

MODUL 1

ALGORITMA PEMROGRAMAN

Siswa mampu memahami tentang dasar-dasar algoritma pemrograman, cara penyajian dan pengimplementasiannya dalam menyelesaikan masalah.

1.1. Pengertian Algoritma

Algoritma adalah langkah-langkah logis penyelesaian masalah, artinya langkah-langkah dalam suatu algoritma harus logis, sesuai dengan tujuan yang akan dicapai dan dapat ditentukan nilai kebenarannya. Langkah yang logis juga dapat diartikan pula tidak ambigu. Selain itu penyusunan langkah-langkah harus sistematis, atau terstruktur menurut aturan/sistem tertentu sesuai dengan tujuan yang dicapai. Artinya langkah-langkah yang ada pada algoritma harus dapat diikuti atau ditelusuri dengan baik sehingga mencapai hasil akhir yang diinginkan. Selain itu langkah-langkah dalam algoritma juga harus terbatas, artinya ia akan berhenti setelah melakukan sejumlah langkah. Jika sebuah algoritma mempunyai langkah yang tidak terbatas, ini berarti usaha mencari penyelesaian yang tidak kunjung berhasil. Maka dapat dikatakan algoritma tersebut sia-sia atau tidak berguna.

1.2. Konsep dan Struktur Algoritma

Inti dari algoritma adalah menemukan solusi dari suatu permasalahan. Untuk menyelesaikan masalah, algoritma membutuhkan spesifikasi input (masukan) sesuai yang diperlukan, memprosesnya melalui serangkaian langkah-langkah dan menghasilkan output sebagai solusi dari permasalahan.

Secara umum struktur Algoritma terdiri dari 3 bagian, yaitu :

1. Nama/judul Algoritma

Nama Algoritma memberikan gambaran secara singkat apa tujuan dari Algoritma. Pemberian nama Algoritma disarankan singkat dan jelas, namun sudah mewakili maksud dari algoritma. Dalam Algoritma komputer biasanya nama algoritma dituliskan tanpa menggunakan spasi, misalkan Algoritma VolumeBalok atau Algoritma Volume_Balok

2. Bagian Deklarasi

Bagian deklarasi merupakan tahap persiapan dari algoritma. Pada bagian ini dijelaskan kebutuhan agar algoritma dapat berjalan. Dalam algoritma pemrograman, bagian deklarasi menjelaskan input (masukan), termasuk jenis data input (tipe data), juga output apa yang akan dihasilkan serta semua hal yang akan dipakai dalam algoritma. Yang didefinisikan dalam algoritma ini : variabel, tipe data, konstanta, nama prosedur, tipe, dan fungsi.

3. Bagian Deskripsi

Pada bagian ini dijelaskan serangkaian langkah-langkah (instruksi) atau pernyataan (statement) untuk memproses alat dan bahan atau inputan untuk menghasilkan output

sesuai yang diharapkan. Langkah-langkah dalam algoritma dituliskan dari atas ke bawah. Urutan penulisan menentukan urutan perintah.

1.3. Penyajian Algoritma yang Baik









Menurut Donald Ervin Knuth, yang dikenal dengan Bapak "Analisis Algoritma", algoritma yang baik dan benar harus memiliki kriteria-kriteria berikut ini:

- 1) Input, Algoritma memiliki nol input atau lebih dari pengguna. Setiap algoritma pasti memiliki input.
- 2) Output, Algoritma minimal harus memiliki 1 output. Tujuan dari algoritma adalah memberikan penyelesaian dari suatu permasalahan dengan langkah-langkah tertentu. Penyelesaian itulah output dari algoritma yang dimaksud.
- 3) Finite, Algoritma yang baik haruslah mempunyai langkah-langkah terbatas, yang berakhir pada suatu titik di mana algoritma itu akan berhenti dan menghasilkan suatu output.
- 4) Definite, Makna dari langkah logis pada definisi algoritma terdahulu tercermin dari langkah-langkah yang pasti, tidak ambigu atau bermakna ganda. Suatu program harus mempunyai arah dan tujuan yang jelas, kapan mulai dan kapan berakhir.
- 5) Efisien, Algoritma disebut efisien jika untuk mendapatkan suatu solusi tidak memerlukan memori yang banyak, proses yang berbelit-belit dan tidak perlu. Jika algoritma terlalu banyak melakukan hal-hal yang tidak perlu akan menyebabkan waktu eksekusi menjadi lebih lama.

1.4. Penyajian Algoritma

Bahasa natural	Bahasa Pseudocode	Flow Chart
<ul style="list-style-type: none"> Menggunakan kalimat deskriptif, bahasa mudah dipahami. Cocok untuk algoritma yang singkat, sulit untuk algoritma yang besar, sulit dikonversi ke bahasa pemrograman. 	<ul style="list-style-type: none"> Penyajian algoritma dalam bentuk bahasa yang "mirip" dengan bahasa pemrograman. Algoritma menjadi bahasa yang universal, khususnya bagi pengembang program. Tidak terikat dengan aturan penulisan kode, lebih bebas, yang penting mudah dipahami oleh orang yang akan menjalankan algoritma tersebut. 	<ul style="list-style-type: none"> Bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antar proses (instruksi) dalam suatu program. Memberikan gambaran visual data diproses, memperjelas pengimplementasian algoritma pada program komputer

Simbol – simbol dalam flowchart diantaranya :

Simbol	Nama	Fungsi
	Terminator	Permulaan atau akhir progam/algoritma
	Garis Alir (Flow Line)	Arah aliran algoritma/program
	Preparation	Proses inisialisasi atau pemberian harga awal, misalkan mendefinisikan konstanta di awal
	Input/Output data	Proses input atau output data, informasi
	Proses	Proses perhitungan atau pengolahan data
	Decision (Percabangan)	Kondisi bersyarat, yang memungkinkan algoritma menjalankan perintah untuk kondisi yang berbeda-beda
	ON Page Connector	Penghubung bagian flow chart yang berada pada satu halaman, digunakan jika algoritma cukup besar sehingga harus di pecah penyajiannya
	OFF Page Connector	Penghubung bagian flow chart yang berada pada satu halaman, digunakan jika algoritma cukup besar sehingga harus di pecah penyajiannya

Berikut ini adalah beberapa contoh penyajian algoritma :

a) Bahasa Natural

Algoritma Jumlah Kuadrat

Input: dua buah bilangan bulat

Output: jumlah kuadrat dua buah bilangan bulat

- 1) Masukkan sebuah bilangan
- 2) Masukkan bilangan berikutnya
- 3) Jumlahkan kuadrat bilangan pertama dengan kuadrat bilangan kedua
- 4) Tuliskan hasil penjumlahan pada langkah sbelumnya
- 5) Selesai

b) Bahasa Pseudocode (Kode Bayangan)

Algoritma Jumlah Kuadrat

{Algoritma ini menghitung jumlah kuadrat dari dua buah bilangan, a dan b}

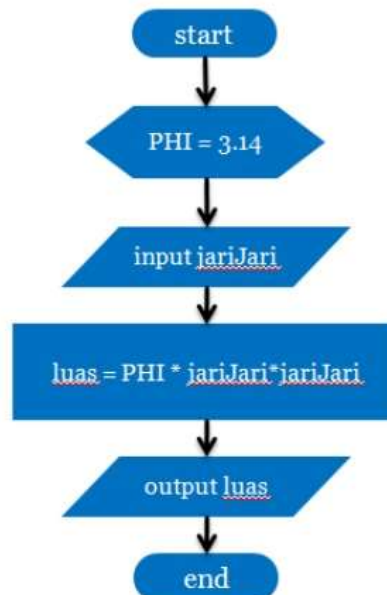
VARIABEL A,B,C:Bilangan Real

- 1) READ A
- 2) READ B
- 3) $C = A^2 + B^2$
- 4) WRITE C
- 5) END

c) Flowchart (Diagram Alir)

Kasus 1 :

Menghitung luas lingkaran



1.5. Tugas

1. Buatlah algoritma inventarisasi barang , dimana program dapat melakukan *input*, *read*, *search*, *update*, dan *delete data*. Sajikan kedalam bentuk bahasa natural, pseudocode dan *flowchart*.

MODUL 2

INSTALASI PERANGKAT LUNAK BAHASA PEMROGRAMAN

Siswa mampu melakukan instalasi perangkat lunak bahasa pemrograman sesuai dengan langkah-langkah atau prosedur yang telah ditetapkan dan menguji hasil instalasi program yang telah dilakukan.

2.1 IDE (Integrated Development Enviroment)

Tools pengembang yang berfungsi menterjemahkan bahasa kedalam bahasa yang dimengerti komputer. Lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun GUI, suatu text atau kode editor, suatu compiler atau interpreter dan suatu debugger. Pada praktik kali ini menggunakan IDE NetBeans.

Sebuah IDE biasanya terdiri dari empat komponen :

- a) Intepreter / Compiler, penerjemah bahasa pemrograman.
- b) Debugger, menguji kode program yang telah dibuat.
- c) Linker, menghubungkan file-file objek, library dll.
- d) Editor, lembar kerja penulisan kode program.

2.2 IDE NetBeans

NetBeans adalah *Integrated Development Environment* (IDE) berbasiskan Java dari Sun Microsystems yang berjalan di atas Swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat berjalan di berbagai macam platforms seperti Windows, Linux, Mac OS X and Solaris. Suatu IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun Graphical User Interface (GUI), suatu text atau kode editor, suatu compiler atau interpreter dan suatu debugger. Netbeans merupakan software development yang Open Source, dengan kata lain software ini di bawah pengembangan bersama dan bebas biaya.

2.3 Tugas

1. Lakukan instalasi IDE NetBeans, pastikan dapat dibuka dan berjalan sesuai dengan mestinya. Coba buat program sederhana menampilkan kata "Hello World".

MODUL PRAKTIKUM 1

PEMROGRAMAN DASAR DENGAN NETBEANS

Siswa mampu memahami tentang dasar-dasar pemrograman dan mengimplementasikan program sederhana dengan Netbeans.

1.1 Pengertian Java

Java merupakan bahasa pemrograman berorientasi objek dan bebas platform, dikembangkan oleh SUN Micro System dengan sejumlah keunggulan yang memungkinkan java dijadikan sebagai bahasa pengembangan enterprise. Teknologi java yang dibahas pada modul praktikum ini adalah J2SE (Standard Edition), yaitu pemrograman java pada PC. J2SE merupakan teknologi yang dirancang untuk berjalan di atas PC dan workstation yang dapat berjalan di platform system operasi Linux, Windows, dll.

1.2 NetBeans

NetBeans adalah *Integrated Development Environment* (IDE) berbasiskan Java dari Sun Microsystems yang berjalan di atas Swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat berjalan di berbagai macam platforms seperti Windows, Linux, Mac OS X and Solaris. Suatu IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun Graphical User Interface (GUI), suatu text atau kode editor, suatu compiler atau interpreter dan suatu debugger. Netbeans merupakan software development yang Open Source, dengan kata lain software ini di bawah pengembangan bersama dan bebas biaya.

Beberapa fungsi yang terdapat dalam pemrograman java, yaitu :

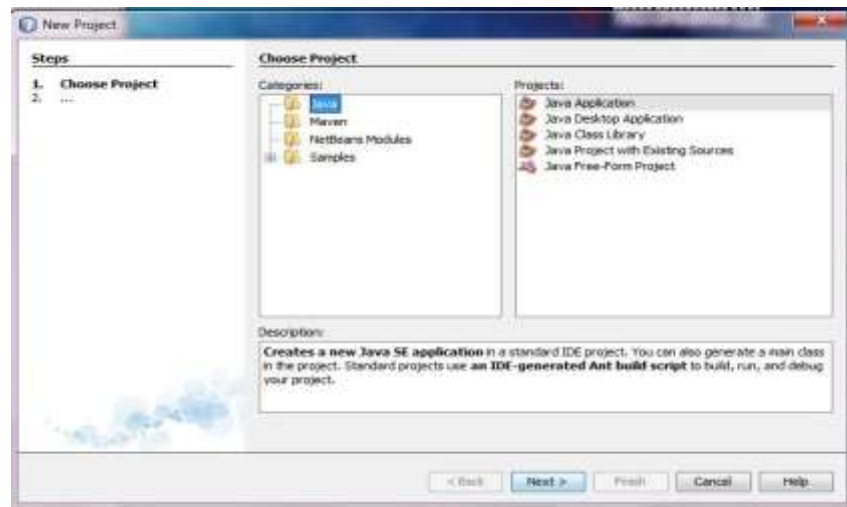
- ❖ Public : mengidikasikan anggota kelas dapat diakses dari manapun di dalam program.
- ❖ Void : menjelaskan kepada compiler bahwa fungsi tidak mengembalikan nilai apapun manakala melakukan eksekusi.
- ❖ Static : mengijinkan main untuk di panggil tanpa harus menciptakan dan kejadian (instance) kelas itu.
- ❖ Main : fungsi yang melaksanakan tugas tertentu dan hal ini merupakan titik awal dari semua aplikasi java.

1.3 Pelaksanaan Praktikum

Langkah – langkah praktikum

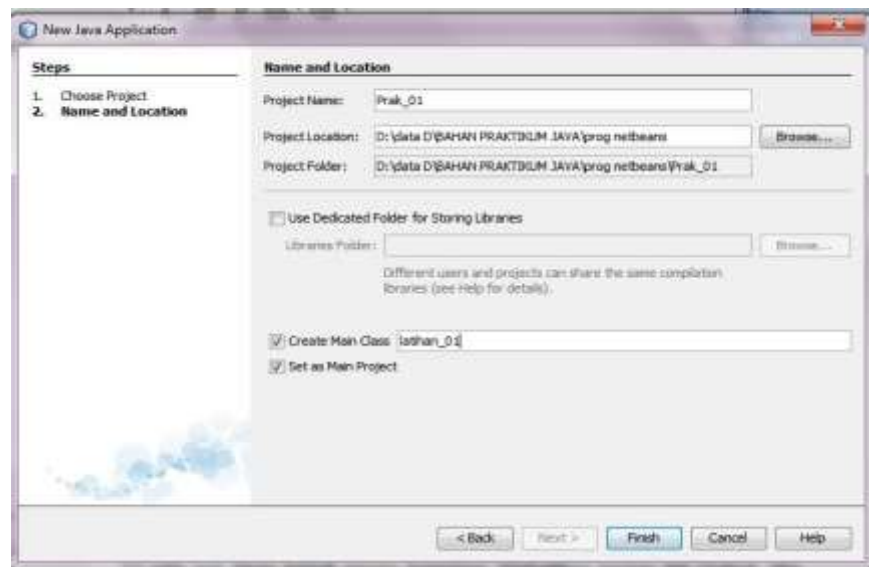
1. Pilih Start → Program → NetBeands

2. Muncul tampilan awal. Kemudian pilih menu bar File → New Project muncul layar dibawah ini :



Gambar 1.1 Tampilan jendela New Project

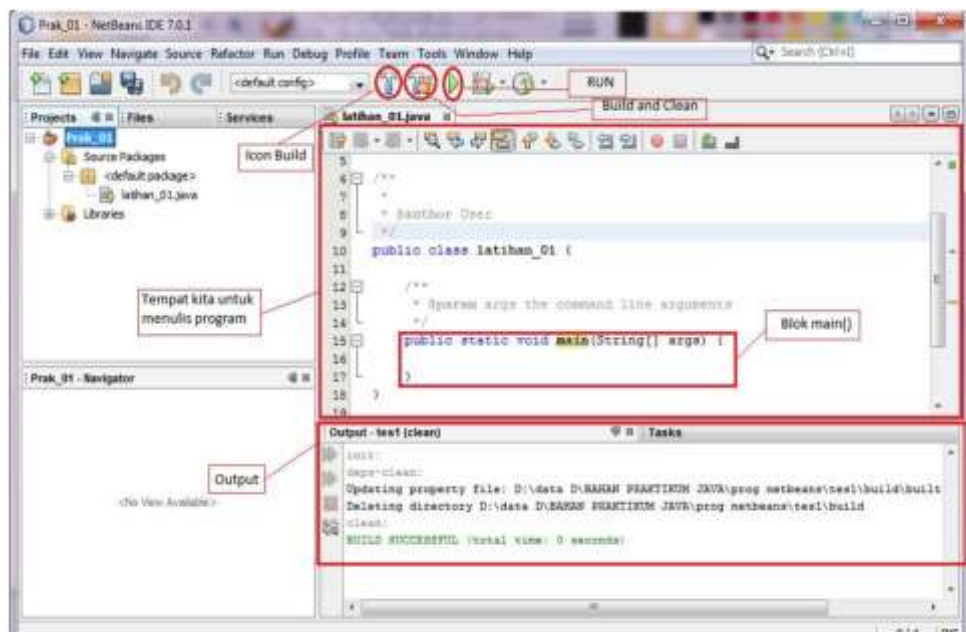
3. Pilih **Categories: Java** dan **Projects: Java Application** → Next Akan muncul layar sebagai berikut :



Gambar 1.2 Tampilan jendela New Java Application

4. Tulis nama project pada sisian Project Name
5. Pilih lokasi penyimpanan pada isian Project Location. Bisa menuliskan lokasinya atau menggunakan tombol Browser.
6. Centang Set as Main Project, ini digunakan untuk menjadi project utama.
7. Centang Create Main Class, ini digunakan jika langsung membuat kelas main dan bisa mengganti nama kelasnya.

8. Kemudian klik finish.

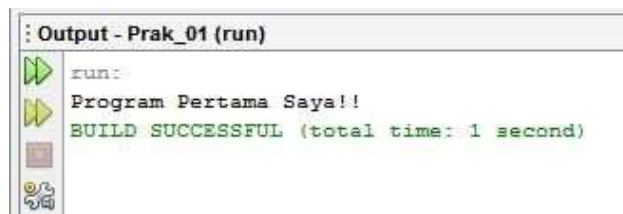


Gambar 1.3 Tampilan Project Prak_01

9. Kemudian ketikkan program di dalam blok **main()**

System.out.println("Program Pertama Saya !!");

10. Pilih run Akan terjadi proses kompilasi. Perhatikan bagian tab Output. Jika berhasil, tidak ada kesalahan akan muncul tampilan sebagai berikut :



Gambar 1.4 Tampilan Output Class latihan_01

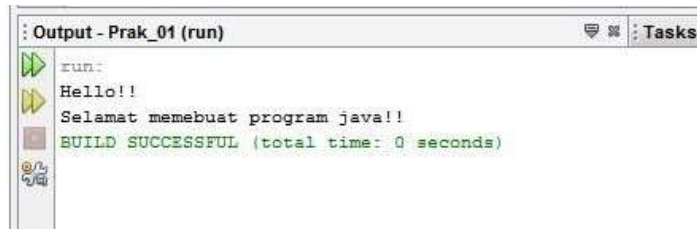
11. Untuk menambah kelas baru. Klik kanan ;project **Prak_01** → New → java class.

12. Kemudian tuliskan nama kelasnya. Dalam contoh ini diberi nama **latihan_02** → Finish

13. Ketikkan program di bawah ini pada blok **main()**

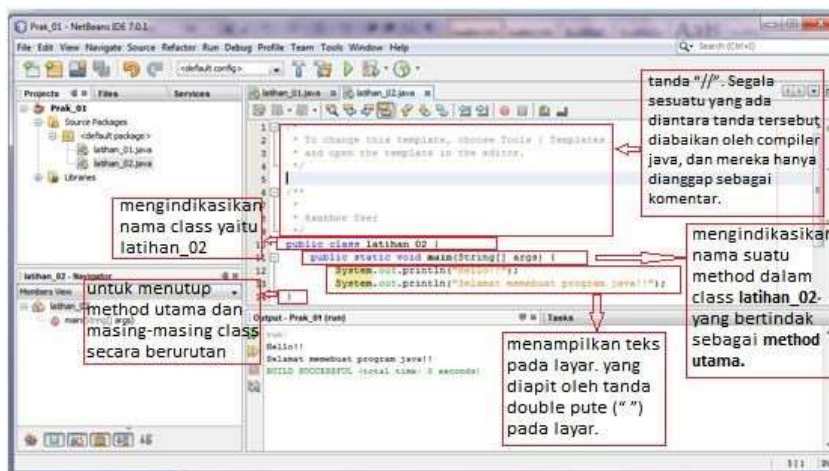
```
public class latihan_02 {  
    public static void main(String[] args) {  
        System.out.println("Hello!!");  
        System.out.println("Selamat membuat program java!!");  
    }  
}
```


14. kemudian jalankan dengan cara klik kanan kelas **latihan_02** → Run. Tampilan program setelah di run main project



Gambar 1.5 Tampilan Output Class latihan_02

Berikut gambar penjelasan dari listing program :



1.4 Tugas

1. Buatlah project baru dengan nama Praktikum1 dan kelas baru dengan nama Prog_01. Ketikkan program berikut :

BIODATA PRIBADI

=====

Nama : Cristiano Ronaldo
NIM : 155777000
Jurusan : Sistem Informasi
Mata Kuliah : Pemrograman Java

2. Lengkapi program diatas dengan menambah jenis kelamin, alamat.

MODUL PRAKTIKUM 2

PROGRAM DENGAN TIPE DATA DAN VARIABEL

Modul praktikum 2 membahas tentang statement output dengan print dan println dan mengenal penggunaan variable.

2.1 Tipe Data

Bahasa Pemrograman Java adalah bahasa pemrograman yang selalu menggunakan tipe data untuk setiap variabelnya. Itu berarti bahwa semua variabel harus dideklarasikan terlebih dahulu sebelum mereka digunakan. Tipe data dibedakan menjadi dua kategori :

1. Tipe data dasar (*Primitive*)

Tabel 2.1 Tipe data dasar pada bahasa pemrograman Java

Nama tipe data	Memori	Keyword
Integer	4 byte	Int
Short integer	2 byte	Short
Long integer	8 byte	Long
Byte-length integer	1 byte	Byte
Single-precision floating-point	4 byte	Float
Double-precision floating-point	8 byte	Double
Single character	1 byte	Char
Boolean value (true or false)	True or false	Boolean

2. Tipe data bentukan (*Reference*)

Tabel 2.2 Tipe data bentuk (reference) pada Java

Nama tipe data	Uraian
Array	Suatu koleksi dari beberapa atribut yang memiliki tipe data yang sama. Sebagai contoh : Nama para siswa, Suhu udara dalam 1 minggu .
Class	Suatu koleksi metode dan variable. Sebagai contoh :

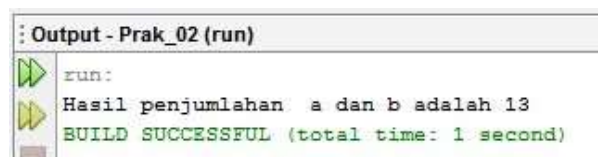
	<ul style="list-style-type: none"> ➤ Kelas siswa, berisi tentang rincian yang lengkap menyangkut siswa, dan metode yang beroperasi pada rincian itu. ➤ Kelas pelanggan, berisi tentang rincian yang lengkap menyangkut pelanggan, dan metode yang beroperasi pada rincian itu, seperti pencetakan faktur.
Interface	Suatu kelas abstrak yang diciptakan untuk menerapkan berbagai warisan di program java.

Latihan_01

```
public class tipe_data {
    public static void main(String args[]){
        float a = 3.14f;
        int b = (int) a + 10;
        System.out.println("Hasil penjumlahan a dan b adalah "+b);
    }
}
```

ini menggunakan konsep tipe data. Nilai float di dalam **a** dikonversikan untuk suatu bilangan bulat : 3.14 dan nilai **a** menambahkan dengan **10**, hingga menghasilkan nilai **13**, yang disimpan di dalam variable **b**.

Hasil Output :



Gambar 2.1 Tampilan Output Class tipe_data

2.2 Variabel

Variabel digunakan dalam program untuk menyimpan suatu nilai, dan nilai yang ada padanya dapat diubah selama eksekusi berlangsung. Di dalam java, untuk menyimpan suatu nilai, seperti numeric, konstanta atau string harus melalui nama variabel yang harus di deklarasikan. Deklarasikan variabel merupakan suatu bentuk pemberitahuan kepada compiler tentang nama variabel.

Tipe_data nama_variabel [=nilai];

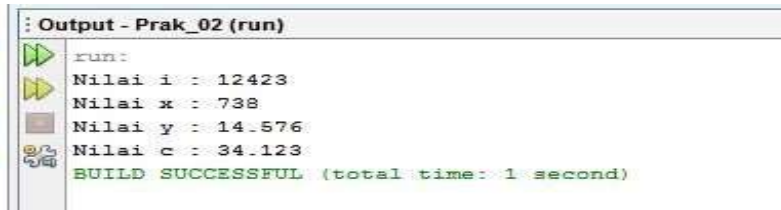
Contoh :

```
float jumlah;
const float PHI = 3.14;
```

Latihan_02 :

```
public class variable{  
    public static void main(String args[]){  
        // Deklarasi Variabel  
        int i;  
        long x;  
        double y;  
        float c;  
        // Pengisian Nilai  
        i = 12423;  
        x = 738;  
        y = 14.576;  
        c = 34.123f;  
        // Mencetak Nilai  
        System.out.println("Nilai i : "+i);  
        System.out.println("Nilai x : "+x);  
        System.out.println("Nilai y : "+y);  
        System.out.println("Nilai c : "+c);  
    }  
}
```

Hasil Output :



```
run:  
Nilai i : 12423  
Nilai x : 738  
Nilai y : 14.576  
Nilai c : 34.123  
BUILD SUCCESSFUL (total time: 1 second)
```

Gambar 2.2 Tampilan Output Class variable

2.3 Tugas

1. Buatlah program mengkonversi suhu dari satuan Celcius ke satuan lain (Reamur, Farenhet, dan Kelvin). Tentukan variabel, tipe data, konstanta, ekspresi dan penugasan apa saja yang ada dalam algoritma tersebut. Cari informasi mengenai rumus konversi suhu.
2. Buatlah program yang lengkap dengan variabel, tipe data, konstanta, ekspresi dan assignmet, untuk menghitung besar tabungan tiap bulan. Besar tabungan dihitung dengan cara tabungan awal ditambah besar bunga yang didapatkan dikurangi dengan biaya administrasi. Input berupa besar tabungan awal, besarnya prosentase bunga dan biaya administrasi tiap bulan, outputnya adalah besar tabungan akhir.

MODUL PRAKTIKUM 3

PROGRAM DENGAN OPERATOR

Modul praktikum 3 membahas tentang berbagai jenis operator pada pemrograman java dan mengenal statement input.

3.1 Operator

Operator merupakan tanda/symbol operasi. Operator digunakan untuk menuliskan suatu statemen dalam bahasa pemrograman. Dalam program keberadaan operator dalam suatu ungkapan dievaluasi. Hal ini di maksud untuk menentukan operator mana yang harus dikerjakan terlebih dahulu dan operator mana yang terakhir dikerjakan. Penetapan pengerjaan operator tersebut sudah diatur dalam prioritas operator.

Tabel 3.1 Prioritas operator

Order	Operator	Contoh
1	Operator unary	+, -, + +, - - dan %
2	Operator aritmatika dan geser	*, /, +, -, < <, > >
3	Operator relasional	>, <, > =, < =, =, !=
4	Operator logis dan bitwise	&&, , &, , ^
5	Operator tugas	=, * =, / =, + =, - =

Contoh :

$X = 2 + 3 * 4$

Berapakah nilai ungkapan itu? 14 atau 20? Berdasarkan table operator bisa kita tebak hasil ungkapan $2 + 3 * 4$ adalah 14. Hal ini dikarenakan operator perkalian memiliki keutamaan yang lebih tinggi dibandingkan operator penjumlahan. Sehingga $3 * 4$ dikerjakan terlebih dahulu, kemudian hasil ditambahkan dengan 2.

$X = (2 + 3) * 4$

Apabila sebuah operator ingin dikerjakan terlebih dahulu, maka letakkan operasi itu di dalam tanda kurung. Dengan demikian, nilai ungkapan **$X = 2 + 3 * 4$** tidak lagi 14 melainkan 20.

3.2 Pernyataan Masukan Dan Keluaran Dalam Java

Dalam bahasa pemrograman Java, kita akan menggunakan pernyataan masukan dan keluaran. Pernyataan masukan adalah pernyataan untuk mendapatkan masukan dari

keyboard. Sedangkan pernyataan keluaran adalah pernyataan untuk menampilkan sesuatu nilai ke layar.

1. Komponen keluaran

Untuk menampilkan ke layar secara tekstual, java mempunyai fasilitas output, yaitu dengan perintah :

```
System.out.print( hal yang akan ditampilkan, bisa lebih dari satu);
```

untuk menampilkan ke layar dan setelah selesai tidak berpindah baris. Akibatnya perintah keluaran berikutnya akan ditampilkan pada baris yang sama. Dan perintah: *System.out.println(hal yang akan ditampilkan, bisa lebih dari satu);* untuk menampilkan ke layar dan setelah selesai berpindah baris. Akibatnya perintah keluaran setelah itu akan ditampilkan pada baris berikutnya.

```
System.out.println("Hasil penjumlahan "+hasil);
```

Untuk menampilkan isi variabel hasil.

2. Komponen masukan

Untuk memasukkan sebuah nilai ke variabel yang sudah didefinisikan digunakan kelas Scanner.

- Input data bertipe Integer

Untuk menginputkan data dengan tipe integer digunakan method `nextInt` di dalam kelas Scanner.

- Input data bertipe String

Untuk menginputkan data dengan tipe string digunakan method `next` di dalam kelas Scanner.

- Input tipe yang lain

Untuk input data dari keyboard dengan tipe yang lain dan disediakan oleh kelas Scanner adalah sebagai berikut

- `nextBoolean` : input tipe boolean (true atau false saja)
- `nextShort` : input tipe short integer
- `nextLong` : input tipe long integer
- `nextFloat` : input tipe float
- `nextDouble` : input tipe double

3.3 Pelaksanaan Praktikum

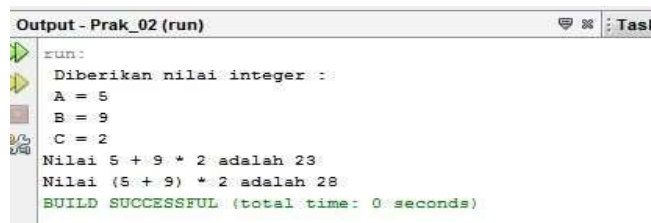
Buatlah program seperti berikut :

Latihan 1_1 :

1. Buka project baru. Kemudian buat class dengan nama latihan_1
2. Ketikkan lah listing program dibawah ini :

```
//Program aritmatika untuk menjumlahkan 3 buah bilangan
public class latihan_1 {
    public static void main (String[] args) {
        int A, B, C, hasil1, hasil2;
        A = 5; B= 9; C = 2; hasil1 = 5 + 9 * 2; hasil2 = (5 + 9) * 2;
        System.out.println(" Diberikan nilai integer :");
        System.out.println(" A = " +A);
        System.out.println(" B = " +B);
        System.out.println(" C = " +C);
        System.out.println("Nilai 5 + 9 * 2 adalah " +hasil1);
        System.out.println("Nilai (5 + 9) * 2 adalah " +hasil2);
    }
}
```

Hasil Output :



Gambar 3.1 Tampilan Output Class Prak_01

Latihan 1_2 :

1. Buatlah class baru dengan nama Jumlah
2. Kemudian ketikkan lah listing program dibawah ini :

```
// Program aritmatika untuk mengitung 2 buah bilangan yang diinput dari
keyboard

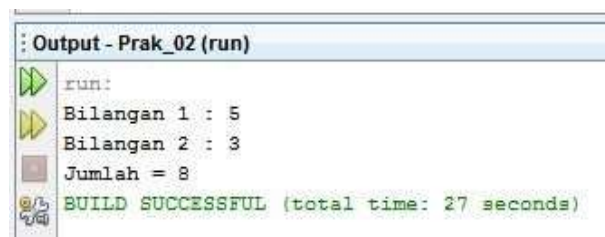
import java.util.Scanner;
public class Jumlah{
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
```

```

int A,B ;
System.out.print("Bilangan 1 : ");
A = masuk.nextInt();
System.out.print("Bilangan 2 : ");
B = masuk.nextInt();
System.out.println("Jumlah = " + (A + B));
}
}

```

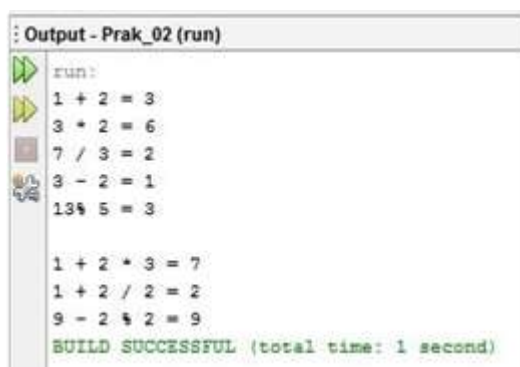
Hasil Output :



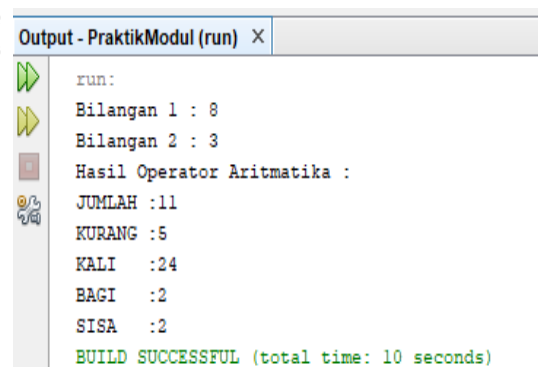
Gambar 3.2 Tampilan Output Class Jumlah

3.4 Tugas

Dengan menggunakan operator aritmatika buatlah program untuk menjumlahkan, mengalikan, membagi dan sisa pembagian dari 2 bilangan yang disimpan dan diinputkan dengan keyboard, output yang diinginkan sebagai berikut :



Hasil output No.1



Hasil output No.2

3. Buat program yang meminta inputan jari-jari, yang kemudian dihitung dan ditampilkan luas dan keliling lingkaran tersebut. (mis : input jari-jari = 4)

MODUL PRAKTIKUM 4

PROGRAM DENGAN KONDISI

Modul praktikum 4 membahas tentang pernyataan seleksi dalam pemrograman java dengan menggunakan statemen If, If – Else, dan Switch – Case.

4.1 Statemen If

Statemen if digunakan untuk menguji hasil dari suatu kondisi. Pada statemen **if** ini, hasil pengujian yang diharapkan adalah ungkapan Boolean yang benar (True) saja.

Tabel 4.1 Sintaks If

Sintaks if
<pre>If (a == 1) { System.out.println ("Nilai A = 1"); }</pre>
<pre>If (ekspresi-kondisi) { Pernyataan; }</pre>
Penjelasan : A merupakan kondisi yang diperiksa oleh if untuk menentukan apakah blok pada if dijalankan atau tidak, perintah-perintah pada blok if dijalankan jika kondisi A bernilai benar.

Tabel 4.2 Ekspresi logika

Ekspresi logika	Bernilai true jika
a == 1	a bernilai 1
a >= 1	a lebih besar atau sama dengan 1

4.2 Statemen If – Else

Penyeleksian if-else digunakan untuk mengecek kebenaran nilai sebuah ekspresi logika. Blok IF akan dijalankan jika ekspresi yang bernilai TRUE. Tapi jika salah, blok ELSE-lah yang akan dijalankan.

Tabel 4.3 Sintaks If-Else

Sintak if-else
<pre>if (ekspresi-kondisi) { Pernyataan; } else{ Pernyataan; }</pre>

Penjelasan :

Perintah dijalankan jika kondisi yang diperiksa IF benar ditempatkan pada blok IF(). Adapun perintah yang akan dijalankan jika kondisi yang diperiksa oleh IF tidak benar ditempatkan pada blok ELSE.

4.3 Statemen Switch – Case

Penyeleksian switch digunakan untuk memeriksa nilai-nilai yang mungkin dari sebuah ekspresi. Ekspresi yang diperiksa haruslah bertipe int, short, char. Sintak penyeleksian kondisi dengan switch adalah :

Tabel 4.4 Sintaks Switch – Case

Sintak switch case
<pre>switch (ungkapan) { case ungkapan1: pernyataan1; break; case ungkapan2: pernyataan2; break; default: pernyataan_x; }</pre>
<p>Penjelasan :</p> <ul style="list-style-type: none">- ungkapan1, ungkapan2 dan seterusnya dilakukan secara berurutan dimulai dari yang pertama, sekiranya cocok pernyataan yang mengikuti case dijalankan.- break untuk mengakhiri blok pernyataan.- default hanya akan dijalankan jika ungkapan pada bagian case tidak ada yang cocok.

4.4 Pelaksanaan Praktikum

Buatlah program berikut ini :

Latihan kondisi_01 :

1. Buatlah Project baru. Dan class baru dengan nama Prak_02
2. Kemudian ketikkan lah listing program dibawah ini :

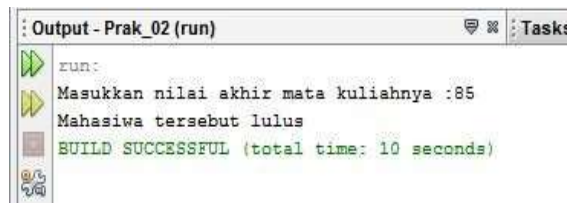
```
// Program untuk menyeleksi nilai akhir mata kuliah dengan
menggunakan if – else
import java.util.Scanner;
public class if {
```

```

public static void main (String[] args) {
    Scanner masukan = new Scanner(System.in);
    int nilai;
    System.out.print("Masukkan nilai akhir mata kuliahnya :");
    nilai = masukan.nextInt();
    if (nilai < 55)
        System.out.println("Mahasiwa tersebut tidak lulus");
    Else
        System.out.println("Mahasiwa tersebut lulus");
    }
}

```

Hasi Ouput :



Gambar 4.1 Tampilan Output Class if

Penjelasan program :

Jika diinputkan nilai < 55, maka akan muncul keterangan Siswa tersebut tidak lulus, tetapi jika nilai > atau = 55, maka muncul keterangan Siswa tersebut lulus.

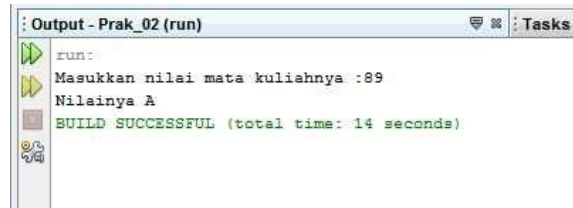
Latihan kondisi_02:

```

// Program untuk menyeleksi nilai akhir mata kuliah menggunakan if - else
import java.util.Scanner; public
class if - else {
    public static void main (String[] args) {
        Scanner masukan = new Scanner(System.in);
        System.out.print("Masukkan nilai mata kuliahnya :");
        int nilai = masukan.nextInt();
        if (nilai >= 80)
            System.out.println("Nilainya A");
        else if (nilai >= 70)
            System.out.println("Nilainya B");
        else if (nilai >= 55)
            System.out.println("Nilainya C");
        else if (nilai >= 40)
            System.out.println("Nilainya D");
        else
            System.out.println("Nilainya E");
    }
}

```

Hasil output :

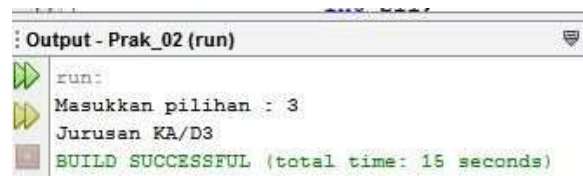


Gambar 4.2 Tampilan Output Class if - else

Latihan kondisi_03 :

```
// Program untuk menyeleksi dengan menu pilihan menggunakan Switch -  
Case  
import java.util.Scanner;  
public class CaseJurusan {  
    public static void main(String args[]){  
        Scanner masuk = new Scanner(System.in);  
        int pil;  
        System.out.print("Masukkan pilihan : ");  
        pil = masuk.nextInt();  
        switch (pil) {  
            case 1:  
                System.out.println("Jurusan SI/D3");  
                break;  
            case 2:  
                System.out.println("Jurusan TK/D3");  
                break;  
            case 3:  
                System.out.println("Jurusan KA/D3");  
                break;  
            case 4:  
                System.out.println("Jurusan TI/S1");  
                break;  
            case 5:  
                System.out.println("Jurusan SI/S1");  
                break;  
            default:  
                System.out.println("Pilihan Salah!!!");  
                break;  
        }  
    }  
}
```

Hasil Output :

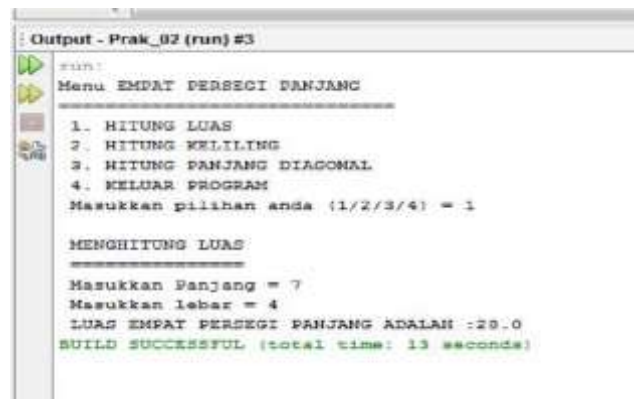


```
Output - Prak_02 (run)
Run:
Masukkan pilihan : 3
Jurusan KA/D3
BUILD SUCCESSFUL (total time: 15 seconds)
```

Gambar 4.3 Tampilan Output Class CaseJurusan

4.5 Tugas

1. Buatlah program inputan memilih menu empat persegi panjang dan melakukan proses perhitungan menggunakan perulangan **switch case**. Output seperti gambar dibawah ini :



```
Output - Prak_02 (run) #3
Run:
Menu EMPAT PERSEGI PANJANG
1. HITUNG LUAS
2. HITUNG KELILING
3. HITUNG PANJANG DIAGONAL
4. KELUAR PROGRAM
Masukkan pilihan anda (1/2/3/4) = 1

MENHITUNG LUAS
Masukkan Panjang = 7
Masukkan lebar = 4
LUAS EMPAT PERSEGI PANJANG ADALAH : 28.0
BUILD SUCCESSFUL (total time: 13 seconds)
```

2. Sebuah dealer memberikan daftar harga per merk berikut adalah daftar harga motor di dealer tersebut:
 - a. Honda {Supra x 125 : 15000000, Megapro : 18000000, Tiger : 25000000 }
 - b. Yamaha {Jupiter : 17000000, Mio : 13000000, Vixion : 22000000 }
 - c. Suzuki {Nex : 12000000, Satria FU : 18000000, Smash : 13000000 }Dari ketentuan diatas buatlah kode program untuk mengetahui harga motor yang di pilih pelanggan !

MODUL PRAKTIKUM 5

PROGRAM DENGAN PERULANGAN

Modul praktikum 5 membahas tentang pemrograman java menggunakan perulangan While, Do – While, dan For.

5.1 Perulangan (While)

Perulangan while digunakan untuk mengulangi eksekusi sebuah blok selama kondisi tertentu masih berlaku (bernilai TRUE).

Sintaks bentuk perulangan while dinyatakan sebagai berikut :

Tabel 5.1 Sintaks while

<pre>while (kondisi) { //body while }</pre>
<p>Penjelasan :</p> <p>Kondisi menentukan kapan looping while berhenti. Looping akan terus dilakukan sampai kondisi bernilai salah.</p> <p>Body while isi dari statement-statement yang ingin di looping</p>

5.2 Perulangan Do – While

Perulangan do-while digunakan untuk mengeksekusi sebuah blok selama kondisi tertentu. Perulangan ini mirip dengan perulangan while. Bedanya adalah pada perulangan do-while tidak diperiksa dulu. Perulangan while, semua perulangan diperiksa, termasuk perulangan yang pertama kali.

Tabel 5.2 Sintaks Do-While

<pre>do { //body while } While (A);</pre>
<p>Penjelasan :</p> <p>A merupakan kondisi (ekspresi logika) yang menentukan kapan looping while berhenti. Selama A bernilai benar, looping akan terus dilakukan.</p> <p>Body while isi dengan statement-statement yang ingin di looping.</p>

5.3 Perulangan For

Perulangan for digunakan untuk mengulang eksekusi sebuah blok beberapa kali.

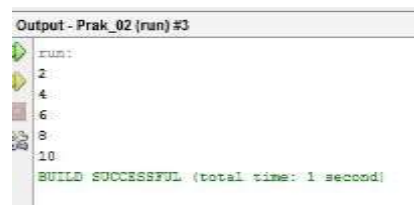
Tabel 5.3 Sintaks For

<pre>For (ekspresi awal, kondisi, ekspresi penilaian) { Statement aksi; }</pre>
<p>Penjelasan :</p> <p>ekspresi awal merupakan perintah yang pertama kali dijalankan oleh for sebelum melakukan looping.</p> <p>Kondisi merupakan kondisi untuk menentukan kapan looping for berhenti.</p> <p>Looping akan terus terjadi sampai kondisi bernilai salah.</p> <p>ekspresi penilaian digunakan untuk menaikkan nilai variable perulangan.</p> <p>Statement aksi blok tempat dimana untuk menulis program yang ingin di looping.</p>

5.4 Pelaksanaan Praktikum Latihan perulangan_01 :

```
// Program perulangan untuk mengurutkan nilai bilangan genap dari 2 – 10
menggunakan While
public class perulangan_01 {
    public static void main(String args[]) {
        int bil;
        bil=2;
        while (bil<=10) {
            System.out.println(bil);
            bil+=2;
        }
    }
}
```

Hasil Output :



Gambar 5.1 Tampilan Output Class perulangan_01

Latihan perulangan_02 :

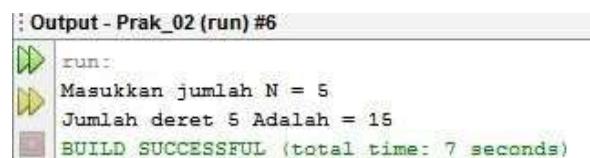
```
// Program perulangan untuk menghitung nilai factorial yang diinput dari keyboard
menggunakan Do - While
import java.util.Scanner;
public class perulangan_02 {
```

```

public static void main(String args[]) {
    int N, i, jumlah;
    i = 1; jumlah = 0;
    Scanner masuk = new Scanner(System.in);
    System.out.print("Masukkan jumlah N = ");
    N = masuk.nextInt();
    do {
        jumlah = jumlah + i;
        ++;
    } while ( i <= N);
    System.out.println("Jumlah deret"+N+" Adalah = "+jumlah);
}
}

```

Hasil Output :



```

Output - Prak_02 (run) #6
run:
Masukkan jumlah N = 5
Jumlah deret 5 Adalah = 15
BUILD SUCCESSFUL (total time: 7 seconds)

```

Gambar 5.2 Tampilan Output Class perulangan_02

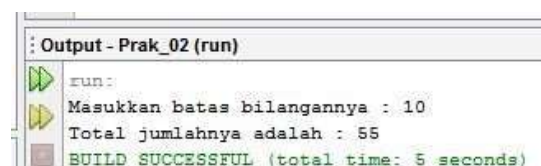
Latihan perulangan_03

```

// Program perulangan untuk mengitung nilai yang diinput dari keyboard menggunakan For
import java.util.Scanner;
public class perulangan_03 {
    public static void main (String[] args) {
        Scanner masukan = new Scanner(System.in);
        System.out.print("Masukkan batas bilangannya : ");
        int batas = masukan.nextInt();
        int hasil = 0;
        for(int i=1; i<=batas; i++)
            hasil += i;
        System.out.println("Total jumlahnya adalah : " + hasil);
    }
}

```

Hasil Output :



```

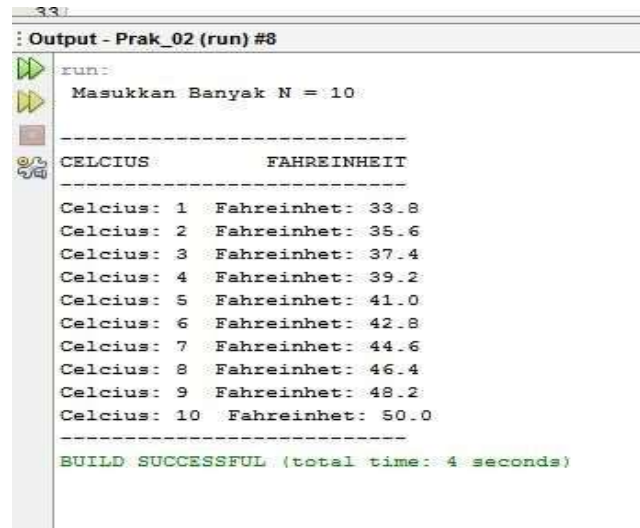
Output - Prak_02 (run)
run:
Masukkan batas bilangannya : 10
Total jumlahnya adalah : 55
BUILD SUCCESSFUL (total time: 5 seconds)

```

Gambar 5.3 Tampilan Output Class perulangan_03

5.5 Tugas

1. Buatlah program mencetak konversi suhu dari celcius ke fahrenheit mulai dari 1 sampai 10 dengan membuat tabel. Hasil output seperti gambar dibawah ini :



```
33/
: Output - Prak_02 (run) #8
run:
Masukkan Banyak N = 10

-----
CELCIUS      FAHREINHEIT
-----
Celcius: 1   Fahrenheit: 33.8
Celcius: 2   Fahrenheit: 35.6
Celcius: 3   Fahrenheit: 37.4
Celcius: 4   Fahrenheit: 39.2
Celcius: 5   Fahrenheit: 41.0
Celcius: 6   Fahrenheit: 42.8
Celcius: 7   Fahrenheit: 44.6
Celcius: 8   Fahrenheit: 46.4
Celcius: 9   Fahrenheit: 48.2
Celcius: 10  Fahrenheit: 50.0
-----

BUILD SUCCESSFUL (total time: 4 seconds)
```

2. Buatlah sebuah program untuk menampilkan deret bilangan prima, dengan memasukkan batas atas dan batas bawah.

MODUL PRAKTIKUM 6

PROGRAM DENGAN FUNGSI (METHOD)

Modul praktikum 6 membahas tentang fungsi pada program Java dan fungsi yang mengembalikan suatu nilai, tidak mengembalikan nilai dan fungsi rekursi.

6.1 Fungsi (Method)

Fungsi adalah satuan-satuan, seperti program untuk melakukan tugas-tugas tertentu yang menghasilkan suatu nilai untuk dikembalikan ke program pemanggil. Dalam suatu program, fungsi dibuat terpisah dengan program pemanggil (**main()**). Tabel 6.1 Sintak mendeklarasikan fungsi

```
Access_specifier modifier datatype method_name  
(parameter_list)  
{  
    // badan fungsi  
}
```

Penjelasan :

Access_specifier : batas akses untuk fungsi. Batas akses bias berupa (public, protected, private).

Modifier : penetapan kita tentang perilaku tugas untuk fungsi itu. Seperti (static, abstract, final, native).

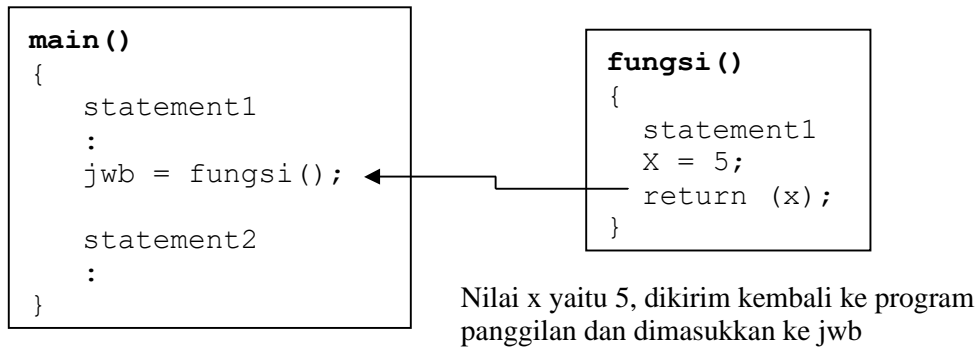
Datatype : tipe data dari nilai yang dikembalikan fungsi. Jika tidak ada nilai yang dikembalikan, tipe data ditulis **void**.

method_name : nama dari fungsi

parameter_list : berisi nama parameter yang sedang diberikan kepada fungsi, dan tipe datanya. Tanda koma memisahkan berbagai parameter.

6.2 Fungsi Yang Menghasilkan Nilai

Fungsi yang bias menghasilkan nilai merupakan suatu fungsi yang tidak membutuhkan argument, tetapi fungsi itu bias menghasilkan nilai. Untuk membuat fungsi seperti itu, maka didalam fungsi yang bisa menghasilkan nilai itu perlu ditambahkan satu bari untuk statement **return()**. Fungsi yang menghasilkan nilai sama dengan fungsion yang mengembalikan nilai adalah suatu sub rutin yang bila dipanggil oleh suatu program (argumen) maka argumen tersebut akan memperoleh nilai balikan dari function tersebut. Atau dengan kata lain, suatu function yang mempunyai nilai.



Gambar 6.1 fungsi menghasilkan/mengirimkan nilai ke program utama

Statement return()

Kegunaan statemen return() ada dua, yaitu :

1. Pelaksanaan statemen ini akan langsung mengalihkan control dari fungsi ke program panggilannya.
2. Apapun yang ada didalam tanda kurung buka-tutup dibelakang return akan dikirim kembali sebagai suatu nilai kepada program pemanggilnya.

6.3 Fungsi Yang Menerima Nilai

Fungsi yang menerima nilai yaitu fungsi yang bias menerima nilai dari fungsi lain. Fungsi yang menerima nilai sama dengan fungsi yang tidak mengembalikan nilai bentuknya sangat mirip dengan function yang mengembalikan nilai. Perbedaannya adalah penggunaan kata kunci atau klausa **void** pada fungsi yang tidak mengembalikan nilai baik pada bahasa C++ maupun bahasa Java.

6.4 Fungsi Rekursif

Rekursi merupakan kemampuan yang memiliki suatu fungsi untuk memanggil dirinya sendiri, secara langsung maupun tidak langsung. Pemanggilan dirinya sendiri mempunyai arti bahwa didalam fungsi tersebut terdapat statemen yang memanggil. Dirinya sendiri. Berikut ini adalah contoh dari konsep **recurse()** yang berkemampuan untuk memanggil dirinya sendiri secara langsung dan tidak langsung.

Fungsi rekursi secara langsung	Fungsi rekursi secara tidak langsung
<pre> recurse ([parameter]) { // statemen; Recurse([parameter]) // statemen; } </pre>	<pre> recurse ([parameter]) { // statemen; test([parameter]) // statemen; } </pre>

6.5 Pelaksanaan Prektikum Latihan_01 :

```
// PROGRAM METHOD YANG MENGEMBALIKAN SUATU NILAI
import java.util.*;
public class method_01 {
    public static void main (String[] args) {
        Scanner masukan = new Scanner(System.in);
        int bil1, bil2, bil3;
        System.out.println("Program Menjumlahkan 3 bilangan");
        System.out.print("Masukkan bilangan 1 : ");
        bil1 = masukan.nextInt();
        System.out.print("Masukkan bilangan 2 : ");
        bil2 = masukan.nextInt();
        System.out.print("Masukkan bilangan 3 : ");
        bil3 = masukan.nextInt();
        System.out.println();
        System.out.println("HASIL BILANGAN : ");
        System.out.println("===== ");
        System.out.println();
        System.out.println("Jumlah          bil1      +      bil2      : 
"+(jumlah(bil1,bil2)));
        System.out.println("Jumlah          bil2      +      bil3      : 
"+(jumlah(bil2,bil3)));
        System.out.println("Jumlah          bil1      +      bil3      : 
"+(jumlah(bil1,bil3)));
    }
    private static int jumlah(int a, int b){
        int hasil = a + b;
        return hasil;
    }
}
```

Hasil Output :



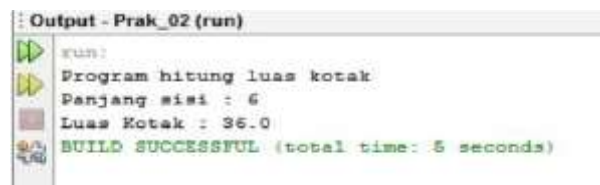
```
Output - Prak_02 (run)
Program Menjumlahkan 3 bilangan
Masukkan bilangan 1 : 5
Masukkan bilangan 2 : 8
Masukkan bilangan 3 : 3
HASIL BILANGAN :
=====
Jumlah bil1 + bil2 : 13
Jumlah bil2 + bil3 : 11
Jumlah bil1 + bil3 : 8
BUILD SUCCESSFUL (total time: 9 seconds)
```

Gambar 6.2 Tampilan Output Class method_01

Latihan_02 :

```
// PROGRAM METHOD YANG TIDAK MENGEMBALIKAN SUATU NILAI
import java.util.*;
public class method_02 {
    static void luasKotak(int sisi){
        System.out.println("Luas Kotak : "+Math.pow(sisi,2));
        // Untuk Menghitung Luas Kotak
    }
    public static void main(String[ ] args){
        Scanner input=new Scanner(System.in);
        System.out.println("Program hitung luas kotak");
        System.out.print("Panjang sisi : ");
        int s=input.nextInt();
        luasKotak(s);
    }
}
```

Hasil Output :



Gambar 6.3 Tampilan Output Class method_02

Latihan_03 :

```
// PROGRAM FUNGSI RAKURSI
import java.util.Scanner;
public class rekursif_03 {
    public static long rekursif (int a, int b){
        if (b==0){
            return 0;
        } else {
            return a+rekursif(a,b-1);
        }
    }

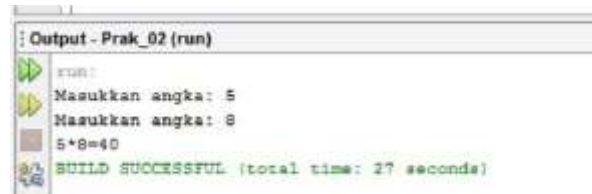
    public static void main (String[] args) {
        int a,b; //deklarasi variabel a dan b
        Scanner input = new Scanner(System.in);
        //proses input data dengan scanner
        System.out.print("Masukkan angka: ");
        a=input.nextInt();//pengambilan data a
        System.out.print("Masukkan angka: ");
        b=input.nextInt();//pengambilan data b
    }
}
```

```

        System.out.println(a+"*"+b+"="+ rekursif (a,b)); //tampilan output
    }
}

```

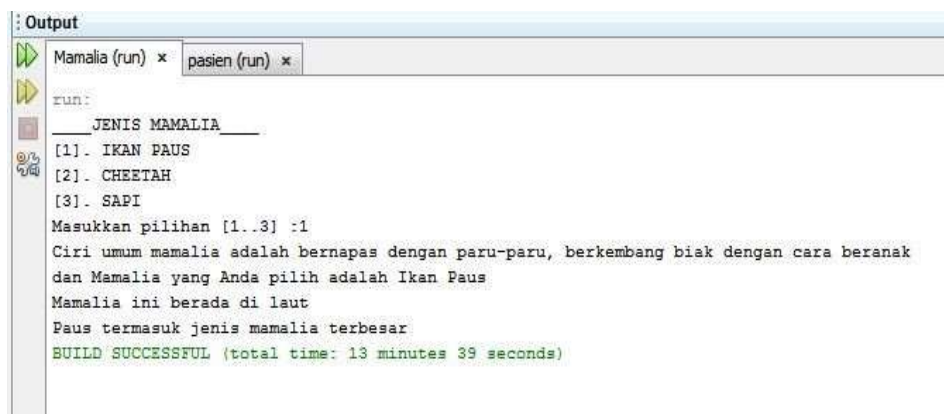
Hasil Output :



Gambar 6.4 Tampilan Output Class rekursif_03

6.5 Tugas

1. Buatlah sebuah program menu yang terdiri sub-sub menu pilihan yang dapat menjelaskan cirri-ciri dari sub menu yang dipilih. Hasil output seperti gambar dibawah ini :



2. Buatlah program untuk mengitung nilai akhir mata kuliah ditentukan oleh komposisi empat nilai :

- ✓ 25 % TUGAS, 20 % QUIZ, 20 % MID, 35 % UAS,
- ✓ Dan buatlah Grade :
 - nilai 0 - 39 dengan grade = E
 - nilai 40 - 55 dengan grade = D
 - nilai 56 - 79 dengan grade = C
 - nilai 80 - 89 dengan grade = B
 - nilai 90 - 100 dengan grade = A

MODUL PRAKTIKUM 7

PROGRAM DENGAN ARRAY

Modul praktikum 7 membahas tentang penggunaan Array untuk dapat mendukung sejumlah metode dalam pemanipulasian array.

7.1 Array

Array dibagi menjadi beberapa dimensi, diantaranya sebagai berikut :

7.1.1 Array Satu Dimensi

Array adalah sebuah struktur data yang terdiri dari data yang bertipe sama. Ukuran larik bersifat tetap, larik akan mempunyai ukuran yang sama pada saat sekali dibuat. Larik dalam Java adalah obyek, disebut juga sebagai tipe referensi. Sedangkan elemen dalam larik Java bisa primitif atau referensi. Posisi dari larik biasa disebut sebagai elemen. Elemen larik dimulai dari 0 (nol). Penyebutan larik diberikan dengan cara menyebutkan nama lariknya dan diikuti dengan indeksinya, dimana indeks dituliskan diantara tanda kurung siku. Gambar 1. memperlihatkan gambaran larik dengan 10 elemen, dimana setiap elemennya bertipe integer, dengan nama A.

Nama	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
Isi larik	12	-56	23	45	-16	-2	85	41	15	20

Gambar 7.1 Array satu-dimensi dengan 10 elemen

A. Deklarasi Dan Menciptakan Larik

Sebagai sebuah obyek, larik harus diciptakan dengan menggunakan kata cadang new. Deklarasi dan penciptaan variabel larik gambar 1 adalah sebagai berikut.

```
int A[] = new int[10];
```

larik dideklarasikan dan langsung diciptakan . Atau

```
int A[];
```

```
A = new int[10];
```

larik dideklarasikan, baru pada pernyataan berikutnya larik diciptakan.

7.1.2 Array Dua Dimensi

Array dua dimensi merupakan suatu array yang dibangun menurut format baris dan kolom. Seperti gambar dibawah ini, untuk mengakses variable dua dimensi diawali baris ke 0 hingga ke kolom 3, selanjutnya dari baris ke 1 bergerak ke kolom 0 hingga ke kolom 3, dan

seterusnya hingga baris ke 2. Array dua dimensi seperti gambar dibawah ini mempunyai baris sebanyak 3 dan kolom sebanyak 4.

Two_D

	0	1	2	3
0	1	3	2	1
1	2	1	3	2
2	3	2	1	5

← Baris

↑
Kolom

Gambar 7.2 Array dua-dimensi ukuran 3x4

Variable elemen array dua dimensi berjenis integer, maka deklarasikan dua dimensi dengan cara inisialisasi nilai-nilai elemen array bias kita tuliskan sebagai berikut :

```
Int two_D [ ] [ ] = {
    {1, 3, 2, 1},
    {2, 1, 3, 2},
    {3, 2, 1, 5}
};
```

Artinya bahwa two_D [0] [0] elemen array_nya akan mempunyai nilai 1, two_D[1] [2] elemen array nya akan mempunyai nilai **3**, dan two_D[2] [3] elemen array_nya akan mempunyai nilai **5**.

7.1.3 Array Tiga Dimensi

Array tiga dimensi merupakan suatu array dari array dari array. Array dari array itu sendiri tidak lain merupakan array dua-dimensi. Sehingga array tiga-dimensi dapat dinyatakan juga sebagai array dari array dua dimensi.

	Kolom Subskrip 0				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	1	2	3	4	
	5	6	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	9	3	2	
	4	6	8	3	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				
	2				
Subskrip 2	0	1	2	3	
	7	2	6	5	
	0	4	7	8	
	Kolom Subskrip 2				
	0				
	1				


```
int three_D [ ][ ] = {
    {
        {1, 2, 3, 4},
        {5, 6, 7, 8},
    },
    {
        {7, 9, 3, 2},
        {4, 6, 8, 3},
    },
    {
        {7, 2, 6, 5},
        {0, 1, 9, 4},
    }
}
```

Artinya bahwa jika kita mendklarasikan array `three_D[0][0][0]` maka elemen array_nya mempunyai nilai integer **1**. Selanjutnya, jika kita mendklarasikan array `three_D [1][1][1]` maka elemen array_nya mempunyai nilai integer **6** dan jika kita mendklarasikan array `three_D[2][0][1]` maka elemen array_nya mempunyai nilai integer **2**.

7.2 Pelaksanaan Praktikum

Latihan Array_01 :

```
// Program untuk menghitung deret bilangan yang diinput dari keyboard
menggunakan array satu dimensi
import java.util.Scanner;
public class array_01 {
    public static void main (String[] args) {
        Scanner masukan =
            new Scanner(System.in);
        int jum_bil, total=0;
        double rata2;
        System.out.print("Berapa banyaknya bilangan ? ");
        jum_bil = masukan.nextInt();
        int data[] = new int[jum_bil];
        for(int i=0;i<jum_bil;i++){
            System.out.print("Masukkan data ke-"+(i+1)+" : ");
            data[i] = masukan.nextInt();
            total = total + data[i];
        }
        System.out.print("Nilai rata-rata dari : ");
        for(int i=0;i<(jum_bil-1);i++)
            System.out.print(data[i] + "+");
```

```

        System.out.print(data[jum_bil-1] + " adalah ");
rata2 = total/jum_bil;
System.out.println(rata2);
    } }

```

Hasil Output :



```

run:
Berapa banyaknya bilangan ? 5
Masukkan data ke-1 : 2
Masukkan data ke-2 : 1
Masukkan data ke-3 : 4
Masukkan data ke-4 : 5
Masukkan data ke-5 : 6
Nilai rata-rata dari : 2+1+4+5+6 adalah 3.0
BUILD SUCCESSFUL (total time: 24 seconds)

```

Gambar 7.4 Tampilan Output Class array_01

Latihan Array_02 :

```

//program untuk menampilkan nilai matrik dengan menggunakan array 2
dimensi
import java.util.Scanner; public class array_02 {
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        int jum_bil1, jum_bil2;
        System.out.print("Masukkan Bilangan 1: ");
        jum_bil1 = masuk.nextInt();
        System.out.print("Masukkan Bilangan 2: ");
        jum_bil2 = masuk.nextInt();
        int hasil[][] = new int[jum_bil1][jum_bil2];
        System.out.println("Masukkan data nilai");
        for (int i = 0; i < jum_bil1; i++) {
            for (int j = 0; j < jum_bil2; j++){
                System.out.print("(" + (i + 1) + " , " + (j+1)
                    + ")" + " : ");
                hasil[i][j]=masuk.nextInt();
            }
        }
        System.out.println("Data nilai yang dimasukan");
        for (int i = 0; i < jum_bil1; i++){
            for (int j = 0; j < jum_bil2; j++)
                System.out.print(hasil[i][j]+" ");
            System.out.println();
        }
    } }

```

Hasil Output :

```

Output - Prak_02 (run) #2
RUN:
Masukkan Bilangan 1: 3
Masukkan Bilangan 2: 2
Masukkan data nilai
(1, 1) : 2
(1, 2) : 3
(2, 1) : 4
(2, 2) : 5
(3, 1) : 6
(3, 2) : 2
Data nilai yang dimasukan
2 3
4 5
6 2
BUILD SUCCESSFUL (total time: 16 seconds)

```

Gambar 7.5 Tampilan Output Class array_02

7.3 Tugas

1. Buatlah sebuah program penjumlahan matriks menggunakan array dua dimensi.
2. Buatlah program inputan pemakaian array dua dimensi untuk menghitung perkalian dua bilangan integer yang sama. Untuk hasil output seperti gambar dibawah ini :

```

Output
Prak_02 (run) x Prak_02 (run) #2 x
RUN:
Masukkan Bilangan: 6
Tabel Perkalian [0...6] X [0...6]
x      0      1      2      3      4      5
0      0      0      0      0      0      0
1      0      1      2      3      4      5
2      0      2      4      6      8      10
3      0      3      6      9      12      15
4      0      4      8      12      16      20
5      0      5      10     15      20      25
BUILD SUCCESSFUL (total time: 3 seconds)

```

MODUL PRAKTIKUM 8

PROGRAM DENGAN STRING

Modul praktikum 8 membahas tentang penggunaan string untuk dapat mendukung sejumlah metode dalam manipulasi string.

8.1 Fungsi String

String merupakan suatu bentuk array satu dimensi bertipe data character. Biasanya string digunakan untuk mengurutkan dan memanipulasi teks, misalnya kata-kata, nama dan berbagai kalimat.

Tabel 8.1 Konstruktor Klas String

Konstruktor	Keterangan
String()	Menciptakan obyek string yg berisi string kosong (jumlah karakter = 0)
String(char [] v)	Menciptakan obyek string yg berisi string yg berasal dari array yg dirujuk oleh v
String(String v)	Menciptakan obyek string yg isinya sama dengan obyek string argumennya

Metode dalam klas string memperlihatkan sejumlah metode penting, seperti :

- copyValueOf(char data[])
- copyValueOf(char data[], int offset, int jum)
- valueOf(boolean b)
- valueOf(double c)
- concat(String s)
- length()
- trim(), dan lain-lain.

Klas StringBuffer adalah klas yg menyimpan string yang konstan, begitu obyek string telah diciptakan maka string tidak dapat diubah. Konstruktor klas ini antara lain :

- StringBuffer() digunakan untuk menciptakan StringBuffer yang kosong
- StringBuffer(int n) digunakan untuk menciptakan StringBuffer dengan n karakter
- StringBuffer(String s) digunakan untuk menciptakan StringBuffer dengan string berupa s

Fungsi – fungsi pengelolaan string :

- **toUpperCase()** digunakan untuk menghasilkan string yang disusun dalam huruf kecil menjadi string dengan huruf besar (KAPITAL).
- **toLowerCase()** digunakan untuk menghasilkan string yang disusun dengan huruf besar menjadi string dengan huruf kecil.
- **Equals()** digunakan untuk membandingkan 2 obyek string dengan pengambilan true atau false.

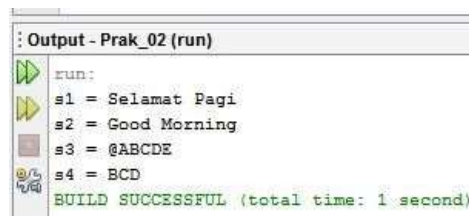
8.2 Pelaksanaan Praktikum

Tuliskan program berikut ini :

Latihan String_01 :

```
// Program untuk memasukkan nilai string
public class string_01 {
    public static void main(String args[]){
        byte data[] = new byte[6];
        data[0] = 64;      data[1] = 65;  data[2] = 66;
        data[3] = 67;      data[4] = 68;  data[5] = 69;
        String s1 = "Selamat Pagi";
        String s2 = new String("Good Morning");
        String s3 = new String(data);
        String s4 = new String(data, 2, 3);
        System.out.println("s1 = " + s1);
        System.out.println("s2 = " + s2);
        System.out.println("s3 = " + s3);
        System.out.println("s4 = " + s4);
    }
}
```

Hasil Output :



Gambar 8.1 Tampilan Output Class string_01

Pada program di atas, pernyataan seperti :

String s1 = "Selamat Pagi";

Sebenarnya identik dengan :

```
String s1 = new String("Selamat Pagi");
```

```
String s3 = new String(data);
```

Akan membuat string yang tersusun atas karakter-karakter yang nilainya sama seperti elemen-elemen pada array data, maka s3 berisi string @ABCDE adalah karakter @ = 64, A=65 dan seterusnya.

```
String s4 = new String(data, 2, 3);
```

Angka 3 menyatakan jumlah karakter yg menyusun string dan angka 2 menyatakan karakter pertama pada string, hasil diambil pada indeks ke-2 array.

Latihan String_02

```
// Program untuk menggabungkan 2 buah string dan menghitung panjang string
public class string_02 {
    public static void main(String args[]){
        String str1, str2, gabung ;
        str1 = "Selamat Belajar";
        str2 = "Program Java";
        gabung = str1.concat(str2);
        System.out.println("Diberi String : "+str1);
        System.out.println("Diberi String : "+str2);
        System.out.println("Diberi String : "+gabung);
        System.out.println("Panjang String Adalah : "+str1.length());
    }
}
```

Untuk menggabungkan string str1 dan str2 menggunakan fungsi **concat()**

```
gabung = str1.concat(str2);
```

length() : digunakan untuk menghasilkan panjang suatu string

Hasil Output :



```
Output - Prak_02 (run)
run:
Diberi String : Selamat Belajar
Diberi String : Program Java
Diberi String : Selamat BelajarProgram Java
Panjang String Adalah : 27
BUILD SUCCESSFUL (total time: 1 second)
```

Gambar 8.2 Tampilan Output Class string_02

8.3 Tugas

1. Buatlah sebuah program string dengan mengimplementasikan fungsi **toUpperCase()** dan **toLowerCase()**