

MODUL PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK



JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH SIDOARJO

2013

Tata Tertib :

1. Praktikum dilaksanakan tepat waktu sesuai dengan jadwal yang ditetapkan.
2. Mahasiswa yang terlambat datang atau absent harus memberikan surat/bukti yang dapat dipercaya (surat dokter atau surat keterangan kerja dari perusahaan).
3. Mahasiswa diperkenankan pindah kelompok/jam/hari praktikum dengan syarat mengkonfirmasi 1 minggu sebelum pelaksanaan praktikum melalui Koordinator Praktikum dan Kepala Lab.
5. Mahasiswa harus berbusana sopan dan rapi (tidak diperkenankan memakai kaos oblong dan sandal atau sepatu sandal)
6. Mahasiswa harus meminjam alat praktikum dengan cara mengisi lembaran bon pinjaman alat yang tersedia
7. Praktikum dianggap selesai jika mahasiswa telah menyerahkan laporan sementara dan alat yang dipinjam dalam keadaan baik, bersih, dan rapi.
8. Kerusakan alat yang dipinjam oleh mahasiswa menjadi tanggung jawab penuh kelompok mahasiswa yang bersangkutan.
9. Selama praktikum berlangsung, mahasiswa dilarang merokok makan, bergurau, bermain alat, menghidupkan Hand Phone, atau pun keluar masuk ruangan tanpa seijin dosen pembimbing/asisten pendamping.
10. Setelah melakukan praktikum, mahasiswa harus membuat laporan sementara hasil pengamatan praktikum rangkap dua dan menyerahkan kepada dosen pembimbing/asisten pada saat meninggalkan ruangan untuk ditanda tangani. (YANG NANTINYA DILAMPIRKAN DALAM LAPORAN AKHIR).
11. Mahasiswa yang tidak melaksanakan praktikum 1 Modul dinyatakan tidak lulus.
12. Laporan Akhir Praktikum Covernya menggunakan Standart Fakultas dan Laporan diserahkan 2 minggu setelah jadwal masing-masing kelompok.
13. Apabila Laporan diserahkan lebih dari 2 minggu maka dinyatakan tidak lulus dan laporan Praktikum diserahkan ke koordinator praktikum dan kepala lab.
14. Mahasiswa yang dinyatakan tidak lulus Praktikum harus mengulang dijadwal praktikum berikutnya dengan membayar biaya praktikum yang telah ditentukan oleh Universitas melalui bank yang ditunjuk oleh UMSIDA.

POKOK BAHASAN 1

ELEMEN DASAR JAVA

PENDAHULUAN

Pada pokok bahasan ini berisi penjelasan disertai contoh mengenai pengenalan type data, variabel, dan identitas array yang menjadi pemahaman dasar bagi mahasiswa sebelum mempelajari Pemrograman Berorientasi Objek dengan membuat program java sederhana menggunakan elemen dasar tersebut, diharapkan mahasiswa dapat :

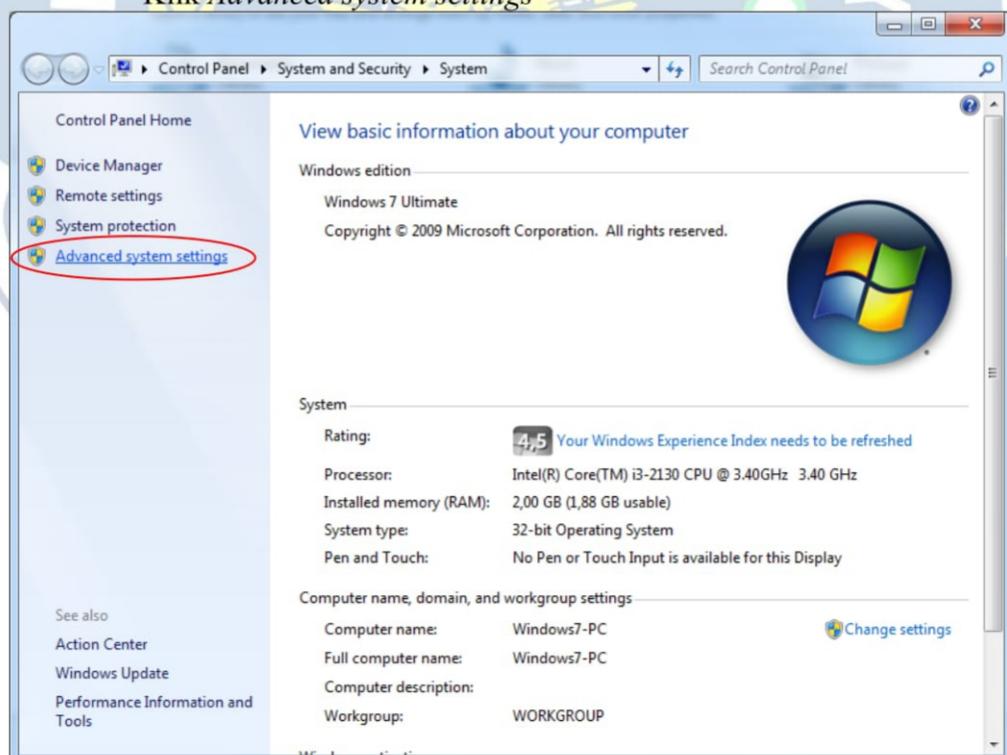
- a. Memahami instalasi java
- b. Mengenali type data, variable dan identitas dan array.
- c. Menggunakan elemen dasar java pada pembuatan pemograman
- d. membuat program java sederhana dengan menggunakan elemen dasar.

PENYAJIAN (TUTORIAL)

A. Instalasi Java

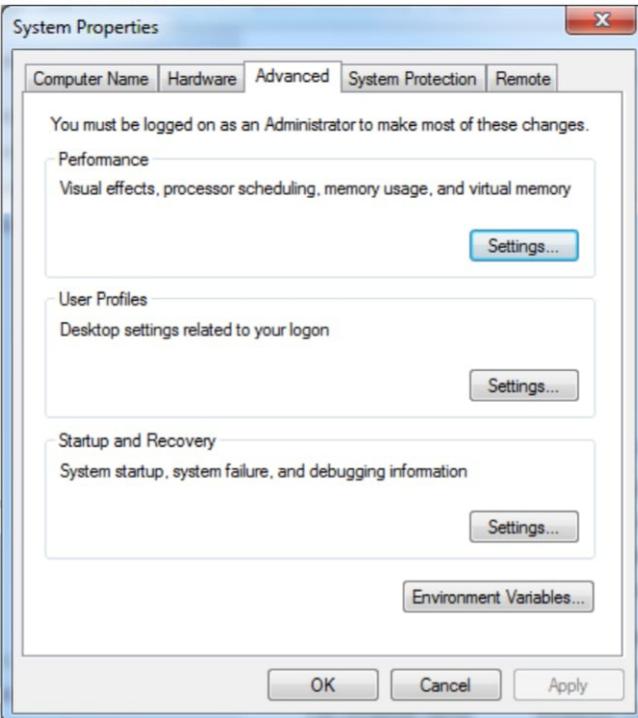
Instal aplikasi Java SDK pada komputer. Berikut langkah-langkah untuk instalasi java:

- a. Jalankan *installer* Java SDK
- b. Ikuti semua proses sampai selesai
- c. Set variabel system path pada komputer. Caranya:
 - Buka *Windows Explorer*
 - Klik kanan *icon My Computer*, klik *Properties*
 - Klik *Advanced system settings*



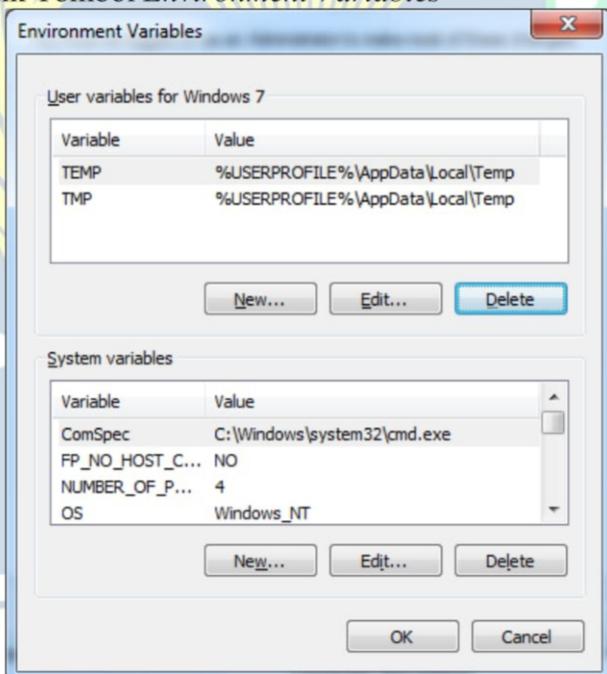
Gambar 1.1 Gambar System properties dalam Windows 7

- Klik tab *Advanced*



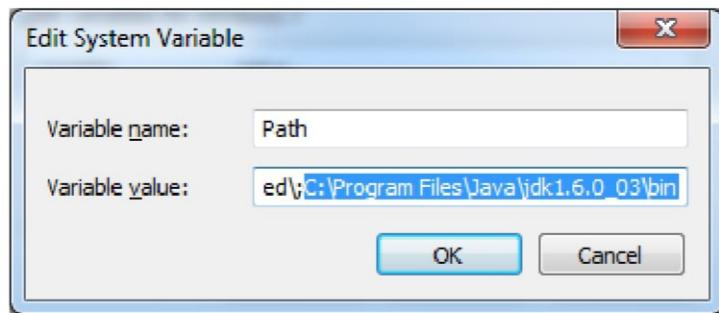
Gambar 1.2 Gambar Tab Advanced System Properties dalam Windows 7

- Klik Tombol *Environment Variables*



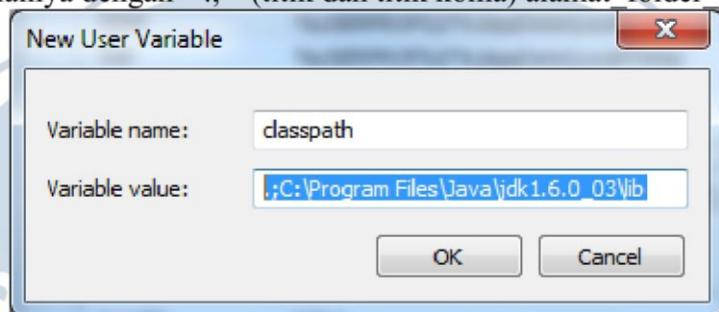
Gambar 1.3 Gambar Environment Variables dalam Windows 7

- Pada bagian *system variable*, cari *variabel path*, kemudian klik *Edit*, tambahkan alamat dari folder *bin* pada Java kompiler Anda ke dalam *value variable path*.



Gambar 1.4 Gambar Mengatur Path Pada System Variables

- Pada bagian user *variable*, tambahkan *variable classpath* kemudian isi nilainya dengan “ .; ” (titik dan titik koma) alamat *folder lib* anda



Gambar 1.5 Gambar Mengatur classpath Pada System Variables

- Cobalah Java Anda, buka *command line*
Ketikkan `java<<enter>>`
Ketikkan `javac<<enter>>`

```

Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Windows ?>java
Usage: java [-options] class [args...]
           <to execute a class>
      or   java [-options] -jar jarfile [args...]
           <to execute a jar file>

where options include:
  -client          to select the "client" VM
  -server          to select the "server" VM
  -hotspot         is a synonym for the "client" VM [deprecated]
                  The default VM is client.

  -cp <class search path of directories and zip/jar files>
  -classpath <class search path of directories and zip/jar files>
             A ; separated list of directories, JAR archives,
             and ZIP archives to search for class files.
  -D<name>=<value>
  set a system property
  -verbose[:class|gc|jnl]
  enable verbose output
  -version         print product version and exit
  -version:<value>

```

Gambar 1.6 Mencoba Hasil Pengaturan Path Java

Kalau kedua sintaks di atas tidak menemukan pesan *error*, maka instalasi berhasil

B. Konsep Dasar Pemrograman Berorientasi Objek

Java adalah salah satu bahasa pemrograman yang menggunakan paradigma pemrograman berorientasi obyek. Pemrograman berorientasi obyek menjadikan object dan class sebagai konsep utamanya. Jadi dalam pemrograman java, object dan class menjadi suatu yang paling sentral. Java adalah salah satu bahasa pemrograman yang tidak tergantung pada operating system (operating system independent), yang sekarang mulai populer dikalangan programmer.

C. Konsep Dasar Variabel

Variabel adalah satuan yang dipakai oleh program sebagai basis untuk penyimpanan data. Penggunaan variable harus terlebih dahulu dideklarasikan nama dan tipenya.

Tipe dasar (primitive) dalam java adalah sebagai berikut ;

Tabel 1.1 Tabel tipe variabel

Tipe	Keterangan
Short	short integer, bilangan bulat 16 bit
Int	integer, bilangan bulat 32 bit
Long	long integer, bilangan bulat 64 bit
Byte	bilangan bulat 8 bit
Float	bilangan real (pecahan) 32 bit
double	bilangan real (pecahan) 64 bit
Char	karakter 16 bit
boolean	boolean value true atau false

Short

Tipe variable short adalah bilangan bulat 16 bit yang mempunyai jangkauan nilai dari -2^{15} sampai dengan $2^{15}-1$ atau dari -32768 sampai dengan ±32767.

Type variable int adalah bilangan bulat 32 bit dengan jangkaun nilai dari -2^{31} sampai dengan $2^{31}-1$.

Long

Tipe variable long adalah bilangan bulat 64 bit dengan jangkauan nilai dari -2^{63} sampai dengan $2^{63}-1$.

Contoh:

```
Long l1 = 5456469;
Long l2 = 0x08ea73aaea4f25d23;
```

Byte

Type variable byte adalah bilangan bulat 8 bit yang mempunyai jangkauan niali dari 2^7 sampai dengan 2^7-1 atau dari -128 sampai dengan +127.

Contoh :

```
Byte b1 = 50;
Byte b2 = 120;
Byte b3 = 127;
```

Float

Float adalah bilangan real (pecahan, dengan format floating point yang menggunakan tempat penyimpanan 32 bit. Tipe float mempunyai jangkaun nilai dari 3.4e-038 sampai dengan 3.4e+038. Ketelitian float sampai 10 angka dibelakang koma.

Contoh :

```
Float f1 ;
Float f2 = 0.13f;
Float f3 = 0.34f;
```

Perhatikan bahwa angka float ditambah suffix huruf f yang berarti float. Hal ini diperlukan karena notasi default untuk tipe variable double.

Double

Dobel adalah bilangan real (pecahan) dengan presisi 2 kali lebih baik dari variable float.

Dobel membutuhkan 64 bit untuk penyimpanannya yang mempunyai jangkauan nilai dari 1.7e-308 sampai dengan 1.7e+308.

Char

Tipe variable char menyimpan kode Unicode (karakter internasional), karena itu membutuhkan 16 bit.

Contoh:

```
Char c;
Char c1='c';
Char c5='\t';
```

Boolean

Variabel Boolean berisi dua nilai, yaitu true dan false. Variabel Boolean dapat diperoleh dari operasi logika seperti perbandingan dua bilangan, perbandingan dua string, dan lain – lain;

Casting

Casting adalah teknik konversi suatu data tipe ke data tipe yang lain secara eksplisit. Yang menerangkan kepada compiler bahwa terjadi perubahan tipe variable, sehingga hasilnya harus dikonversi ke tipe variable tersebut.

Contoh :

```
Int I;
Double d;
Byte b;
I = 250;
D = I; // tidak terjadi kesalahan, penyesuaian otomatis
B = I; // terjadi kesalahan, tipe tidak compatible
B = (byte); //tidak terjadi kesalahan, casting dari integer ke byte
```

D. Konsep Dasar Array

Array adalah sekumpulan data yang sama tipenya. Isi array dapat diakses melalui index yang dimulai dari 0 sampai dengan jumlah_elelen_array-i.

```
Tipe_variabel nama_array[];
```

Contoh

```
Int MyArr[];
```

Pada saat didefinisikan (dideklarasikan), jumlah elemen array belum diketahui. Jumlah elemen yang sesungguhnya diciptakan dengan 2. Array MyArr dapat diakses dengan cara sbb;

```
MyArr[0] = 340;
MyArr[1] = -129;
MyArr[2] = 499;
```

Array dapat diinisialisasi secara otomatis pada saat pendeklarasian dengan cara sbb:

```
Int MyArr[] = {340,-129, 499};
```

Pada saat runtime jika indeks array yang diakses melebihi jumlah elemen yang ada maka java akan mengeluarkan pesan kesalahan berupa kesalahan runtime error.

Array Multidimesi

Kita dapat mendeklarasikan array lebih dari satu dimensi. Sebagai contoh kita akan membahas array dua dimensi. Untuk array berdimensi lebih dari dua dapat kita analogikan dengan array dimensi dua. Contoh pendeklarasian array dua dimensi:

Double Matrik[] [];

Menciptakan element array dua dimensi

Matrik = new double [2] [3];

Atau dengan cara diciptakan satu persatu (per baris)

Matrik=new double [2] []; // dimensi berikutnya belum disiapkan

Matrik[0] = new double[3];

Matrik[1] = new double[3];

Matrik[0] [0] = 9.5;

Matrik[0] [2] = 10.0;

Matrik[1] [2] = 340.9;

Atau kita dapat menginisialisasi array multi demensi pada saat pendeklarasian dengan cara :

Double Matrik [] []={

{20.5, 10.9, 309.0}

{-8.8, 98.7, 100.0}

E. Kerangka Program Java

Progam dalam java dibagi- bagi menjadi class-class. Oleh karena itu program dalam java dimulai dengan kata kunci **class** atau **public class**. Lihat Contoh Berikut :

```
public class Hello {
    public static void main(String argv [] ) {
        System.out.println("Hello java programmer UMSIDA");
    }
}
```

Progam selalu dimulai dengan pembuka blok yaitu kurung kurawal buka, dan diakhiri dengan penutup blok yaitu kurung kurawal tutup. Perlu juga diperhatikan bahwa class dalam java yang bersifat public harus disimpan dalam file sama dengan nama class dan ditambah ekstensi **java** dalam hal ini bersifat case sensitive. Jadi program di atas harus disimpan dalam file yang bernama Hello.java.

Prasyarat sebuah program aplikasi (yang bisa dieksekusi) dalam java adalah mempunyai metoda (fungsi) sebagai entry point (pintu masuk) yang bernama **main ()** untuk memulai eksekusi.

main () adalah sebuah metode dengan argument array dari String yang bernama **argv**. Klasifikasi dari metode **main ()** adalah public static yang artinya bisa diakses oleh metoda diluar class tersebut. Sedangkan kata kunci **void** menjelaskan bahwa metoda **main ()** tidak memberikan nilai balik (return value) ke pemanggil.

Perintah (statement) **System.out.println()**, merupakan perintah untuk mencetak string yang menjadi argument ke layar dan setelah pencetakan akan ganti baris. Perlu diperhatikan bahwa antara satu perintah yang lain dipisah dengan semi colon (;).

Program dalam java yang kita tulis harus kita compile dengan java compiler dengan perintah :

C:\>javac Hello.java

Javac adalah program java compiler, sedangkan Hello.java adalah program yang akan di compile. Jika proses compile berjalan lancar maka akan dibuat file sesuai dengan nama file yang di compile ditambah dengan ekstensi **class**. Jadi proses compile di atas akan menghasilkan file class yang bernama **Hello.class**. File class yang dihasilkan dari proses kompilasi dapat kita eksekusi dengan java interpreter dengan perintah :

C:\>java Hello

Setelah File Hello.class dieksekusi maka akan muncul di monitor :

```
Hello.java programmer UMSIDA
```

Komentar

Pemberian komentar dalam proram adalah penting karena akan memperjelas alur program serta akan memberikan dokumentasi yang jelas.

Pemberian komentar dalam java dapat dilakukan dengan memberikan karakter `//`. Semua teks setelah `//` akan dianggap komentar dan diabaikan dalam proses kompilasi sampai dengan garis baru (new line). Contoh :

```
System.out.println("Hello"); //mencetak kata hello
```

Selain dengan tanda `//`, penulisan komentar juga dapat menggunakan sepasang tanda `/*` dan `*/`. Jadi semua teks diantara tanda `/*` dan tanda `*/` adalah dianggap komentar dan diabaikan dalam proses kompilasi. Contoh :

```
/* perintah di bawah ini
akan mencetak teks
Hello.java programmer */
System.out.println("Hello.java programmer");
```

Kata Kunci

Program java mempunyai beberapa kata kunci yang tidak boleh kita gunakan sebagai nama variable, nama class dan yang lainnya. Kata kunci tersebut adalah :

Tabel 1.2 Tabel Kata Kunci

abstract	default	Goto	Operator	synchronized
boolean	do	If	Outer	this
Break	double	implements	Package	throw
Byte	else	Import	Private	throws
byvalue	extends	Inner	Protected	transient
Case	false	Instaceof	Public	true
Cast	final	Int	Rest	try
Catch	finally	Interface	Return	var
Char	float	Long	Short	void
Class	for	Native	Static	volatile
Const	future	New	Super	while
continue	generic	Null	switch	

LEMBAR KERJA DAN TUGAS

1. Program menggunakan type data Array dan Integer

Script	<pre>public class Array1 { public static void main (String[] args) { int[] i = new int[5]; i[0] = 200; i[1] = 1234; i[2] = 2347; i[3] = 45; i[4] = 9; System.out.println("Jumlah elemen i = " + i.length); System.out.println(" i[0] = " + i[0]); System.out.println(" i[1] = " + i[1]); System.out.println(" i[2] = " + i[2]); System.out.println(" i[3] = " + i[3]); System.out.println(" i[4] = " + i[4]); } }</pre>
--------	--

2. Program menggunakan type data Array dan String

Script	<pre>public class Array7 { public static void main (String[] args) { String[] s = {"aku", "kamu", "dia", "saya"}; System.out.println("Jumlah elemen s = " + s.length); for (int i=0; i<s.length; i++) System.out.println ("s[i] " + " panjang = " + s[i].length()); } }</pre>
--------	--

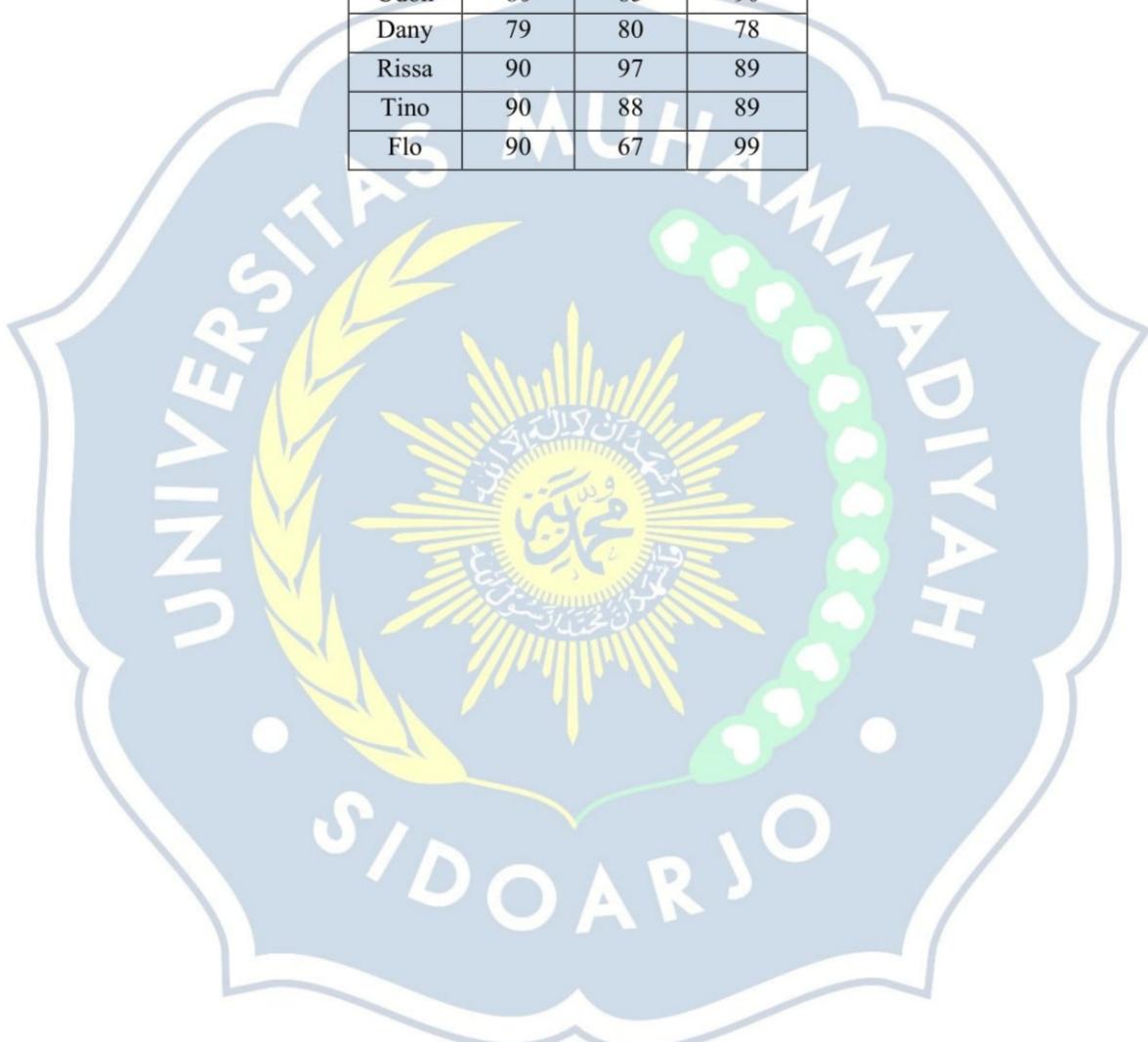
3. Program menggunakan type data integer

Script	<pre>class AutoOperator { public static void main (String[] args) { int angka = 89; int x, y, z; int hitung = 0; x = y = z = angka; System.out.println ("angka = " + angka); System.out.println ("x = " + x); System.out.println ("y = " + y); System.out.println ("z = " + z); for (hitung=0; hitung<5; hitung++) System.out.println (hitung); } }</pre>
--------	---

TUGAS

1. Buatlah program java untuk menampilkan biodata pribadi anda (min 7 kategori)!⁴
2. Buatlah sebuah program untuk menghitung persamaan berikut ini. Jika $f(x) = 2x^4 + 3x^3 + 6x^2 - 4x - 27$, tentukan berapa nilai $f(x)$ bila $x = 8$.
3. Buatlah program java dengan menggunakan array dari int yang berisi 6 element, kemudian cetak masing – masing element tersebut pada leyer monitor.
4. Dengan menggunakan array 2 dimensi, buatlah program yang menghitung nilai akhir dari 3 mahasiswa dengan aturan : $Na = 0.35 * \text{UTS} + 0.45 * \text{UAS} + 0.2 * \text{Tugas}$;

Nama	UTS	UAS	Tugas
Udon	80	85	90
Dany	79	80	78
Rissa	90	97	89
Tino	90	88	89
Flo	90	67	99



POKOK BAHASAN 2

OPERATOR , CONTROL FLOW dan STRING

PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai pendeklarasian variable, baik instance variable, class variable, maupun local variable, serta membahas fungsi dari masing-masing simbol operatot dan control flow. Sehingga setelah mempelajari bab ini diharapkan mahasiswa mampu :

- a. Mendeklarasikan variable dengan berbagai macam tipe data
- b. Mengakses instance variable, class variable, maupun local variable
- c. Memahami fungsi dari masing-masing simbol operator
- d. Memahami control flow
- e. Mahasiswa mampu membuat program yang berkaitan dengan String.
- f. Mahasiswa mampu memahami prinsip penggunaan String.

PENYAJIAN (TUTORIAL)

Operator dan Control Flow adalah jantung dari program Control Flow mempunyai kemampuan untuk mengendalikan jalan dari program. Tanpa Operator dan Control Flow, program tidak akan mampu melakukan banyak hal. Ada beberapa jenis Operator dan Control Flow dalam Java, yaitu:

1. Aritmatika Operator

Java mempunyai operator aritmatika seperti + (penjumlahan), - (pengurangan), * (perkalian), / (pembagian) dan % (modulo). Operator – operator ini berlaku seperti dalam operasi aritmatika. Dalam Java juga dimungkinkan untuk penyederhanaan penulisan operator seperti yang terlihat di bawah ini;

```
Int a, b, c, d, e,  
a = 15;b = 10; //=80;d=100;e=20;  
a+= 5;      //sama dengan a = a+5  
b-= e;      //sama dengan b = b-e  
c*= b;      //sama dengan c = c*b  
d/= 4;      //sama dengan d = d/4  
e% =3;      //sama dengan e = e % 3, hasilnya = 2  
d++;        //sama dengan d = d+1 (increment )  
c-- ;       //sama dengan c = c-1 (decrement)
```

2. Logika Operator

Ada tiga logika Operator dalam java yaitu:

&&	Logika /nd
	Logika Or
	Logika Not

Ketiga operator tersebut dioperasikan terhadap Boolean data type yang menghasilkan bolean data type juga (true atau false)

3. Bit per bit operator

Variabel bertipe long, int, short, byte dan char dapat mempunyai operator yang dioperasikan langsung kesusunan binernya (bit atau bit).

Tabel 2.1 Tabel bit per bit Operator

Operator	Nama	Keterangan
\sim	NOT	Mengubah 0 menjadi 1 dan 1 menjadi 0
$\&$	AND	$1 \& 1 = 1$ selain itu hasilnya selalu 0
$ $	OR	$0 0 = 0$ selain itu hasilnya selalu 1
\wedge	XOR	$0\wedge 0 = 0, 1\wedge 1 = 0$ selain itu hasilnya 1
$>>$	SHIFT RIGT	Dorong 1 bit ke kanan termasuk tanda minus
$>>>$	SHIFT RIGT	Dorong 1 bit ke kiri nisi bit paling kiri dengan 0
$<<$	SHIFT LEFT	Dorong 1 bit ke kiri

Seperti halnya aritmatika operator, Bit per bit operator juga bisa dilakukan Penyederhanaan penulisannya.

4. Relasional

Operator relasional digunakan untuk membandingkan dua nilai. Hasil dari perbandingan tersebut adalah data tipe Boolean (true atau false).

Tabel 2.2 Tabel operator Relasional

Operator	Keterangan
$>$	Lebih Besar
$<$	Lebih kecil
\geq	Lebih besar atau sama dengan
\leq	Lebih kecil atau sama dengan
\neq	Tidak sama dengan
\equiv	Sama dengan

Tipe variable di java seperti short, int, long, float, double, Boolean, dan character dapat dibandingkan satu dengan yang lainnya .

Contoh :

```
Char c = '*';
Int i = 0x2a;
Boolean b;
b = c == i;
```

5. Prioritas

Setiap operator mempunyai prioritas sendiri – sendiri, sehingga perhitungan matematika dilakukan sesuai prioritasnya. Lihat contoh dibawah ini :

$C = a + b * d;$

Akan dijalankan sesuai prioritas yaitu $*, +, =$

Dibawah ini diberikan table prioritas dari operator, urutan prioritas dari atas ke bawah Operator yang sama prioritasnya akan dioperasikan dari kanan ke kiri.

Tabel 2.3 Tabel operator Prioritas

Operator
$() []$
$++ -- \sim !$
$* / %$
$+ -$
$>> >>> <<$
$>>= < <=$
$= = !=$

&
^
&&
?:
=

6. Pernyataan *if*

Pernyataan *if* adalah control flow yang paling sederhana. Pernyataan ini mengevaluasi suatu ekspresi. Jika ekspresi benar (true) maka blok pernyataan akan dijalankan. Sebaliknya computer akan melompati blok pernyataan tersebut. Sintaks dari pernyataan *if*:

```
If (ekspresi)
    Pernyataan;
```

7. Pernyataan *if-else*

Pernyataan *if-else* mengevaluasi suatu ekspresi. Jika ekspresi benar (true) maka blok pernyataan dijalankan. Sebaliknya computer akan menjalankan blok pernyataan yang lain. Sintaks dari pernyataan *if-else*:

```
If(ekspresi )
    Pernyataan_1;
Else
    Pernyataan_2;
```

8. Pernyataan *switch, case dan break*

Switch digunakan untuk percabangan (seleksi) lebih dari dua. Ini digunakan untuk menghindari penggunaan *if-else* yang berlapis-lapis.

9. Pengulangan (Loop)

Pengulangan dengan *while*

Alur program blok while akan dilakukan berulang-berulang selama ekspresi benar (true)

Sintaks dari pengulangan while adalah sbb:

```
While (ekspresi)
    Pernyataan;
```

Pengulangan dengan *do-while*

Pengulangan dengan while pemeriksaan ekspresi dilakukan sebelum pengulangan. Ada kalanya pengulangan pemeriksaan ekspresi dilakukan setelah pengulangan. Jadi minimal satu kali pengulangan dilakukan. Pengulangan ini menggunakan do-while, sintaksnya.

```
do
    pernyataan;
while (ekspresi);
```

Pengulangan dengan *for*

Pengulangan dengan for menyerupai while, namun dilengkapi dengan inialisasi, test kondisi dan update counter. Jadi sintaks pengulangan dengan for adalah :

```

for (inialisasi_counter ; test_kondisi ; update_counter) {
    instruksi;
    -----
};


```

Inialisasi_counter dilakukan sekali pada saat akan dilakukan pengulangan, test_kondisi diperiksa setiap kali sebelum pengulangan dilakukan, sedangkan update_counter dilakukan setiap kali selesai pengulangan.

Break, continue

Dalam pengulangan break digunakan untuk mematahkan (mengakhiri) pengulangan secara paksa. Sedangkan continue digunakan untuk melanjutkan pengulangan berikutnya secara paksa dengan mengabaikan perintah dibawah pernyataan continue.

String

String adalah obyek yang terdiri atas deretan karakter. Tidak seperti array dari karakter, String tidak perlu ditentukan jumlah panjang karakternya. String dapat langsung diinisiasi tanpa kata kunci new, contoh :

```

String str = "Hello";
Str = str + " Java Programmer";

```

String dapat juga diciptakan melalui cara berikut :

```

char c_arr [ ] = { 'h', 'e', 'l', 'o' };
String s1 = new String (c_arr);
String s2 = new String (c_arr, 2, 2);
// offset ke 2 sebanyak 2 karakter

```

Ada beberapa method dalam String yang perlu kita ketahui antara lain:

Int length ()

memberikan nilai balik (return value) panjang dari aktual string.

char charAt (int index)

memberikan nilai balik berupa karakter pada posisi index dalam aktual string.

public class TestString {

```

    public static void main(String argv[]){
        String tabel[]={"amir","bambang","hasan"};
        String str="Ini adalah # String yang diproses";
        boolean found=false;
        int I;
        for (I=0;I<str.length();I++){
            if (str.charAt(I)=='#'){
                found=true;
                break;
            }
        }
        if (found)
            System.out.println("Posisi =" +I);
    }
}

```

int indexOf(char ch)

Memberikan nilai balik posisi pada String di mana karakter ch berada

int indexOf(String s)

Memberikan nilai balik posisi index pada String dimana string s ditemukan

int indexOf(char ch, int starPos)

int indexOf(String s, int startPos)

Sama dengan indexOf sebelumnya, hanya pencarian tidak dimulai dari index 0, melainkan dari index = startPos

Int lastIndexOf(char c)

```
Int lastIndexOf(char c, int startPos)
```

```
Int lastIndexOf(String s);
```

```
Int lastIndexOf(string s, int startPos)
```

Method `lastIndexOf()` mencari karakter/String pada String yang diberikan, hanya pencarian dimulai dari kanan ke kiri.

```
Boolean equals (String s)
```

Membandingkan aktual string dengan string s dengan mengabaikan huruf besar maupun huruf kecil. Jika sama maka akan memberikan nilai balik true, sebaliknya akan memberikan nilai balik false.

```
String substring (int offset)
```

Memberikan nilai balik isi aktual string dari posisi offset sampai akhir string

```
String substring(int offset, int endIndex)
```

Memberikan nilai balik si aktual string dari posisi offset sampai posisi endIndex

```
String toLowerCase()
```

Mengubah string menjadi huruf kecil semua

```
String toUpperCase()
```

Mengubah spasi yang berdekatan dalam string

```
Char[ ] getChars(int startPos, int endPos, char buffer[ ], int offset)
```

Mengambil karakter dan startPos sampai endPos, dan menyimpannya ke dalam buffer dimulai dari index = offset.

```
Int compareTo(String str)
```

Membandingkan String dalam UTF (Unicode Transfer Format) dengan nilai balik 0 bila sam, negatif bila kecil, dan positif bila lebih besar.

StringTokenizer

Class StringTokenizer berfungsi untuk membagi string berdasarkan token tertentu. Class ini sangat membantu untuk menginterpretasikan suatu string ke suatu maksud tertentu. Contoh yang paling populer adalah membagi kalimat menjadi beberapa kata. Kita juga membaca data dari file kemudian data tersebut kita uraikan berdasarkan token tertentu.

LEMBAR KERJA DAN TUGAS

1. Menggunakan bit operator

Script	<pre>public class OprTest { public static void main (String [] args){ byte b=0x8; byte c, d, e, f, g, h, i; c=(byte) ~b; c &=0xf; d= (byte) (b c); e= (byte) (c>>2); f= (byte) (e + 3); g= (byte) (c & f); h= c; h ^=0x3; i= (byte) (h >>>1); i <<=2; System.out.println(" c = " + c); System.out.println(" d = " + d); System.out.println(" e = " + e); System.out.println(" f = " + f); System.out.println(" g = " + g); System.out.println(" h = " + h); System.out.println(" I = " + i); } }</pre>
--------	---

	}
--	---

2. Menggunakan pernyataan if

Script	<pre>public class TestIf { public static void main (String argv []) { int x = -3; if (x < 0) System.out.println("Negatif"); } }</pre>
--------	--

3. Menggunakan pernyataan if-else

Script	<pre>public class TestIfElse { public static void main (String argv []){ int a=10, b=13, x=0, y=0; if (a>b) x=a; else{ x=b; y=100; } System.out.println("a="+a+" "+b); x+="x+" y+="y"; } }</pre>
--------	---

4. Menggunakan pernyataan switch case, dan break.

Script	<pre>public class TestSwitch { public static void main (String argv []) { char c=' '; switch (c) { case '+': System.out.println("Operasi penjumlahan"); break; case '-': System.out.println("Operasi pengurangan"); break; case '/': System.out.println("Operasi pembagian"); break; case '*': System.out.println("Operasi perkalian"); break; default: System.out.println("Operasi tidak dikenal"); } } }</pre>
--------	--

5. Menggunakan Pengulangan dengan while

Script	<pre>public class TestWhile { public static void main (String argv []) { int m ; m = 5 ; while (m > 0) { System.out.println ("Nilai m sama dengan = "+m); m--; } } }</pre>
--------	---

6. Menggunakan Pengulangan do-while

Script	<pre>public class TestDoWhile { public static void main (String argv []) { int m = 5; do { System.out.println(" m =" +m); m--; }while (m > 0); } }</pre>
--------	---

7. Menggunakan pengulangan dengan for

Script	<pre>public class TestFor { public static void main (String argv []) { int m ; for (m = 5 ; m > 0 ; m--) { System.out.println(" nilai m adalah "+m); } } }</pre>
--------	--

8. Menggunakan break

Script	<pre>public class TestBreak { public static void main (String argv[]) { int i, j ; j = 50 ; for (i=5; i>0; i--) { if (j>100) break; j += 50; System.out.println(" i = " +i); } System.out.println("Program berhenti i=" +i + j+""); } }</pre>
--------	---

9. Menggunakan continue

Script	<pre>public class TestContinue { public static void main (String argv []) { int i, j ; j = 50 ; for (i=5; i>0; i--) { if (j>100) continue; j += 50; System.out.println(" i = " +i); } System.out.println("Program berhenti i=" +i + j+"+j); } }</pre>
--------	---

10. Menggunakan Boolean equals (string s)

Script	<pre>import java.io.*; public class Compare { public static void main(String argv[])throws IOException{ String tabel[]={ "ari", "gali", "ika", "rozi"}; DataInputStream dis=new DataInputStream(System.in); System.out.print("Berikan nama :"); System.out.flush(); String nama=dis.readLine(); for (int i=0;i<tabel.length;i++){ if (tabel[i].equals(nama)){ System.out.println(nama+" terdaftar dalam tabel"); break; } } }</pre>
--------	--

11. Menggunakan Int CompareTo (string str)

Script	<pre>public class StringOperasi { public static void main(String argv[]){ String salam="Terima Kasih, Anda Belajar Java Programming"; String bagian=salam.substring(5,16); String satu="Satu", dua="Dua", tiga="Tiga"; System.out.println(bagian); System.out.println(salam.toUpperCase()); System.out.println(salam.toLowerCase()); System.out.println(satu.compareTo("Satu")); System.out.println(satu.compareTo(dua)); System.out.println(dua.compareTo(tiga)); } }</pre>
--------	--

12. Membagi String berdasarkan token tertentu

Script	<pre>import java.util.*; public class TestToken { public static void main(String argv[]){ String kalimat="Di sini Pencerahan Bersemini";</pre>
--------	--

	<pre> StringTokenizer st = new StringTokenizer(kalimat, " "); while(st.hasMoreTokens()) { System.out.println(st.nextToken()); } } </pre>
--	--

13. Membaca data dari file kemudian menguraikan data tersebut berdasarkan token tertentu

- Membuat file customer.txt dengan struktur sbb

Nama_depan : nama_belakang : kota : propinsi

- Dari file customer.txt tersebut bisa diuraikan data-data customer dengan menggunakan class dibawah ini

Script	<pre> import java.io.*; import java.util.*; public class TokenFile { public static void main(String args[]){ TokenFile tt=new TokenFile(); tt.dbTest(); } public void dbTest(){ DataInputStream dis=null; String dbRecord=null; try { File f=new File("customer.txt"); FileInputStream fis=new FileInputStream(f); BufferedInputStream bis=new BufferedInputStream(fis); dis=new DataInputStream(fis); //read db while((dbRecord=dis.readLine())!=null){ StringTokenizer st=new StringTokenizer(dbRecord, ":"); String namaDepan =st.nextToken(); String namaBelakang=st.nextToken(); String kota=st.nextToken(); String propinsi=st.nextToken(); System.out.println("Nama Depan :" +namaDepan); System.out.println("Nama :" +namaBelakang); System.out.println("Kota :" +kota); System.out.println("Propinsi :" +propinsi+"\n"); } } catch (IOException e){ // catch error System.out.println("IOException error!" +e.getMessage()); }finally{ if(dis!=null){ try{ dis.close(); }catch (IOException ioe){ </pre>
--------	---

	}
	}
	}

Tugas

1. Buatlah program untuk menghitung jumlah bilangan 1 s/d 100 ($1+2+3+4+\dots+100$). Tampilkan hasilnya pada layar monitor.
2. Buatlah program untuk menampilkan semua bilangan dari 1 s/d 100 yang habis dibagi 5 atau habis dibagi 9.
3. Buatlah kode program dengan menggunakan statement if, yang menampilkan nilai yang dimasukkan dan memberikan keterangan terhadap nilai tersebut. dengan ketentuan sebagai berikut : nilai >80 mendapat nilai = A, keterangan : Lulus - Memuaskan nilai $>70 \leq 80$ mendapat nilai = B. keterangan Lulus - Baik nilai $>60 \leq 70$ mendapat nilai = C. keterangan Lulus - Cukup nilai $>50 \leq 60$ mendapat nilai = D. keterangan Tidak Lulus – Tidak Memuaskan
4. Buatlah program untuk menampilkan susunan bilangan biner dari suatu bilangan bulat, dengan susunan terbalik misalkan biner dari 12 akan ditulis 0011.
5. Buatlah class untuk membalik urutan kata dalam suatu kalimat.
6. Input sebuah nim, dan ganti semua digit ‘1’ menjadi ‘*’.

Contoh Layout :

Input NIM Anda : 0111500382
 NIM Anda menjadi : 0***500382
 Input NIM Anda : 0244500016
 NIM Anda menjadi : 02445000*6

POKOK BAHASAN 3

CLASS

PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai objek dan class, serta penggunaannya dalam sebuah program java sederhana. Setelah mempelajari materi ini diharapkan mahasiswa mampu untuk :

- Mengetahui pengertian dari objek & class
- Dapat membuat program sederhana dari java dengan menggunakan objek dan class.

PENYAJIAN (TUTORIAL)

Class merupakan bagian pokok dari Pemrograman Berorientasi Obyek (Object Oriented Programming). Class merupakan cara merakit data-data dan menentukan semua metode yang diperlukan untuk mengakses, menggunakan dan mengubah data-data tersebut. Dalam java class juga bisa dianggap sebagai type dari object.

Setiap class memiliki dua hal utama. Yang pertama adalah yang kita sebut state/atribut/property. State dalam obyek tidak lain adalah nilai dari setiap variable. Bagian yang kedua adalah tool atau method yang menentukan utility yang dimiliki oleh class.

1. Pendeklarasian class

Sintaks pendeklarasian class dalam java adalah

```
modifiers class NamaClass {
    <class attribute list>
    <class method list>
}
```

modifier menentukan bagaimana class dapat ditangani pada fase development berikutnya. Ada tiga jenis modifier dalam class yaitu *public*, *final*, *abstract* dan *friendly*.

- *Public*
Dengan menempatkan public didepan pendeklarasian class, mempunyai arti bahwa class didefinisikan untuk public. Ini berarti class tersebut dapat dipakai atau dikembangkan (extend) oleh sembarang object.
- *Final*
Dengan menempatkan final didepan pendeklarasian class menandakan bahwa class tersebut tidak bisa memiliki subclass (turunan). Dengan kata lain class tersebut tidak bisa dikembangkan.
- *Abstract*
Dengan menempatkan class didepan pendeklarasian class, menandakan bahwa kita tidak bisa menciptakan obyek dari class tersebut. Class tersebut harus dipakai sebagai super class dari class lain. Dalam abstract class paling tidak ada satu method yang belum komplit.
- *Friendly*
Secara default modifiers class adalah friendly ,artinya class tersebut tersebut boleh dikembangkan atau dipakei oleh class lain dalam satu paket (package) . jika kita tidak memberikan modifiers didepan pendefinisian class maka class bersifat friendly.

<class attribute list> adalah daftar semua attribute (variable) dari class, sedangkan *<class method list>* adalah daftar semua method (fungsi\operasi) yang bisa dilakukan oleh class tersebut.

Sintaks pendefinisan class attribute :

modifiers data_type nama attribute ;

Sintaks penderfinsian class method :

modifiers return_type nama method (<parameters list>)

Akses modifiers dalam class attribute maupun class method menentukan bagaimana aksesibilitas attribute dan method dari class lain. Ada beberapa akses modifier yaitu:

- *Friendly*

Secara default akses modifiers adalah friendly, yang berarti bahwa attribute\method bisa diakses oleh class lain dalam satu paket (package).

- *Public*

Akses modifiers public membuat attribute\method tersebut dapat diakses oleh pihak luar manapun.

- *Private*

Akses modifiers private membuat attribute\method hanya dapat diakses oleh class itu sendiri

- *Protected*

Akses modifiers protected membuat attribute\method hanya dapat diakses oleh class itu sendiri dan class turunannya (subclass nya).

- *Final*

akses modifiers final membuat attribute tersebut tidak boleh diubah pada saat eksekusi. sebab attribute ini tidak boleh diubah maka harus diberi nilai pada saat pedeklarasian misal *final int SIZE = 5;*

- *Static*

Akses modifiers static membuat attribute tersebut mempunyai nilai sama untuk semua obyek dari class tersebut.

2. Konstruktur

Konstruktur adalah method dengan property yang unik dan untuk keperluan inialisasi pada saat obyek diciptakan. Jadi konstruktur digunakan memberikan nilai kepada attribute tertentu pada saat obyek diciptakan. Konstruktor didefinisikan dengan nama yang sama dengan nama class. Konstruktor tidak perlu *return_type* sebab konstruktur secara aktual dipanggil sebagai method. Konstruktur ini dipanggil pada saat kita memberikan perintah *new*.

3. Inheritance (Pewarisan)

Inheritance (Pewarisan) adalah suatu mekanisme yang memungkinkan satu class mewarisi (memiliki) attribute dan method dari class lain. Mekanisme pewarisan ini diperlukan untuk mengembangkan class-class yang sudah ada tanpa harus membuka source code dari class-class tersebut. Class yang mewarisi disebut subclass sedangkan class yang diwarisi disebut super class. Pewarisan class dalam java menggunakan sintaks:

*modifier<NamaSubClass> extends<NamaSuperCalss> {
}*

4. Kata Kunci *this*

this adalah referensi yang menunjuk pada obyek aktual. Nilai *this* ini menunjukkan bahwa obyek yang dimaksud adalah obyek yang sedang aktif itu sendiri. Contoh penggunaan dari *this* adalah sbb:

5. Overriding Method

Overriding method adalah suatu mekanisme pewarisan class di mana subclass mendefinisikan ulang satu atau lebih method (fungsi) yang ada di supper class. Dalam overriding ini subclass bisa mendefinisikan method (fungsi) yang di-override dengan fungsi yang baru sama sekali, atau menambahi method (fungsi) yang di-override yang sudah ada dengan fungsinal-fungsinal yang lain. Contoh class yang menggunakan mekanisme overriding :

LEMBAR KERJA DAN TUGAS

1. Mendefinisikan Class

Script	<pre>public class Kotak { private int Panjang, Lebar, Tinggi; public void setPanjang(int pjg){ Panjang=pjg; } public void setLebar(int lbr){ Lebar=lbr; } public void setTinggi(int tg){ Tinggi=tg; } public int getPanjang(){ return Panjang; } public int getLebar(){ return Lebar; } public int getTinggi(){ return Tinggi; } public int luas(){ int hasil; hasil=2*Panjang*Lebar+2*Panjang*Tinggi+2*Lebar*Tinggi; return hasil; } public int volume(){ int hasil; hasil=Panjang*Lebar*Tinggi; return hasil; } }</pre>
--------	---

Script	<pre>public class TestKotak { public static void main (String argv []){ Kotak k1; int vol,luas,pjg; k1=new Kotak (); k1.setPanjang (10); k1.setLebar (15); k1.setTinggi (4); vol=k1.volume(); luas=k1.luas(); pjg=k1.getPanjang(); System.out.println("Kotak k1 luas="+luas+ " volume="+vol+" Panjang="+pjg); } }</pre>
--------	--

	}
--	---

2. Menggunakan konstruktur

Script	<pre>public class Kotak2 { private int panjang, lebar, tinggi; public Kotak2(){ panjang = 0; lebar = 0; tinggi = 0; } public Kotak2(int pjg,int lbr,int tg){ panjang = pjg; lebar = lbr; tinggi = tg; } public void setpanjang(int pjg){ panjang=pjg; } public void setLebar(int lbr) { lebar=lbr; } public void setTinggi(int tg) { tinggi=tg; } public int getPanjang() { return panjang; } public int getLebar() { return lebar; } public int getTinggi() { return tinggi; } public int luas(){ int hasil; hasil=2*panjang*lebar+2*panjang*tinggi+2*lebar*tinggi; return hasil; } public int volume(){ int hasil; hasil=panjang*lebar*tinggi; return hasil; } }</pre>
--------	--

Script	<pre>public class TestKotak2 { public static void main (String argv[]){ Kotak2 k1, k2; int vol1, luas1, pjg1, vol2, luas2, pjg2; k1=new Kotak2 (10,15,4); k2=new Kotak2 (); vol1=k1.volume(); luas1=k1.luas(); pjg1=k1.getPanjang(); System.out.println("Kotak2 k1 Luas = "+luas1+ " volume =" +vol1+ " Panjang = "+pjg1); vol2=k2.volume(); luas2=k2.luas();</pre>
--------	---

	<pre> pjg2=k2.getPanjang(); System.out.println("Kotak2 k2 Luas = "+luas2+ " volume =" +vol2+ " Panjang = "+pjg2); } } </pre>
--	--

3. Menggunakan Inheritance (pewarisan)

Script	<pre> public class KotakHitam extends Kotak2 { private String rekaman; public KotakHitam(){ super(); rekaman=""; } public KotakHitam(int pjg,int lbr,int tg,String rkm){ super(pjg,lbr,tg); rekaman=rkm; } public void rekam(String teks){ rekaman=rekaman+" "+teks; } public void putarRekaman(){ System.out.println(rekaman); } } </pre>
--------	--

Script	<pre> public class TestKotakHitam { public static void main (String argv[]){ KotakHitam k1, k2; int vol1, luas1, pjg1, vol2, luas2, pjg2; k1=new KotakHitam (10,15,4,"Halo Java "); k2=new KotakHitam (); vol1=k1.volume(); luas1=k1.luas(); pjg1=k1.getPanjang(); System.out.println("Kotak k1 Luas = "+luas1+ " Volume =" +vol1+ " Panjang = "+pjg1); vol2=k2.volume(); luas2=k2.luas(); pjg2=k2.getPanjang(); System.out.println("Kotak k2 Luas = "+luas2+ " Volume =" +vol2+ " Panjang = "+pjg2); k1.rekam("Ada Suara Pesawat Terbang"); k2.rekam("Ada Tanda Sirene dari arah Utara"); k1.putarRekaman(); k2.putarRekaman(); } } </pre>
--------	---

4. Menggunakan kata kunci this

Script	<pre> public class Point { private int x, y; public Point (int x, int y) { this.x = x ; this.y = y ; } } </pre>
--------	---

```

        }
        public Point () {
            this (10,10);
        }
    }
}

```

5. Menggunakan Overriding Method

Script	<pre> public class P { int a=5; public void cetak(){ System.out.println("super class a="+a); } } </pre>
--------	---

Script	<pre> public class pDemol extends P{ int b=10; public void cetak (){ System.out.println("Class aktual: b="+b); } public static void main (String[] args) { pDemol p=new pDemol(); p.cetak(); } } </pre>
--------	---

Script	<pre> public class pDemo2 extends P{ int b = 10; public void cetak(){ System.out.println("Class Actual : b = "+ b); super.cetak(); //memanggil cetak di super } public static void main (String argv[]){ pDemo2 P = new pDemo2(); P.cetak(); } } </pre>
--------	--

Tugas

1. Buatlah class Titik yang merepresentasikan titik koordinat dua dimensi x dan y
2. Buatlah class Titik3D turunan dari class Titik yang merepresentasikan titik koordinat tiga dimensi x, y dan z
3. Buat Class Konversi yang anggotanya :
 - Atribut:
 - jarak (dalam meter)
 - Method :
 - meterKekilo();
 - kiloKemeter();

POKOK BAHASAN 4

THREADS

PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai threads. Setelah mempelajari bab ini diharapkan mahasiswa mampu :

- a. Mahasiswa dapat membuat threads dengan menggunakan pemrograman java
- b. Mahasiswa dapat menjelaskan kegunaan threads

PENYAJIAN (TUTORIAL)

Property unik dari java adalah bahwa Java memberi support adanya threads. Threads memperbolehkan kita melakukan banyak hal pada waktu yang bersamaan. Threads adalah komponen java yang memungkinkan kita untuk membuat program yang bersifat multitasking programming. Sebenarnya setiap program dalam java selalu menggunakan paling tidak satu thread

Kita bisa membuat aplikasi dan class berjalan dalam threads yang terpisah dengan dua cara, yaitu :

- Mengembangkan dari class Thread (extending class Thread)
- Mengimplementasikan Runnable interface

Mengembangkan class Thread

Dengan menggunakan extends Thread, mekanisme Thread dapat diaplikasikan dengan membuat fungsi run()

Untuk melihat concurrency (multitasking) pada program tersebut tambahkan method sleep() untuk actual Thread (current Thread). Method sleep() hanya dapat dilakukan dengan menangkap sinyal

Method yang berhubungan dengan Thread

start()

Mengaktifkan Thread Obyek Thread kemudian akan menjalankan method run().

stop()

Menghentikan Thread yang sedang aktif.

sleep(int delta)

Menunda aktifitas Thread untuk delta millisecond.

Contoh selanjutnya adalah mendemonstrasikan animasi yang dilakukan dengan menjalankan Thread. Tugas Thread adalah melakukan repaint() pada obyek. Setelah repaint() dilakukan, Thread beristirahat untuk beberapa detik dan untuk kemudian melakukan repaint() kembali. Obyek Thread kemudian akan menjalankan method run().

Kemudian class TThread digunakan oleh class yang merupakan pengembangan (extend) dari class Applet. Class Applet digunakan untuk class-class yang digunakan dalam aplikasi applet (untuk ditampilkan dalam html file)

Setelah kita compile kedua class tersebut, yaitu class TThread dan class Animasi, kita dapat membungkusnya ke dalam file html. Selanjutnya kita dapat mengeksplor file tersebut dengan internet browser.

Mengimplementasikan Runnable interface

Pada contoh di atas Thread digunakan dengan mengembangkan class Thread, yang mana berarti bahwa Subclass Thread hanya dapat mewarisi sifat-sifat dari Thread. Pengikatan Class ini dapat dihindari dengan menggunakan konsep interface. Dengan

mengimplementasikan Runnable interface. Interface Runnable ini hanya berisi satu method yang harus diimplementasikan, yaitu run().

Konstruktor Thread adalah :

Thread(Runnable ThreadObject)

Atau

Thread(Runnable ThreadObject, String nama Thread)

ThreadObject pada contoh di atas adalah **this**, yang merupakan Obyek itu sendiri yang mengimplementasikan Runnable.

Method is Active() akan memberikan nilai true jika Thread actual aktif, dan false jika tidak aktif.

LEMBAR KERJA DAN TUGAS

1. Mengaplikasikan mekanisme Thread dengan menggunakan fungsi run ()

Script	<pre>public class SimpleThread extends Thread{ public SimpleThread(String nama) { super(nama); } public void run(){ for (int i=0 ;i<10; i++){ System.out.println(i+" "+getName()); } System.out.println("SELESAI "+getName()); } }</pre>
Script	<pre>public class TwoThreadTest { public static void main (String argv []){ SimpleThread t1,t2; t1 = new SimpleThread ("Surabaya"); t2 = new SimpleThread ("Jakarta"); t1.start (); t2.start (); } }</pre>

2. Melihat concurrency (multitasking) pada program dengan menambahkan method sleep() untuk actual Thread (current Thread). Method sleep() hanya dapat dilakukan dengan menangkap sinyal interrupt dengan mekanisme try & catch.

Script	<pre>public class SimpleThread1 extends Thread { public SimpleThread1(String nama){ super(nama); } public void run(){ for (int i=0;i<10;i++){ System.out.println(i+" "+getName()); try{ sleep((int)Math.random()*10000); }catch(Exception e){} } System.out.println("SELESAI "+getName()); } }</pre>
--------	---

Script	<pre>public class TwoThreadTest1 { public static void main(String argv[]){ SimpleThread t1,t2; t1=new SimpleThread("Surabaya"); t2=new SimpleThread("Jakarta"); t1.start(); t2.start(); } }</pre>
--------	---

3. Mendemonstrasikan animasi yang dilakukan dengan menjalankan Thread.
(Dalam animasi ini diperlukan file image T1.gif s/d T17.gif.)

Script	<pre>import java.awt.*; public class TThread extends Thread{ Component oby; int delay; public TThread(Component oby, int delay){ this.oby=oby; this.delay=delay; } public void run(){ while(true){ try{ oby.repaint(); sleep(delay); } catch (Exception e) {} } } }</pre>
--------	---

Script	<pre>import java.applet.*; import java.awt.*; public class Animasi extends Applet{ int count, lastcount; Image gambar[]; TThread timer; Button btnStart; Button btnStop; public void init(){ gambar=new Image[17]; lastcount=17; count=0; MediaTracker mt=new MediaTracker(this); for (int i=0; i<lastcount; i++){ gambar[i]=getImage(getCodeBase(),"T"+(i+1)+".gif"); mt.addImage(gambar[i],0); } mt.checkAll(true); btnStop=new Button("Stop"); add(btnStop); btnStart=new Button("Start"); add(btnStart); show(); } public void start(){ } }</pre>
--------	--

```

        if(timer==null){
            timer=new TThread(this,75);
            timer.start();
            showStatus("Start...");
        }
    }
    public void stop(){
        if (timer!=null){
            showStatus("Stop...");
            timer.stop();
            timer=null;
        }
    }
}

```

Setelah mengcompile kedua class tersebut, yaitu class TThread dan class Animasi, lalu membungkusnya ke dalam file animasi.html. Selanjutnya mengeksplor file tersebut dengan internet browser.

```

<html>
<head> Contoh Animasi </head>
<title> Contoh Animasi </title>
<applet code = Animasi.class width=300 height=200>
</applet>
</html>

```

4. Mengimplementasikan Runnable Interface

Script	<pre> import java.applet.*; import java.awt.*; import java.lang.*; public class Animasi implements Runnable{ int count, lastcount; Image gambar[]; Button btnStart; Button btnStop; Thread timer; public void init(){ gambar=new Image[17]; lastcount=17; count=0; MediaTracker mt=new MediaTracker(this); for (int i=0; i<lastcount; i++){ gambar[i]=getImage(getCodeBase(),"T"+(i+1)+".gif"); mt.addImage(gambar[i],0); } mt.checkAll(true); btnStop=new Button("Stop"); add(btnStop); btnStart=new Button("Start"); add(btnStart); show(); } public void start(){ if(timer==null){ timer=new Thread(this); timer.start(); showStatus("Start.."); } } } </pre>
--------	--

```
        }
        public void stop(){
            if (timer!=null){
                showStatus("Stop..");
                timer.stop();
                timer=null;
            }
        }
        public boolean action(Event e, Object o){
            if(e.target==btnStop){
                stop();
                return true;
            }
            if(e.target==btnStart){
                start();
                return true;
            }
            return false;
        }
        public void paint(Graphics g){
            g.drawImage (gambar[count++],70,70,null);
            if(count==lastcount)
                count=0;
            showStatus ("Gambar no"+(count+1));
            setLayout (new
            FlowLayout (FlowLayout.CENTER,5,5));
            setBackground (java.awt.Color.lightGray);
            //setSize (0,0);
        }
        public void run(){
            while (isActive()){
                try{
                    repaint();
                    Thread.sleep(100);
                }
                catch(Exception e){
                    showStatus ("Exception :" +e);
                }
            }
            showStatus ("Not Active");
            timer=null;
        }
    }
```

Tugas

1. Buatlah class dalam java yang mengaktifkan tiga thread sekaligus. Tunjukkan bahwa ketiga thread tersebut bekerja bersama-sama dengan mencetak output ke layar monitor.

POKOK BAHASAN 5

EXCEPTION (EKSEPSI)

PENDAHULUAN

Setelah mempelajari bab ini diharapkan mahasiswa mampu :

- Memahami perbedaan antara kode program java yang memanfaatkan Exception Handling dan yang tidak memanfaatkan.
- Mahasiswa mampu mengimplementasikan kata kunci try, catch, dan throw dan throws.

PENYAJIAN (TUTORIAL)

Eksepsi adalah mekanisme yang memungkinkan program untuk menangkap kondisi-kondisi tertentu yang membutuhkan penanganan khusus. Eksepsi dapat berupa kesalahan yang terjadi dalam program pada saat run time. Format eksepsi adalah :

```
Try {
    // blok program
    . . .
} catch (EksepsiTipe1 e) {
    // blok program untuk EksepsiTipe1
    . . .
} catch (EksepsiTipe2 e) {
    // blok program untuk EksepsiTipe2
    . . .
} catch (EksepsiTipe3 e) {
    // blok program untuk EksepsiTipe3
    . . .
} finally {
    // blok program
    . . .
}
```

try menyatakan bahwa dalam blok try dapat terjadi suatu ekspresi dan bila ekspresi terjadi, maka aktifkan program yang berada dalam kelompok (blok) **catch** sesuai dengan tipe ekspresinya.

Object **e** dari tipe eksepsi dapat digunakan sebagai referensi penanganan eksepsi aktual. **finally** menyatakan bahwa apapun jenis tipe ekspresinya, maka program dalam blok ini harus tetap dijalankan.

Dengan menangkap eksepsi program tidak langsung berhenti dan melaporkan kesalahan tersebut, melainkan membelokkan dengan program pengaman tertentu.

Pada saat run time, jika java menemukan kesalahan dan melempar pesan ke program. Dengan pesan ini, program akan melakukan suatu aksi tertentu. Mekanisme melempar ini disebut juga sebagai **throw**, alur program akan dihentikan, kemudian program pengendali (exception handling) akan diaktifkan.

throw

Throw digunakan secara eksplisit untuk membuang sebuah eksepsi. Contoh berikutnya, TestThrow mendefinisikan Obyek Eksepsi “NullPointerException” pada sebuah variabel t. Dengan instruksi throw, program Eksepsi untuk kemudian ditangkap oleh instruksi catch. Namun tentunya yang ditangkap harus sesuai dengan tipe eksepsi yang dilemparkan. Dalam hal ini program

throws

Eksepsi selalu ditangkap dengan **catch** pada blok dimana eksepsi tersebut dilempar. Tanpa **catch**, program tidak akan dikompilasi. Program dibawah ini menghilangkan **try** dan **catch** pada subroutine `demo()`, namun dengan menggunakan tambahan kata kunci **throws Exception**.

Tabel 6.1 Common Jaxa Exceptions

Exceptions	Description
Arithmeticexception	Disebabkan oleh error secara matematika seperti pembagian dengan nol
ArrayIndexOutOfBoundsException	Disebabkan oleh array index yang salah
ArrayStoreException	Disebabkan ketika program mencoba menyimpan file data yang salah
FileNotFoundException	Disebabkan oleh usaha mengakses file yang tidak ada
IOException	Disebabkan oleh general I/O failures, seperti kegagalan membaca file
NullPointerException	Disebabkan oleh mereferensi obyek null
NumberFormatException	Disebabkan ketika konversi antara string dan bilangan gagal
OutOfMemoryException	Disebabkan ketika tidak cukup memory untuk alokasi obyek
SecurityException	Disebabkan ketika sebuah applet mencoba melakukan aksi yang tidak diperbolehkan oleh setting security browser
StackOverflowException	Disebabkan ketika sistem run melebihi kapasitas stack
StringIndexOutOfBoundsException	Disebabkan ketika program mencoba mengakses posisi karakter yang tidak ada di dalam string

LEMBAR KERJA DAN TUGAS

1. Menggunakan exception

```
public class Arg0 {
    public static void main (String argv[]) {
        String str;
        str = argv[0];
        System.out.println("Argument="+str);
    }
}
```

Bila program di atas dijalankan :

C:>javac Arg0.java

C:>java Arg0 hello

Argument = hello

Namun bila argument tidak disertakan :

C:>java Arg0

java.lang.ArrayIndexOutOfBoundsException:0 at Arg0.main(Arg0.java:5)

untuk mencegah hal itu, maka program perlu dilengkapi dengan penanganan eksepsi

Script	<pre> public class Arg01 { public static void main (String argv[]) { String str="??????"; try{ str = argv[0]; }catch(ArrayIndexOutOfBoundsException e) { System.out.println("Argument Kosong"); } System.out.println("Argument="+str); } } </pre>
--------	--

2. Menggunakan Throw

Script	<pre> public class CobaThrow { static void demo(){ try{ NullPointerException t; t = new NullPointerException("Test Throw"); throw t; //Baris Ini tak pernah dijalankan }catch (ArithmetricException e){ System.out.println("Auuuu"); } } public static void main (String argv[]) { try{ demo(); System.out.println("Selesai"); }catch (NullPointerException e){ System.out.println("Saya tangkap sinyal NullPointerException :" + e); } } } </pre>
--------	--

3. Menggunakan Throws

Script	<pre> public class ThrowDemo { static void demo()throws IllegalAccessException{ System.out.println("xxxxxx"); throw new IllegalAccessException ("ThrowDemo"); } public static void main (String argv[]) { try{ demo(); System.out.println("Selesai"); }catch (IllegalAccessException e){ System.out.println("Saya tangkap " + e); } } } </pre>
--------	--

Tugas

1. Buatlah program yang hanya bisa menerima input berupa angka saja. Bila diinputkan selain angka (numeric) maka akan ditangkap oleh blok catch
2. Berdasarkan program diatas tambahkan blok finally untuk menyatakan bahwa telah dilakukan pengecekan terhadap data yang diinputkan



POKOK BAHASAN 6

JAVA GRAPHICAL USER INTERFACE

Tujuan

1. Mahasiswa memahami dan menguasai pembuatan java GUI
2. Mahasiswa memahami konsep dasar pembuatan java GUI menggunakan Swing
3. Mahasiswa memahami Komponen dasar user interface dengan swing

PENYAJIAN(TUTORIAL)

Graphical User Interface

Graphical User Interface merupakan antar muka grafis yang memfasilitasi interaksi antara pengguna dengan program aplikasi. Salah satu komponen dalam bahasa pemrograman Java untuk membangun GUI adalah Swing. Komponen ini didefinisikan di dalam paket javax.swing. Swing diturunkan dari *Abstract Windowing Toolkit* dalam paket java.awt. Hierarki dari komponen Swing adalah :

`java.swing.Jcomponent → java.awt.Container → java.awt.Component → Java.Lang.Object`

JComponent adalah *superclass* dari semua komponen Swing. Sebagian besar fungsionalitas komponen diturunkan dari *superclass* ini. Beberapa komponen utama dalam GUI adalah:

1. Containers: merupakan wadah yang berfungsi untuk menempatkan komponen-komponen lain di dalamnya.
2. Canvas.: merupakan komponen GUI yang berfungsi untuk menampilkan gambar atau untuk membuat program grafis. Dengan canvas, kita bisa menggambar berbagai bentuk seperti lingkaran, segitiga, dll.
3. User Interface (UI) components: contohnya adalah buttons, list, simple popup menus, check boxes, text fields, dan elemen lain
4. Komponen pembentuk window: seperti frames, menu bars, windows, dan dialog boxes.

Komponen-komponen Dasar User Interface dengan Swing

Berikut adalah beberapa komponen dasar antarmuka pengguna (*user interface*) yang disediakan oleh Swing:

1. *Top level containers*: adalah container dasar untuk meletakkan komponen-komponen lain. Contohnya adalah JFrame, JDialog, dll.
2. *Intermediate level containers*: merupakan container perantara. Umumnya digunakan hanya untuk meletakkan atau mengelompokkan komponen-komponen yang digunakan, baik itu container atau berupa atomic component. Contoh dari *Intermediate level container* adalah JPanel (panel).
3. *Atomic component*: merupakan komponen yang memiliki fungsi khusus. Umumnya pengguna langsung berinteraksi dengan komponen ini. Contohnya adalah JButton (tombol), JLabel (label), JTextField, JTextArea (area untuk menulis teks), dll.
4. *Layout Manager* Berfungsi untuk mengatur bagaimana tata letak/posisi dari komponen yang akan diletakkan, satu sama lain di dalam suatu container. Secara default ada 6 buah layout : BorderLayout, BoxLayout, FlowLayout, CardLayout, GridBagLayout & GridLayout
5. *Event Handling* : Menangani event yang dilakukan oleh user seperti menekan tombol, memperbesar atau memperkecil ukuran frame, mengklik mouse, mengetik sesuatu dengan keyboard, dll.

Konstruktor untuk membentuk JFrame adalah JFrame() atau JFrame(String NamaFrame). Salah satu method yang sering digunakan di kelas JFrame adalah setDefaultCloseOperation(int ops). Method ini berfungsi untuk mengatur apa yang akan dilakukan oleh program ketika tombil close di klik. Parameter method tersebut di antaranya adalah:

1. JFrame.DO NOTHING ON CLOSE: tidak ada satu aktivitas pun yang dilakukan secara otomatis apabila frame ditutup.
2. JFrame.HIDE ON CLOSE: frame hanya disembunyikan, namun secara fisik frame masih ada di memori sehingga dapat dimunculkan kembali. Merupakan aktivitas default.
3. JFrame.DISPOSE ON CLOSE: menghilangkan tampilan frame dari layar, menghilangkannya dari memori, dan membebaskan resource yang dipakai.
4. JFrame.EXIT ON CLOSE: menghentikan eksekusi program.

Top level container yang lain adalah JDialog. Berbeda dengan JFrame, JDialog tidak dibuat berdiri sendiri, melainkan dibuat bersama-sama dengan frame sebagai *parent*-nya. Jika frame *parent*-nya ditutup, maka dialog akan dihapus dari memori. Kemunculan dialog akan membuat semua input terhadap frame *parent*-nya akan terblokir sampai dialog tersebut ditutup.

Layout Management

Untuk mengatur layout dari setiap komponen-komponen GUI yang diletakkan pada container, digunakan *layout manager*. Setiap pane secara default pasti memiliki layout manager. Jika ingin mengubah layout-nya, gunakan void setLayout(LayoutManager mgr) Java menyediakan 6 buah class standar yang dapat digunakan sebagai layout manager yang terdapat dalam package java.awt, yaitu FlowLayout , GridLayout , BorderLayout , CardLayout ,GridBagLayout , BoxLayout

1. Flow Layout merupakan layout manager default yang digunakan JPanel. Layout manager ini meletakkan komponen yg ada secara berurutan dari kiri ke kanan dan berpindah baris jika dibutuhkan. Jarak antar komponen baik secara vertikal maupun horizontal dapat ditentukan sendiri. Selain itu, kita bisa menentukan alignment dari komponen yang diletakkan, yaitu rata kanan, rata kiri atau di tengah
2. GridLayout merupakan layout manager berbentuk grid. Semua komponen dalam layout ini memiliki ukuran yang sama. Konstruktor GridLayout adalah sebagai berikut.
 - a. GridLayout(rows, columns, hGap, vGap). Konstruktor ini mendefinisikan jumlah baris, kolom, dan ukuran gap horizontal/vertikal antar elemen dalam pixel.
 - b. GridLayout(rows, columns). Konstruktor yang digunakan dengan nilai default hGap dan vGap sama dengan 0.
3. BorderLayout adalah default manager untuk ContentPane. Layout ini menempatkan komponen dengan pendekatan arah mata angin north, south, west, center, east.
4. Cardlayout digunakan untuk menampilkan container-container/panel-panel seperti tumpukan kartu. Hanya satu container yang akan tampil untuk satu waktu. komponen yang diletakkan dengan layout ini berupa object yang bertipe intermediate container yang di dalamnya terdapat atomic component. Salah satu contoh intermediate container adalah tabbed pane (JTabbedPane)
5. GridBagLayout
Grid bag layout adalah pengaturan layout yang dapat digunakan untuk meletakkan komponen secara bebas. Setiap komponen dapat menempati ukuran yang berbeda dengan komponen lainnya.
6. BoxLayout digunakan untuk meletakkan komponen berurutan ke kanan (sumbu X) atau berurutan ke bawah (sumbu Y). Penggunaan BoxLayout menggunakan class Box dimana

merupakan container yang secara internal telah menggunakan BoxLayout sebagai layout manager-nya.

LEMBAR KERJA DAN TUGAS

1. Membuat Frame

Script

```
import javax.swing.*;
public class ContohFrame1{
public static void main(String[] args){
JFrame a1;
a1 = new JFrame("contoh frame");
a1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
a1.setSize(300,300);
a1.setVisible(true);
}
}
```

2. Membuat Dialog

Script

```
import javax.swing.*;
public class Dialogku{
public static void main (String[] args){
JFrame a2;
a2 = new JFrame("frame");
a2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
a2.setVisible(true);
 JOptionPane.showConfirmDialog(a2,"Contoh Dialog","UMSIDA KAMPUSKU",
JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE);
}
}
```

3. Membuat GridLayout

Script

```
import javax.swing.*;
import java.awt.*;
public class Contohgridlayout {
    public static void main(String[] args) {
        JFrame frame = new JFrame ("UMSIDA KAMPUSKU");
        GridLayout layout = new GridLayout(3,2);
        layout.setHgap(5);
        layout.setVgap(10);
        frame.getContentPane().setLayout(layout);

        JButton btn1 = new JButton("pencet 1");
        JButton btn2 = new JButton("pencet 2");
        JButton btn3 = new JButton("pencet 3");
        JButton btn4 = new JButton("pencet 4");

        frame.getContentPane().add(btn1);
        frame.getContentPane().add(btn2);
        frame.getContentPane().add(btn3);
        frame.getContentPane().add(btn4);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
    }
}
```

```

        frame.show();
    }
}

```

4. Membuat Flowlayout

Script

```

import javax.swing.*;
import java.awt.*;

public class Contohflowlayout1 {
    public static void main(String[] args) {
        JFrame frame = new JFrame ("UMSIDA KAMPUSKU");
        FlowLayout layout = new FlowLayout(FlowLayout.LEFT);
        layout.setVgap(15);
        layout.setHgap(15);
        frame.getContentPane().setLayout(layout);

        JButton btn1 = new JButton("Umsida 1");
        JButton btn2 = new JButton("Umsida 2");
        JButton btn3 = new JButton("Umsida 3");
        JButton btn4 = new JButton("Umsida 4");

        frame.getContentPane().add(btn1);
        frame.getContentPane().add(btn2);
        frame.getContentPane().add(btn3);
        frame.getContentPane().add(btn4);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300,300);
        frame.show();
    }
}

```

5. Membuat Border

Script

```

import javax.swing.*;
import java.awt.*;

public class Contohborder {
    public static void main(String[] args) {
        JFrame frame = new JFrame ("Contoh Border Layout");
        BorderLayout layout = new BorderLayout(1,1);
        frame.getContentPane().setLayout(layout);
        JButton btnNorth = new JButton(" NORTH");
        JButton btnSouth = new JButton(" SOUTH");
        JButton btnEast = new JButton(" EAST");
        JButton btnWest = new JButton(" WEST");
        JButton btnCenter = new JButton(" CENTER");
        frame.getContentPane().add(btnNorth,BorderLayout.NORTH);
        frame.getContentPane().add(btnSouth,BorderLayout.SOUTH);
        frame.getContentPane().add(btnEast,BorderLayout.EAST);
        frame.getContentPane().add(btnWest,BorderLayout.WEST);
        frame.getContentPane().add(btnCenter,BorderLayout.CENTER);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.pack();
            frame.show();
    }
}

```

6. Membuat Cardlayout

Script

```
import javax.swing.*;
import java.awt.*;
public class Contohcardlayout {
    public static void main(String[] args) {
        JFrame frame = new JFrame ("UMSIDA KAMPUSKU");
        JPanel panel1 = new JPanel();
        JPanel panel2 = new JPanel();
        JPanel panel3 = new JPanel();
        JButton btn1 = new JButton("Tombol Umsida1");
        JButton btn2 = new JButton("Tombol Umsida2");
        JButton btn3 = new JButton("Tombol Umsida2");

        panel1.add(btn1);
        panel2.add(btn2);
        panel3.add(btn3);

        JTabbedPane tab = new JTabbedPane();
        tab.add(panel1,"sisi 1");
        tab.add(panel2,"sisi 2");
        tab.add(panel3,"sisi 3");

        frame.getContentPane().add(tab,BorderLayout.NORTH);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300,150);
        frame.show();
    }
}
```

7. Membuat GridBagLayout

Script

```
import javax.swing.*;
import java.awt.*;
public class Contohgridbag {
    public static void main(String[] args) {
        JFrame frame = new JFrame ("Contoh GridBagLayout
UMSIDA");
        GridBagLayout layout = new GridBagLayout();
        GridBagConstraints c = new GridBagConstraints();
        frame.getContentPane().setLayout(layout);
        c.fill = GridBagConstraints.HORIZONTAL;

        JButton btn1 = new JButton("Tombol umsida 1");
        c.weightx = 0.5
        c.gridx = 0;
        c.gridy = 0;
        layout.setConstraints(btn1,c);
        frame.getContentPane().add(btn1);

        JButton btn2 = new JButton("Tombol umsida 2");
        c.gridx = 1;
        c.gridy = 0;
        layout.setConstraints(btn2,c);
        frame.getContentPane().add(btn2);

        JButton btn3 = new JButton("Tombol umsida 3");
        c.ipady = 30;
    }
}
```

```

c.gridx      = 0;
c.gridy      = 1;
layout.setConstraints(btn3,c);
frame.getContentPane().add(btn3);

JButton btn4 = new JButton("Tombol umsida 4");
c.ipady     = 50;
c.gridx      = 1;
c.gridy      = 0;
c.gridx      = 2;
layout.setConstraints(btn4,c);
frame.getContentPane().add(btn4);

JButton btn5 = new JButton("Tombol umsida 5");
c.ipady     = 0;
c.gridx      = 1;
c.gridy      = 1;
c.gridy      = 2;
c.insets    = new Insets(10,0,0,0); //t,l,b,r
c.anchor    = GridBagConstraints.SOUTH;
layout.setConstraints(btn5,c);
frame.getContentPane().add(btn5);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.pack();
frame.show();
}
}

```

8. Membuat Boxlayout

Script	<pre> import javax.swing.*; import java.awt.*; public class Contohboxlayout { public static void main(String[] args) { JFrame frame = new JFrame (" Box Layout"); Box comp = new Box(BoxLayout.X_AXIS); JButton btn1 = new JButton("Posisi 1"); JButton btn2 = new JButton("Posisi 2"); JButton btn3 = new JButton("Posisi 3"); JButton btn4 = new JButton("Posisi 4"); JButton btn5 = new JButton("Posisi 5"); comp.add(btn1); comp.add(btn2); comp.add(btn3); comp.add(btn4); comp.add(btn5); frame.getContentPane().add(comp); frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE); frame.pack(); frame.show(); } } </pre>
--------	---

TUGAS

Buatlah project dengan adanya textfield sebagai inputan nim, nama dan alamat. Dan tambahkan button ok dan cancel ?

