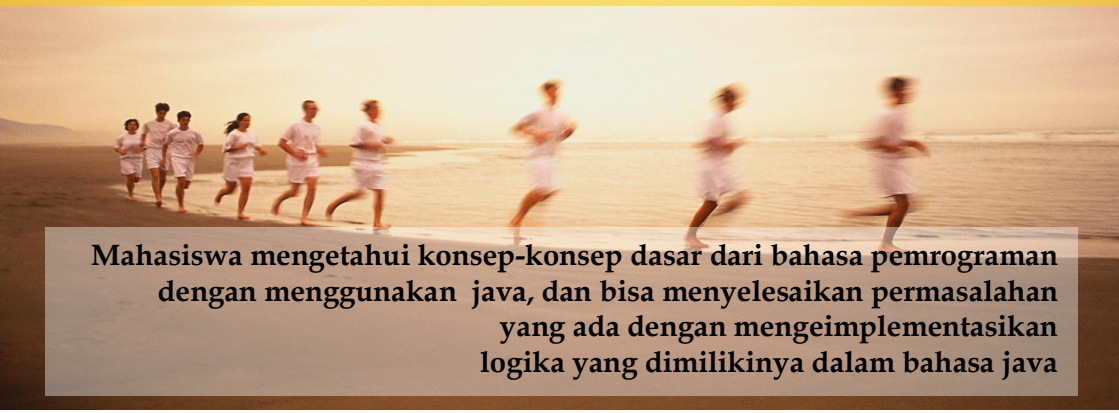
ST-RK-1.16-082-007/R-

STIKOMP SURABAYA

Modul Praktikum

BAHASA PEMROGRAMAN



Mahasiswa mengetahui konsep-konsep dasar dari bahasa pemrograman dengan menggunakan java, dan bisa menyelesaikan permasalahan yang ada dengan mengeimplementasikan logika yang dimilikinya dalam bahasa java

Laboratorium Komputer

DAFTAR ISI

PETUNJUK UMUM

Praktikum Bahasa Pemrograman merupakan mata praktikum paling dasar untuk praktikum yang berjenis pemrogaman. Sebelum anda mempelajari tentang pemrograman yang bersifat OOP(Object Oriented Programming), pemrograman visual, pemrograman WEB anda wajib untuk memahami dasar dari semua jenis pemrograman yang ada yaitu **LOGIKA**. Apapun jenis pemrogramannya dan apapun bahasa pemrogramannya hanya satu kunci untuk menyelesaikan semua permasalahan pada pemrograman yaitu **LOGIKA**. Maka dari itu pada praktikum Bahasa Pemrograman ini anda akan belajar dasar-dasar pemrograman dengan menggunakan JAVA. Disini fokus utama pembelajarannya bukan pada JAVA'nya tetapi pada penggunaan **LOGIKA** untuk menyelesaikan permasalahan pemrograman yang ada.

MODUL 1
PENGENALAN JAVA

The great aim of education
is not knowledge but action
- Herbert Spencer -

Tujuan

Praktikan mengetahui konsep dasar bahasa pemrograman dengan java

Materi

Pengenalan JAVA, Struktur Dasar Program JAVA

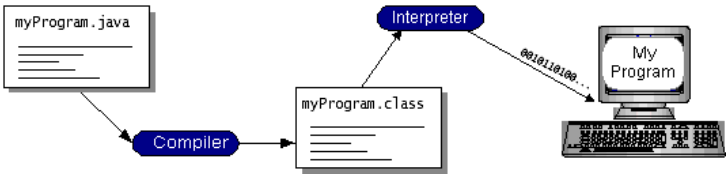
Referensi

- Abdul Kadir, Dasar Pemrograman Java 2, 2004.
- Rangsang Purnama, S.Kom, Pemrograman JAVA Jilid – 1, Juli 2002
- Java Handbook(Noughton, Patrick)
- http://java.sun.com/docs/books/tutorial/getStarted/application/index.html/Pengantar

1.1. Pengantar

Program Komputer adalah kumpulan dari instruksi –instruksi yang memandu komputer untuk menjalankan tugas tertentu, dalam kehidupan nyata dapat dicontohkan seperti sebuah resep yang akan digunakan untuk memasak masakan tertentu. Didalam resep terdapat daftar bahan yang dibutuhkan yang kita sebut variable atau data , dan juga langkah – langkah untuk membuatnya, yaitu yang memandu komputer untuk mengolah data atau variable yang telah ada. Jadi pemrograman adalah teknik untuk membuat komputer melakukan apa yang kita inginkan.

Saat ini banyak sekali bahasa pemrograman, salah satunya adalah Java. Java merupakan bahasa pemrograman tingkat tinggi yang memiliki karakteristik simple, object-oriented, distributed, interpreted, aman, dan memiliki performance yang tinggi. Bahasa pemrograman Java merupakan compiler sekaligus interpreter, dimana sebagai compiler, program yang telah dibuat akan diubah menjadi java bytecodes dan kemudian sebagai interpreter java bytecodes tersebut dijalankan pada komputer. Gambar berikut menjelaskan bagaimana java bekerja sebagai compiler dan interpreter.



Java platform memiliki dua komponen yaitu Java Virtual Machine yang berfungsi sebagai jembatan antara bytecode dengan hardware dan Java Application Programming Interface (Java API) yang merupakan komponen -komponen dan kelas java yang telah jadi dan memiliki kemampuan untuk menangani objek, string, angka, dan sebagainya.

Untuk pemrograman java dekstop sederhana, yang kita butuhkan adalah JDK (Java Development Kit) dan Editor. JDK dapat anda download di website sun microsystem.

Java Development Kit (JDK) merupakan perlengkapan yang mendasar dalam pengembangan aplikasi dengan Java. Dua program utama yang disediakan dalam JDK adalah :

javac, yaitu program untuk meng-compile kode sumber.
java, yaitu program untuk meluncurkan aplikasi.

Sedangkan untuk menulis kode program bisa menggunakan segala macam text editor yang ada termasuk notepad tetapi harus disimpan dengan ekstensi .java. Untuk praktikum Bahasa Pemrograman ini Anda akan menggunakan editor untuk bahasa Java yaitu EditPlus atau jcreator LE atau notepad++.

Pengembangan aplikasi dengan bahasa pemrograman Java pada dasarnya melalui beberapa langkah.

1. Menulis program dalam bahasa pemrograman Java, dan disimpan dalam file berekstensi .java.
2. Meng-compile program tersebut menggunakan compiler yang disediakan JDK, yaitu javac. Hasilnya adalah sebuah Java class yang disimpan sebagai file berekstensi .class.
3. Meluncurkan aplikasi dengan program java yang disediakan JDK

1.2. Membuat Program JAVA

Sekarang kita coba membuat program java untuk pertama kali. Program yang akan dibuat adalah program sederhana untuk menampilkan ”Hello World”.

Coba anda ketik kode program dibawah ini dalam text editor yang anda gunakan. Simpan kode program tersebut dengan nama HelloWorldApp.java.

Ingat : Java compiler mencocokkan nama file dengan nama class, sehingga nama file yang anda buat dengan extensi .java harus sama dengan nama class. Java bersifat Case Sensitif sehingga Hello ≠ hello.

```
/**
 * The HelloWorldApp class implements an
application that
 *   simply   prints   "Hello   World!"   to
standard output.
 */

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

```
        System.out.println("Hello
World!"); // Display the string.
    }
}
```

Code diatas, terdiri dari 3 komponen utama, yaitu : comment, class definition, dan method main.

Comment

Kode yang bercetak tebal berikut merupakan comment

```
/**
 * The HelloWorldApp class implements an
application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
// Display the string.
    }
}
```

Comment diabaikan oleh compiler, tapi sangat berguna untuk memberi catatan terhadap program yang kita buat, sehingga dapat membantu kita memahami logika yang saat ini kita buat dikemudian hari.

Berikut adalah beberapa penulisan comment di java:

`/* text */`

Compiler akan mengabaikan text diantara /* dan */

`/** documentation */`

Compiler mengabaikan text diantara /** dan */ , biasanya digunakan untuk dokumentasi.

`// text`

Compiler mengabaikan text satu baris

Class Definition

Kode yang bercetak tebal berikut adalah class definition

```
/**
 * The HelloWorldApp class implements an
application that
 * simply displays "Hello World!" to the standard
output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); //
Display the string.
    }}
```

Dari kode program diatas, kita dapat men-generalisasi-kan penulisan syntax sebuah class :

```
class name {
    . . .
}
```

Keyword class digunakan untuk mendefinisikan class dan diikuti oleh nama class. Kode program diletakkan diantara kurung kurawal.

Method Main

Kode yang bercetak tebal berikut adalah class definition

```
/**
 * The HelloWorldApp class implements an
application that
 * simply displays "Hello World!" to the standard
output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
//Display the string.
    }
}
```

Setiap aplikasi java, harus memiliki method main yang dituliskan dalam syntax:

```
public static void main(String[] args)
{ ……..
        }
```

method main menerima sebuah argument array bertipe string, biasanya programmer menggunakan nama argument args, namun anda dapat menggantinya sesuka anda.

Menampilkan Informasi Ke Layar

Pada aplikasi HelloWordApp, kita akan menampilkan sebuah string "Hello World", untuk menampilkan String ke layar, kita akan menggunakan class System dari library java, yaitu System.out.print atau System.out.println, coba anda cari perbedaan keduanya. Untuk menggabungkan String digunakan +.

Menerima Inputan Dari Keyboard

Hampir seluruh aplikasi akan membutuhkan inputan dari keyboard. Apa yang harus kita lakukan untuk menangkap inputan yang diberikan keyboard? Pada praktikum kali ini, kita akan menggunakan Class BufferedReader yang disedian oleh Java API.

Perhatikan Kode program berikut:

```
import java.io.*;
class SelamatDatang
{
 public static void main(String[] args)
 {
      try{
    String nDepan;

              BufferedReader in = new
BufferedReader(new
InputStreamReader(System.in));
              System.out.print("Masukkan Nama
Depan: ");
              nDepan = in.readLine();
              System.out.print("Masukkan Nama
Belakang: ");
              String nBelakang = in.readLine();
```

```
                System.out.println();
                System.out.println("Welcome 2
Java World " + nDepan + " " + nBelakang);
                //System.out.println("String ini
tidak akan ditampilkan);
        }
        catch(Exception e){
                //ini untuk menampilkan
errornya...
                System.out.println(e);
        }
  }
}
```

a. Untuk menggunakan Class BufferedReader kita harus mengimport class Java.io.

b. Mendefinisikan sebuah objek, pada contoh diatas kita beri nama in, anda dapat mengganti dengan nama sesuka anda.

```
BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
```

c. Membuat sebuah variable untuk menampung inputan dari keyboard

```
String nDepan;
```

Dimulai dari type data variable dan diikuti nama variable.

d. dan menangkap inputan dengan perintah readLine()

```
nDepan = in.readLine();
```

atau

```
String nBelakang = in.readLine();
```

e. Selanjutnya adalah perintah untuk menangkap kesalahan program yaitu **try..catch**. Semua baris perintah yang akan dijalankan ditempatkan didalam **try**, apabila ada kesalahan maka program akan menjalankan baris perintah yang ada di dalam **catch**. Perhatikan syntak penulisan.

Baris program di atas akan menghasilkan tampilan seperti berikut ini:

Syarat utama untuk penamaan class yaitu

a. Diawali huruf Kapital.
b. Bila lebih dari satu kata, huruf kedua diawali huruf kapital juga.
c. Tidak boleh mengandung spasi.
d. Karakter yang diperbolehkan adalah huruf dan angka.
e. Pada bahasa Java terdapat istilah kode *escape*, yaitu yang penulisannya diawali dengan simbol "\". Tabel berikut ini berisi daftar sejumlah karakter *escape*

Kode	Keterangan
\b	Backspace
\f	Form feed
\n	Baris baru (line feed)
\r	Carriage return
\t	Tabulasi
\'	Tanda kutip tunggal
\"	Tanda kutip ganda
\\	Garis miring
\ddd	Karakter oktal
\xdd	Heksadesimal (dd=0 s.d. FF atau ff)

Latihan

1 Buat program dengan tampilan berikut

```
Contoh :
Masukkan Data-data anda :
Nama depan     : andi
Nama belakang : rianto
Alamat rmh     : jl. Beringin 15
Kota           : Surabaya
Tempat Lahir   : Jakarta
Tanggal Lahir  : 04 Maret 1985

Outputnya :
============================
                Biodataku
============================
Nama Lengkap          : andi rianto
Alamat                : jl. Beringin 15 Surabaya
```

Tempat,Tanggal Lahir : Jakarta, 04 Maret 1985

2 Buat program untuk mencetak angka yang dimasukkan user, apabila yang dimasukkan bukan angka maka akan menampilkan errornya.

Contoh :
Masukkan Angka : aku
maka akan muncul pesan error :
java.lang.NumberFormatException : For Input String “aku”

MODUL 2
VARIABEL, KONSTANTA & TIPE DATA

Tujuan belajar adalah mengganti pikiran
Yang kosong dengan suatu hal baru yang
BERMAKNA

Tujuan

Praktikan memahami pengertian dari variabel,tipe data, dan operator serta dapat menggunakannya dalam program.

Materi

Variabel, Tipe Data, Operator

Referensi

- Abdul Kadir, Dasar Pemrograman Java 2, 2004.
- Rangsang Purnama, S.Kom, Pemrograman JAVA Jilid – 1, Juli 2002
- Java Handbook(Noughton, Patrick)

2.1. Pengantar

Hampir bisa dipastikan bahwa sebuah program yang kita buat selalu membutuhkan lokasi memori untuk menyimpan data yang sedang diproses. Kita tidak pernah tahu di lokasi memori mana komputer akan meletakkan data dari program kita. Kenyataan ini menimbulkan kesulitan bagi kita untuk mengambil data tersebut pada saat dibutuhkan. Maka dikembangkanlah konsep variabel. Berikut ini penjelasan selengkapnya.

2.2. Variabel

Variabel adalah suatu tempat penyimpanan yang bersifat *temporary* di memori yang digunakan dalam suatu program, karena bersifat sementara maka apabila program selesai dijalankan maka isi dari variabel akan hilang. Variabel dapat bersifat lokal, misalkan di dalam perulangan for, atau dapat juga berupa variabel instan yang dapat diakses oleh semua method dalam class. Berikut ini cara mendeklarasikan dan memberikan nilai terhadap suatu variabel:

```
tipeData variabel = nilai awal;
```

Perhatikan potongan program berikut :

```
class Hitung
{
        public static void main(String[] args)
        {
                int a = 3;
                int b;
                float c;
                b = 2;
                c = a + b;
                System.out.println(c);
        }
}
```

Pada potongan program diatas yang dimaksud dengan variabel adalah **a**, **b** dan **c** sedangkan **int** dan **float** adalah suatu tipe data.

2.3. Tipe Data

Tipe data bisa dikatakan sebagai sifat dari suatu variabel, yang hanya menyatakan model data yang diproses, bukan menyatakan

tempat untuk menyimpan data tersebut. Pada bahasa pemrograman java tipe data dikelompokkan dalam 4 kelompok yaitu :

Integer : *byte*, *short*, *int*, dan *long*
Semua tipe data Integer berupa bilangan bulat.
Pecahan : *float* dan *double*
Kedua tipe data diatas berupa bilangan pecahan
Karakter : *char*
Mewakili simbol pada himpunan karakter seperti tulisan dan angka
Boolean : *boolean*
Merupakan tipe data khusus untuk menunjukkan besaran logika (True atau False).
Untuk lebih jelas dan tepat dalam menggunakan tipe - tipe data diatas perhatikan tabel berikut ini :

Nama	Lebar (bit)	Rentang Nilai
Long	64	-9223372036854775808 s/d 9223372036854775807
Double	64	1.7E-308 s/d 1.7E+308
Int	32	-2147483648 s/d 2147483647
Float	32	3.4E-038 s/d 3.4E+038
Short	16	-32768 s/d 32767
Byte	8	-128 s/d 127

2.4. Operator

Operator adalah suatu karakter khusus yang memerintahkan *compiler* untuk melakukan suatu operasi terhadap sejumlah *operand*.
Pada contoh diatas ada satu operasi yaitu :
c = a + b;
pada contoh tersebut yang disebut sebagai operator adalah "**+**" dan operand-nya adalah **a** dan **b**.
Berikut ini adalah beberapa contoh operator pada java yang paling sering digunakan :

Operator Aritmatika

Operator	Hasil
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian

%	Modulus
++	Increment
--	Decrement
+=	Persamaan penjumlahan
-=	Persamaan pengurangan

Operator Logika

Operator	Hasil
&&	AND
||	OR
!	NOT

Operator Relasi

Operator	Hasil
==	Sama dengan
!=	Tidak sama dengan
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar dari atau sama dengan
<=	Lebih kecil dari atau sama dengan

Latihan

1 Buat program kalkulator sederhana untuk menghitung penjumlahan, pengurangan, perkalian, dan pembagian 2 bilangan

Input :
Bilangan Pertama : 5
Bilangan Kedua : 4
Output :
Hasil Penjumlahan : 9
Hasil Pengurangan : 1
Hasil Perkalian : 20
Hasil Pembagian : 1,25

2 Buat program untuk menghitung banyaknya jumlah satuan uang

Input :
Masukkan Jumlah Uang : 15250
Output :
Jumlah Pecahan Uang :
Sepuluh Ribuan = 1
Lima Ribuan = 1
Seratusan = 2
Lima Puluhan = 1

MODUL 3
PERCABANGAN

We are born to succeed, not fail
- Henry David -

Tujuan

Praktikan bisa memahami konsep percabangan dan dapat mengimplementasikannya dalam program dengan menggunakan perintah if-else dan switch

Materi

Percabangan dengan menggunakan if..else
Percabangan dengan menggunakan switch..case

Referensi

- Abdul Kadir, Dasar Pemrograman Java 2, 2004.
- Rangsang Purnama, S.Kom, Pemrograman JAVA Jilid – 1, Juli 2002
- Java Handbook(Noughton, Patrick)

3.1. Pengantar

Pada suatu program kita tidak mungkin hanya membuat pernyatan – pernyataan yang dijalankan secara urut dari baris pertama sampai terakhir secara bergantian. Program yang baik memerlukan suatu syarat khusus untuk menjalankan suatu pernyataan karena itu sekarang kita pelajari yang dinamakan **percabangan** atau ***branching*.** Secara analogi dalam kehidupan sehari – hari percabangan dapat kita lihat pada saat kita berjalan saat tiba di persimpangan kita akan memilih jalan kemana yang akan kita lewati untuk mencapai tujuan kita apakah lurus, belok kekiri, atau belok ke kanan. Apabila tujuan kita lurus maka kita tidak akan mungkin sampai ke tujuan apabila kita belok kiri atau kanan, sama dengan program untuk memperoleh hasil sesuai yang kita inginkan maka harus dilakukan suatu seleksi terhadap kondisi tertentu.

3.2. Pernyataan IF

Bentuk umum dari pernyataan if tunggal adalah:

```
if (kondisi_1 [&&/|| kondisi_2..])
{
   …instruksi jika hasil logika bernilai true…
}
```

Blok instruksi yang terletak setelah if akan dikerjakan jika hasil logika dari kondisi di belakangnya bernilai *true*. Hasil logika ini bisa dibentuk dari satu kondisi atau lebih. Sebuah instruksi if hanya bisa mengerjakan satu instruksi saja. Jika Anda menginginkan lebih banyak instruksi, Anda harus menggunakan kurung kurawal.

Bentuk lain percabangan dengan if adalah bentuk if majemuk yang merupakan susunan perintah if sedemikian rupa sehingga jika hasil logika true sudah diperoleh, maka perintah if berikutnya tidak dikerjakan. Bentuk umum dari if majemuk adalah :

```
if (kondisi_1 [&&/|| kondisi_2..])
{
   …instruksi jika hasil logika bernilai true…
}
else{
```

```
   …instruksi jika hasil logika bernilai
false…
}
```

Kata kunci else digunakan sebagai penghubung antar pernyataan if yang akan diseleksi dalam satu tingkat.

Perhatikan program berikut :

```
class Hitung
{
   public static void main(String[] args)
   {
 int angka = 4;
 if (angka%2 == 0){
    System.out.println("Bilangan Genap");
 }
 else{
    System.out.println("Bilangan Ganjil");
 }
   }
 }
```

Program diatas akan mencetak "Bilangan Genap" apabila variabel angka habis dibagi 2, sedangkan program akan mencetak "Bilangan Ganjil" apabila angka tidak habis dibagi 2 disinilah yang dimaksud dengan percabangan, program akan memilih kondisi – kondisi yang sesuai untuk menjalankan suatu pernyataan.
Bentuk lain dari pernyataan if adalah susunan if secara bertingkat. Di dalam blok instruksi if bisa terdapat blok if yang lain. Keberadaan blok if terdalam ditentukan oleh blok if di luarnya.

```
if (kondisi_1)
{
       if (kondisi_2)
             …instruksi jika hasil logika
bernilai true…
       else
             …instruksi jika hasil logika
bernilai false…
}
else
{
```

```
        if (kondisi_2)
              …instruksi jika hasil logika
bernilai true…
        else
              …instruksi jika hasil logika
bernilai false…
}
```

3.3. Pernyataan SWITCH

Perintah switch memungkinkan untuk melakukan sejumlah tindakan berbeda terhadap sejumlah kemungkinan nilai. Pada perintah switch terdapat pernyataan break, yang digunakan untuk mengendalikan eksekusi ke akhir pernyataan switch, atau dengan kata lain digunakan untuk mengakhiri eksekusi switch.

Perintah switch tidak bisa digunakan untuk ekspresi string. Bentuk umum perintah ini:

```
switch(ekspresi) {
    case nilai_1 :
      pernyataan;
      break;
    case nilai_2 :
      pernyataan;
      break;
    default :
     pernyataan;
}
```

sebagai contoh kita gunakan perintah **switch** untuk contoh program diatas :

```
class Hitung
{
   public static void main(String[] args)
   {
  int angka = 4;
  switch(angka%2){
      case 0:
              System.out.println("Bilangan
Genap");
          break;
```

```
        case 1:
        System.out.println("Bilangan Ganjil");
          break;
  }
    }
}
```

Dalam penggunaannya, pernyataan if dan pernyataan switch, masing-masing memiliki kelebihan dan kekurangan, jadi Anda harus benar – benar memperhatikan perintah mana yang cocok untuk digunakan pada program yang Anda buat.

Latihan

1 Program menentukan kelulusan dihitung dari UTS, UAS, Quiz, Tugas

Contoh :
Masukkan nama anda : Anton
Nilai UTS : 80
Nilai UAS : 70
Nilai Quiz : 75
Nilai Tugas : 60
Output :"Saudara Anton nilai akhir anda 73,anda dapat B"

(Nilai Akhir = (0.3*UTS)+ (0.4*UAS)+ (0.2*Quiz)+ (0.1*Tugas))

Nilai A → 80 – 100	Nilai C → 55 – 59
Nilai B+ → 75 – 79	Nilai D → 45 – 54
Nilai B → 65 – 74	Nilai E → 0 – 44
Nilai C+ → 60 – 64	

2 Buat program untuk mengetahui inputan user

Contoh :
Masukkan Inputan Anda : A
Hasil → Anda menginputkan huruf kapital

Masukkan Inputan Anda : b
Hasil → Anda menginputkan huruf biasa

Masukkan Inputan Anda : 12
Hasil → Anda menginputkan bilangan genap

3.Buat Program untuk mengetahui waktu dengan syarat
Pukul 00.01 – 10.00 adalah Pagi
Pukul 10.01 – 15.00 adalah Siang
Pukul 15.01 – 18.00 adalah Sore
Pukul 18.01 – 24.00 adalah Malam

Contoh : (jam saat ini pukul 08.00)
Masukkan Nama Anda : Joni
Output : "Selamat Pagi Joni"

MODUL 4
PERULANGAN

Tidak ada kata sia-sia dalam Belajar,
Berhasil atau Tidak pada akhirnya pasti ada
satu hal yang BERMAKNA

Tujuan

Praktikan bisa memahami konsep perulangan dan dapat mengimplementasikannya dalam program dengan menggunakan perintah for, while dan do..while, serta dapat menentukan perintah perulangan yang paling tepat untuk menyelesaikan suatu permasalahan dalam program

Materi

Perulangan dengan menggunakan FOR, WHILE, dan DO..WHILE

Referensi

- Abdul Kadir, Dasar Pemrograman Java 2, 2004.
- Rangsang Purnama, S.Kom, Pemrograman JAVA Jilid – 1, Juli 2002
- Java Handbook(Noughton, Patrick)

4.1. Pengantar

Satu waktu kita hendak membuat suatu program untuk mencetak angka mulai dari 0 sampai dengan 100. maka secara manual kita akan menuliskan:

```
System.out.println(0);
       :
       :
System.out.println(100);
```

Kita tidak mungkin melakukan hal itu karena tidak efisien dan membuat baris program menjadi sangat panjang. Dalam bahasa pemrograman ada yang disebut dengan ***looping*** atau **perulangan** dimana kita bisa menjalankan proses yang sama tanpa harus mengetikkan perintah tersebut secara berulang – ulang.
Bahasa pemrograman Java menyediakan 3 macam perintah untuk melakukan looping atau perulangan, yaitu :

- Perulangan dengan For
- Perulangan dengan While
- Perulangan dengan Do..While

4.2. Pernyataan FOR

Pernyataan **for** dikenal sebagai pernyataan untuk mengendalikan proses berulang dengan jumlah perulangan yang sudah ditentukan sebelumnya. Bentuk pemakaiannya :

```
for (inisialisasi;kondisi;penaikan_penurunan)
{
   Pernyataan_pernyataan;
}
```

Pada pernyataan ini :
Bagian inisialisasi digunakan untuk memberikan nilai kepada variabel yang digunakan untuk mengontrol pengulangan.

Bagian kondisi digunakan untuk mengontrol pengulangan untuk dilanjutkan atau diakhiri.

Bagian penaikan_penurunan digunakan untuk menaikkan atau menurunkan nilai variabel pengontrol perulangan.

Contoh berikut menunjukkan cara menampilkan tulisan Java lima kali dengan menggunakan for.

```
class PernyataanFor
{
   public static void main (String[] args)
   {
      for (int jumlah=1; jumlah<=5; jumlah++)
      {
           System.out.println ("Java");
      }
   }
}
```

Pada contoh di atas :
`Int jumlah=1` digunakan untuk mendeklarasikan `jumlah` dan memberikan nilai 1 ke dalam variabel tersebut.
`jumlah<7` digunakan untuk menguji apakah nilai `jumlah` kurang dari 5. Kalau ya, bagian pertanyaan akan dijalankan dan bagian `jumlah++` akan dieksekusi, kemudian pengujian dilakukan kembali. Kalau tidak maka for akan berakhir.
`jumlah++` digunakan untuk menaikkan `jumlah` sebesar 1.

4.3. Pernyataan WHILE

Pernyataan while berguna untuk melakukan proses yang berulang. Bentuk pemakaiannya:

```
 while (kondisi){
            blok_pernyataan;
 }
```

Dalam hal ini pernyataan ini akan dijalankan secara terus menerus selama kondisi bernilai **true** (benar). Contoh berikut menunjukkan cara menampilkan tulisan Java lima kali dengan menggunakan **while**.

```
public class PernyataanWhile
{
   public static void main (String[] args)
   {
     int jumlah=1;
```

```
      while (jumlah<=5)
      {
         System.out.println ("Java");
         jumlah++; //menaikkan nilai jumlah
sebesar 1
         }
    }
}
```

4.4. Pernyataan DO..WHILE

Pernyataan do..while menyerupai pernyataan while. Akan tetapi pada pernyataan do..while melakukan pengecekan terhadap suatu kondisi setelah melakukan perintah–perintah yang ada di dalamnya. Sehingga pada do..while perintah dalam blok looping pasti dijalankan satu kali. Looping akan berhenti jika kondisi bernilai *false*(salah). Bentuk pemakaiannya:

```
do{
        pernyataan_pernyataan;
  } while( kondisi);
```

Contoh berikut menunjukkan cara menampilkan tulisan Java lima kali dengan menggunakan **do..while**.

```
public class PernyataanDoWhile{
        public static void main (String[]
args){
                int jumlah = 5;
                do{
                        System.out.println("java");
                        jumlah--;
                }while(jumlah>=1);
        }
  }
```

Latihan

1 Buat program untuk menampilkan dan menghitung total bilangan fibonacci ,jumlah bilangan sesuai dengan inputan user.

contoh :
Masukkan banyak bilangan : 5
Output :
Bilangan Fibonacci : 1 1 2 3 5
Total : 12

2 Buat Program untuk menghitung nilai faktorial suatu bilangan, dan tampilkan urutan prosesnya.

contoh :
Masukkan Angka : 5
Step 1 : 5 * 4
Step 2 : 20 * 3
Step 3 : 60 * 2
Step 4 : 120 * 1
Faktorial 5! = 120

3 Buat program kalkulator.java sederhana, tetapi menggunakan menu.

contoh :
==========
Kalkulator Sederhana
==========
Menu :
Penambahan (+)
Pengurangan (-)
Perkalian (*)
Pembagian (/)
Modulus (%)
Exit
Masukkan pilihan anda : 1
Masukkan Bilangan1 : 5
Masukkan Bilangan2 : 4
Hasil Penjumlahan : 9
(Bila ditekan Enter akan kembali ke menu awal)

MODUL 5
ARRAY 1 DIMENSI

Education is the ability to listen to almost anything without losing your temper or your self-confidence
- Robert Frost -

Tujuan
Praktikan bisa memahami konsep tipe data array dan menggunakannya dalam program.

Materi
Array 1 Dimensi

Referensi

- Abdul Kadir, Dasar Pemrograman Java 2, 2004.
- Rangsang Purnama, S.Kom, Pemrograman JAVA Jilid – 1, Juli 2002
- Java Handbook(Noughton, Patrick)

5.1. Pengantar

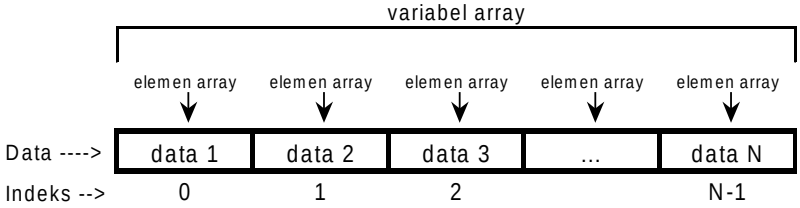
Program yang kompleks memerlukan banyak variabel sebagai inputannya. Kita bisa saja mendeklarasikan variabel–variabel tersebut satu persatu sesuai dengan jumlah yang kita butuhkan. Misalkan kita membutuhkan 5(lima) variabel bertipe *int*, kita bisa menuliskannya dengan cara :

int a, b, c, d, e;

Tetapi apabia kita memerlukan 100 variabel apakah kita harus mendeklarasikan sebanyak 100 kali juga????

Java menawarkan konsep Array untuk solusi pendeklarasian variabel dalam jumlah besar. Sebuah variabel array adalah sejumlah variabel berbeda dengan nama yang sama tetapi memiliki nomor indeks yang unik untuk membedakan setiap variabel tersebut.

Untuk lebih memahami konsep tipe data Array perhatikan ilustrasi berikut :



Indeks adalah sebuah angka yang menyatakan urutan sebuah elemen pada suatu variabel array. Karena seluruh kotak memiliki nama yang sama, maka untuk membedakannya diperlukan suatu cara yaitu dengan memberi nomor urut. Ibarat deretan rumah pada sebuah jalan, untuk membedakan antara rumah yang satu dengan rumah yang lain maka setiap rumah diberi nomor unik yang berbeda antara rumah satu dengan rumah lainnya.

Nomor indeks variabel array selalu dimulai dari 0 (nol), sehingga nomor indeks bagi elemen terakhir adalah sebesar (N-1), dimana N adalah jumlah total elemen. Kita bisa mengakses setiap elemen dalam variabel array dengan mengacu pada nomor indeksnya. Awalan nol untuk nomor indeks array sering menimbulkan kerancuan bagi kita yang terbiasa dengan awalan angka 1.

5.2. Array 1 Dimensi

Ada beberapa cara untuk mendeklarasikan variabel array, yaitu:
Mendeklarasikan variabel array tanpa menyebutkan berapa jumlah elemen yang diperlukan.

```
int [] angka;
```

Variabel angka kita deklarasikan sebagai variabel array dimana setiap elemennya akan menyimpan data bertipe **int** tetapi kita belum bisa menggunakannya sebelum kita menyebutkan berapa jumlah elemen yang diperlukan, maka untuk memesan jumlah elemen yang kita perlukan kita tuliskan :

```
angka = new int[5];
```

Berarti kita memesan 5 elemen untuk variabel angka.
Mendeklarasikan variabel array dengan menyebutkan jumlah elemen yang diperlukan.

```
int[] angka = new int[5];
```

Variabel angka kita deklarasikan sebagai variabel array dimana setiap elemennya akan menyimpan data bertipe **int**. Pada saat mendeklarasikan ini kita langsung memesan 5 elemen array yang kita perlukan. Mendeklarasikan variabel array secara otomatis :

```
int[] angka = {5, 3, 23, 99, 2};
```

Kita tidak menyebutkan berapa elemen yang kita pesan tetapi kita langsung menyebutkan isi data dari elemen array tersebut.

Contoh :

```
import java.io.*;
class BelajarArray
{
  public static void main(String[] args)
   {
        try{
```

```
            int[] angka = new int[5];
            System.out.println("Masukkan 5
Data");

 System.out.println("================");
            BufferedReader in =
            new BufferedReader(new
InputStreamReader(System.in));
            for(int i=0;i<angka.length;i++)
            {
                  System.out.print("Masukkan
Data Ke-"+(i+1)+" : ");
                  angka[i] =
Integer.parseInt(in.readLine());
            }

            System.out.println("\nData Yang
Ada Di Array : ");
            for(int i=0;i<angka.length;i++)
            {
                  System.out.println("Data
Ke-"+(i+1)+" : "+angka[i]);
            }
      }
      catch(Exception e){
            System.out.println("Error");
      }
 }
}
```

Pada baris source int[] angka = new int[5] ,kita mendeklarasikan array dengan nama angka yang mempunyai 5 elemen.
Untuk mengetahui banyaknya elemen dari suatu array kita bisa menggunakan fungsi ***length,*** pada contoh bisa dilihat di source angka.length baris 12 dan 19.

Di baris source angka[i] = Integer.parseInt(in.readLine()), kita melakukan instruksi untuk memasukkan angka yang kita masukkan ke dalam elemen array.
Baris source 18 sampai 21 digunakan untuk menampilkan isi dari array angka.

5.3. Pencarian Data

Pencarian data dalam sebuah array adalah proses membandingkan sebuah data masukan dengan sejumlah data yang tersedia berdasarkan kriteria tertentu. Salah satu contoh metode sederhana dalam pencarian data adalah sequential search. Metode ini dilakukan dengan melakukan pencarian data secara urut dari index 0 sampai akhir dari data.

```
for (int index =0; index < array.length ;
index++ )
     {
        if (dicari = array[index])
        {
        System.out.println("Data ditemukan
pada posisi :" + index);
        }
     }
```

5.4. Pengurutan Data

Salah satu manfaat penggunaan variabel array adalah kita dapat mengurutkan sejumlah data menurut kriteria tertentu. Contoh metode pengurutan data adalah selection Sort. Prinsip metode ini adalah membandingkan satu data dengan data lain dalam array yang terletak sesudahnya, jika kedua data tersebut memiliki urutan yang salah,maka kedua data tersebut akan ditukar tempat, sehingga memiliki urutan yang sesuai dengan kriteria tertentu.

```
for   (int   dataKiri=   0;   dataKiri   <
array.length-1 ; dataKiri++ )
     {
        for (int dataKanan= dataKiri + 1;
dataKanan < array.length ;dataKanan++ )
        {
        if        (array[dataKiri]        >
array[dataKanan])
        {
        int bantu = array[dataKiri];

array[dataKiri]=array[dataKanan];
              array[dataKanan] = bantu;
        }
        }
     }
```

Latihan

1 Buat program untuk mengurutkan data dari angka – angka yang diinputkan user, angka yang boleh diinputkan 0 – 9, program akan berhenti bila user menginputkan -1

contoh :
Masukkan Angka : 4
Masukkan Angka : 3
Masukkan Angka : 4
Masukkan Angka : 2
Masukkan Angka : 2
Masukkan Angka : 5
Masukkan Angka : 6
Masukkan Angka : 7
Masukkan Angka : 8
Masukkan Angka : 9
Masukkan Angka : 4

Output :

Data Setelah diurutkan	: 2 2 3 4 4 4 5 6 7 8 9
Data terkecil	: 2
Data Terbesar	: 9
Angka yang paling sering muncul	: 4

2 Buat program untuk mengurutkan data dari angka – angka yang diinputkan user, angka yang boleh diinputkan 0 – 9, program akan berhenti bila user menginputkan -1. Kemudian akan muncul inputan untuk angka yang dicari.

contoh :
Masukkan Angka : 4
Masukkan Angka : 3
Masukkan Angka : 4
Masukkan Angka : 2
Masukkan Angka : 2
Masukkan Angka : 5
Masukkan Angka : 6
Masukkan Angka : 7
Masukkan Angka : 8

Masukkan Angka : 9
Masukkan Angka : 4

Angka yang dicari : 5

Output :
Angka yang dicari 5 berada pada posisi 5

MODUL 6
ARRAY 2 DIMENSI

Experience is the name everyone gives to their mistakes
- Oscar Wilde -

Tujuan
Praktikan bisa memahami konsep tipe data array dan menggunakannya dalam program.

Materi
Array 2 Dimensi

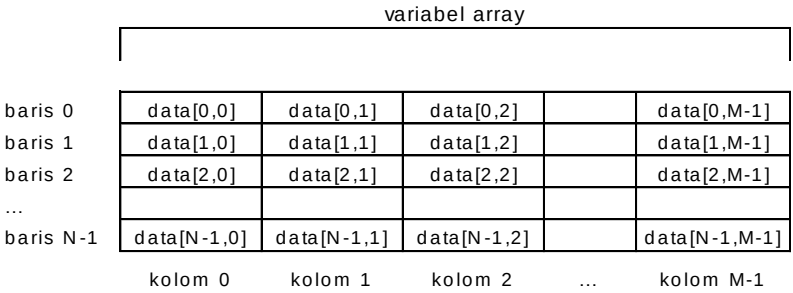
Referensi
- Abdul Kadir, Dasar Pemrograman Java 2, 2004.
- Rangsang Purnama, S.Kom, Pemrograman JAVA Jilid – 1, Juli 2002
- Java Handbook(Noughton, Patrick)

6.1. Array 2 Dimensi

Array dua dimensi merupakan bentuk yang lebih kompleks dari array yang sudah kita pelajari. Bentuk umum pendeklarasian variabel array dua dimensi di Java adalah:

```
tipeData[][] nama_variabel[=new
tipeData[jumlah_baris]
[jumlah_kolom]];
```

Pada array dua dimensi ini satu elemen array bisa memiliki beberapa isi yang bisa diilustrasikan seperti berikut :



N adalah nilai yang menyatakan jumlah baris dari array, sedangkan **M** menyatakan jumlah kolom dari array. Sama seperti array satu dimensi, penomoran indeks untuk array dua dimensi juga dimulai dari 0 untuk baris maupun kolomnya. Lebih jelasnya lihat contoh berikut ini :

```
class ArrayDuaDimensi
{
 public static void main(String[] args)
 {
       int[][] angka = new int[3][4];
       for(int i=0;i<angka.length;i++)
             for(int
j=0;j<angka[i].length;j++)

 angka[i][j]=(int)(Math.random()*10);
       System.out.println("Data Array Dua
Dimensi :");
       for(int i=0;i<angka.length;i++)
       {
             for(int
j=0;j<angka[i].length;j++)
```

```
                {

  System.out.print(angka[i][j] + " ");
                }
                System.out.println();
        }
  }
}
```

Pada baris ke-5 kita mendeklarasikan array dua dimensi dengan nama angka yang mempunyai 3 baris dan 4 kolom. Untuk mengisi array dua dimensi harus menyertakan baris dan kolom keberapa yang akan diisi, pada contoh diatas bisa dilihat pada baris 8. Pada saat mengambil isi dari array juga sama kita harus menyebutkan baris dan kolom keberapa yang akan kita ambil datanya.

Latihan

1 Buat program untuk mencetak bintang seperti contoh berikut.

Input : 3
Output :
```
***
*  *
***
```

Input : 5
Output :
```
*****
*   *
* * *
*   *
*****
```

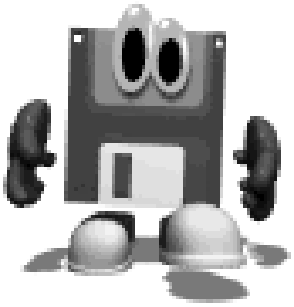
Input : 4
Output :
```
****
*  *
*  *
****
```

Input : 6
Output :
```
******
*    *
* ** *
* ** *
*    *
******
```

2. Buatlah sebuah program untuk melakukan operasi matrix, dimulai dari menginputkan dimensi matrix, mengisi matrix, penjumlahan matrix, perkalian matrix dan transpose matrix.

MODUL 7
ARRAY DINAMIS & VECTOR

Experience teaches slowly and at
the cost of mistakes.
- James A.Froude -



Tujuan

Praktikan bisa memahami konsep array dinamis dan menggunakan vector

Materi

Array Dinamis, Vector

Referensi

- Abdul Kadir, Dasar Pemrograman Java 2, 2004.
- Rangsang Purnama, S.Kom, Pemrograman JAVA Jilid – 1, Juli 2002
- Java Handbook(Noughton, Patrick)

7.1. Pengantar

Setelah anda mempelajari konsep array satu dimensi dan array dua dimensi pada modul sebelumnya sekarang anda akan mempelajari konsep array dinamis dan class Vector.

7.2. Array Dinamis

Saat kita membuat program yang menggunakan array terkadang kita mengisi jumlah elemen dari array tersebut tidak bersifat satis tapi bersifat dinamis, dalam artian jumlah elemen berdasarkan inputan user. Untuk lebih jelasnya perhatikan contoh program berikut

```
import java.io.*;
class ArrayDinamis
{
  public static void main(String[] args){

try{
 BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
 int[] angka;
 int jmlElemen = 0;
 do{
        System.out.print("\nKetik Jumlah Elemen
:");
        jmlElemen =
Integer.parseInt(in.readLine());
        if(jmlElemen > 0){
        angka = new int[jmlElemen];
        for(int i=0;i<jmlElemen;i++)
          angka[i]=(int)(Math.random()*100);

        System.out.print("Data Acak "+
jmlElemen +"elemen :");

        for(int i=0;i<jmlElemen;i++)
          System.out.print(angka[i] + "  ");
         }
        System.out.println();

         }while (jmlElemen > 0)
 }
```

```
        catch(Exception e){
              System.out.println("Error");
        }
  }
}
```

Pada baris 8 dan 9 kita mendeklarasikan variable array tanpa menginisialisasinya. Pada baris14 ita baru menginisialisasi array yang ada dengan jumlah elemen sesuai dengan yang diinputkan user.

7.3. Class Vector

Class Vector merupakan model array-of-object yang bersifat *growable,* dalam artian ukuran atau jumlah elemen yang disimpan bias bertambah atau berkurang secara dinamis. Pada modul ini kita bahas sedikit saja mengenai Class Vector.

```
import java.io.*;
import java.util.Vector;
import java.util.Enumeration;
class MyVector
{
  public static void main(String[] args)
   {
     try{
       System.out.println("================");
 BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
 Vector vectorku = new Vector();
 Enumeration enum;
 int angka = 0;
 System.out.println("Ketik sembarang angka
untuk disimpan di Vector");
 System.out.println(" dan ketik 0 bila
selesai");
       do{
       System.out.print("Masukkan Angka :");
       angka =
Integer.parseInt(in.readLine());

       if(angka != 0)
         vectorku.add(new Integer(angka));
       }while (angka != 0);
```

```
        System.out.println("Data yang ada di
Vector :");
        while(enum.hasMoreElements())
          System.out.print(enum.nextElement() +
" ");


  }
        catch(Exception e){
              System.out.println("Error");
        }
  }
}
```

Sebuah variable enum digunakan untuk menyimpan data yang tersimpan di vector vectorku. Variabel enum merupakan *instance* dari class Enumeration, yaitu class yang mengimplementasikan deretan data dari berbagai tipe. Class ini ada di package java.util sehingga kita harus mengimpornya di awal program.

Latihan

1 Buatlah sebuah program kasir sederhana untuk toko alat tulis seperti dibawah ini :

```
++++++++++++++++++++++++++++++
Menu Kasir
++++++++++++++++++++++++++++++
Input Pembelian
Cetak Nota
Keluar

Masukkan pilihan Anda : 1
Silahkan masukkan item pembelian(ketik "/" bila selesai)
Nama Item      : Pensil 2B
Harga Satuan   : 1200
Banyaknya      : 5

Nama Item      : Buku Tulis SiDu
Harga Satuan   : 2000
Banyaknya      : 4
```

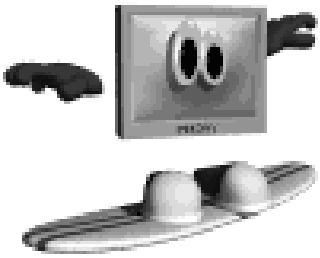
```
        Nama Item       : Bolpen Faster
        Harga Satuan    : 1900
        Banyaknya       : 3

Nama Item       : / (Kembali ke Menu Kasir)

Masukkan pilihan Anda : 2
++++++++++++++++++++++++++++++++++++++++++++++++
                Nota Pembelian Toko Alat Tulis Barokah
++++++++++++++++++++++++++++++++++++++++++++++++
Nama Item               Harga           Banyaknya       Total
------------------------------------------------------------------------------
Pensil 2B               1200                    5       6000
Buku Tulis SiDu         2000                    4       8000
Bolpen Faster           1900                    3       5700
------------------------------------------------------------------------------
                                        Total Pembelian19700
Kembali ke Menu Kasir (Y/N): Y
```

MODUL 8
PROSEDUR & FUNGSI

Janganlah jadi pintar kalau tujuanmu untuk membodohi orang lain, jadilah pintar agar orang tahu banyak hal baru darimu



Tujuan

Praktikan dapat menggunakan prosedur dan fungsi dalam suatu program

Materi

Fungsi
Fungsi rekursif
Fungsi overloading

Referensi

- Abdul Kadir, Dasar Pemrograman Java 2, 2004.
- Rangsang Purnama, S.Kom, Pemrograman JAVA Jilid – 1, Juli 2002
- Java Handbook(Noughton, Patrick)

8.1. Pengantar

Pada saat kita membuat program kadangkala kita butuh melakukan suatu hal yang sama secara berulang – ulang, misal kita memerlukan perhitungan yang serupa pada beberapa bagian program lalu kita ingin mengganti rumus dari perhitungan tersebut. Apakah kita harus mengubahnya satu persatu? Tentu saja tidak. Oleh karena itu Java menyediakan suatu fasilitas untuk mengatasi masalah tersebut, kita bisa menuliskan baris perintah program yang akan dilakukan berulang – ulang didalam suatu **sub program**. Pada modul ini akan dibahas tentang sub program dengan topik sebagai berikut:

- Sub program berjenis prosedur
- Sub program berjenis fungsi
- Sub program dengan parameter berupa variabel biasa
- Sub program dengan parameter berupa variabel array
- Sub program yang ditulis ulang: function overloading
- Sub program yang dikerjakan berulang-ulang: recursive function

8.2. Sub program berjenis prosedur

Sebenarnya Java tidak memiliki sub program yang disebut prosedur. Seluruh sub program di Java masuk ke dalam kategori fungsi. Hanya kata kunci **void** yang menyebabkan suatu sub program disebut sebagai prosedur. Prosedur adalah suatu sub program yang bertugas untuk mengerjakan suatu proses tertentu tanpa mengembalikan hasil proses tersebut. Untuk lebih jelas perhatikan contoh berikut :

```
class CetakGaris
{
 private static void garis()
 {

 System.out.println("==========================
====");
 }
 public static void main(String[] args)
 {
```

```
        garis();
        System.out.println("Data Mahasiswa");
        garis();
        System.out.println("Nim :
03.41010.0000");
        System.out.println("Nama      :
Programmer Java");
        System.out.println("Jurusan   : S1
Sistem Informasi");
        garis();
  }
}
```

Pada baris 3 kita mendeklarasikan suatu sub program bertipe void, didalamnya kita tuliskan perintah–perintah yang akan dikerjakan pada saat prosedur dipanggil. Untuk memanggil prosedur kita cukup menuliskan nama prosedur yang akan dipanggil, pada baris 9 merupakan cara untuk memanggil prosedur **`garis()`**.

8.3. Sub program berjenis fungsi

Fungsi merupakan jenis sub program yang mengembalikan suatu nilai. Seperti prosedur, tipe data pada fungsi dapat bertipe int, double, String dsb. Pada bagian akhir sebuah fungsi terdapat pernyataan **return** yang menyatakan nilai yang dikembalikan oleh fungsi.

```
import java.io.*;
class  FungsiHitung
{
 private static int a, b, c;
 private static int tambah()
 {
        return(a + b);
 }
 private static int kurang()
 {
        return(a - b);
 }
 public static void main(String[] args)
 {
        try{
        BufferedReader in =
```

```
      new BufferedReader( new
InputStreamReader( System.in ) );
        System.out.print("Masukkan Bilangan 1
:");
        a = Integer.parseInt(in.readLine());
        System.out.print("Masukkan Bilangan 2
:");
        b = Integer.parseInt(in.readLine());
        c = tambah();
        System.out.println("Hasil Penjumlahan =
" + c);
        System.out.println("Hasil Pengurangan =
" + kurang());
        }
        catch(Exception e){}
  }
}
```

Pada contoh di atas kita bisa melihat perbedaan antara prosedur dengan fungsi, pada baris 7 dan 11 merupakan perintah untuk mengembalikan nilai. Baris 22 merupakan baris perintah untuk memanggil fungsi dan menampung nilai kembalinya pada suatu variabel penampung, pada contoh diatas adalah variabel c. Nilai balik dari suatu fungsi juga dapat langsung digunakan, seperti dicontohkan pada baris 24. untuk selanjutnya kita menggunakan istilah fungsi untuk semua jenis sub program.

8.4. Sub program dengan parameter berupa variabel biasa

Pada contoh program Hitung di atas, fungsi untuk menambah dan mengurangi 2 bilangan menggunakan variabel yang bersifat tetap yaitu a dan b, lalu bagaimana jika kita akan menggunakan kedua fungsi tersebut dari beberapa bagian program yang berbeda? Dalam menyelesaikan permasalahan ini, perlu diterapkan konsep parameter di dalam fungsi. Parameter adalah data yang dikirim ke dalam suatu fungsi untuk diproses. Yang dimaksud dengan parameter berupa variabel biasa adalah parameter fungsi bertipe skalar, yaitu int, double, boolean, char dan sebagainya.

```
import java.io.*;
class  FungsiHitung
{
```

```
 private static int tambah(int bil1, int bil2)
 {
       return(bil1 + bil2);
 }
 public static void main(String[] args)
 {
       int a, b, c;
       try{
       BufferedReader in =
         new BufferedReader( new
InputStreamReader( System.in ) );
       System.out.print("Masukkan Bilangan 1
:");
       a = Integer.parseInt(in.readLine());
       System.out.print("Masukkan Bilangan 2
:");
       b = Integer.parseInt(in.readLine());
       c = tambah(a,b);
       System.out.println("Hasil Penjumlahan =
" + c);
       }
       catch(Exception e){}
 }
}
```

Pada pendeklarasian fungsi tambah kita menambahkan parameter yaitu bil1 dan bil2, untuk memanggilnya harus disertakan juga data yang akan diproses sebagai isi dari parameter tersebut.

8.5. Sub program dengan parameter berupa variabel array

Selain bisa mengirim parameter bertipe variabel biasa, pada suatu fungsi kita juga bisa mengirim parameter yang berupa data array. Untuk lebih jelasnya perhatikan contoh program berikut :

```
class ParamArray
{
 private static double rataRataArray (int[]
data)
 {
       double jumlah = 0;
       for (int i=0; i<data.length; i++)
             jumlah += data[i];
       return (jumlah/data.length);
```

```
  }
  private static void cetakArray (int[] data)
  {
        for (int i=0; i<data.length; i++)
              System.out.println ("Data ke-" +
(i+1) + " : " + data[i]);
  }
  public static void main (String[] args)
  {
        int[] angka = {7,3,5,8};
        double rataRata = rataRataArray
(angka);
        cetakArray(angka);
        System.out.println ("\nRata-rata
seluruh data : " + rataRata);
  }
}
```

Pada contoh program diatas bisa kita lihat, penulisan dan pemanggilan fungsi sama dengan fungsi berparameter biasa hanya saja tipe datanya berupa tipe data Array.

8.6. Overloading Function

Fungsi overloading adalah suatu fungsi yang bisa dideklarasikan lebih dari satu kali. Bingung???sebenarnya fungsi yang dideklarasikan lebih dari sekali tidak sama persis, fungsi – fungsi tersebut memiliki nama yang sama tetapi parameter atau tipe datanya harus berbeda satu sama lain. Untuk lebih jelasnya perhatikan contoh berikut :

```
class FungsiOverloading
{
  private static int tambah (int bil1, int
bil2)
  {
        return (bil1+bil2);
  }
  private static double tambah(double
bil1,double bil2)
  {
        return(bil1+bil2);
  }
```

```
 public static void main(String[] args)
 {
       int a = 4;
       int b = 5;
       int c;
       double d = 5.3;
       double e = 4.5;
       double f;
       c=tambah(a,b);
       f=tambah(d,e);
       System.out.println("Hasil1 = "+c);
       System.out.println("Hasil2 = "+f);
 }
}
```

Pada contoh diatas kita lihat fungsi dengan nama tambah dideklarasikan dua kali tetapi tipe data dari kedua fungsi itu berbeda yaitu **`int`** dan **`double`**. Tetapi saat memanggil fungsi tersebut program secara otomatis akan memanggil fungsi dengan parameter yang sesuai tipe datanya.

8.7. Recursive Function

Fungsi yang dikatakan bersifat rekursif adalah suatu fungsi dimana salah satu baris perintah pada suatu program memanggil fungsi yang sama dengan dirinya atau dengan kata lain fungsi rekursif adalah fungsi yang memanggil dirinya sendiri.

```
class Rekursif
{
 private static void tampilkanKata(String
kata, int banyak)
 {
       if(banyak>0){
             System.out.println(kata);
             tampilkanKata(kata,banyak-1);
       }
 }
 public static void main(String[] args)
 {
       tampilkanKata("JAVA",5);
 }
}
```

Ada dua syarat yang harus dimiliki suatu fungsi agar bisa dikatakan sebagai fungsi rekursif, yaitu :

- Ada terminating point
  contoh : **if** (banyak > 0)
- Ada bagian yang berulang – ulang
  contoh : tampilkanKata(kata,banyak-1);

Latihan

1 Buat program untuk menghitung luas dan keliling bidang datar. Perhatikan contoh berikut.

```
==========================
   Luas & Keliling Bidang Datar
==========================
Menu :
1.Bujur Sangkar
2.Persegi Panjang
3.Segitiga
4.Lingkaran
5.Exit

Masukkan pilihan anda          : 1
Masukkan panjang sisi  : 5
Luas bujur sangkar     : 25
Keliling bujur sangkar  : 20

Press Enter to continue……
(Bila ditekan Enter akan kembali ke menu awal )
```

2 Buka kembali kalkulator.java yang telah anda buat pada modul4, modifikasi program tersebut dimana masing – masing perhitungan dijadikan fungsi sendiri dan untuk mencetak hasilnya memanggil sebuah fungsi cetak.

3. Buat program untuk melakukan decrement dengan menggunakan fungsi rekursif.

```
Contoh :
Masukkan angka          : 5
Output          : 5 4 3 2 1 0
```

4. Buat program mencari faktorial dengan fungsi rekursif.

```
Contoh :
Masukkan Bilangan      : 5
Hasil Faktorial  : 120
```

5. Buat program untuk menggabungkan kata dimana ada fungsi untuk menggabungkan kata dengan kata dan kata dengan bilangan.