

Laporan Resmi Praktikum Pemrograman Berorientasi Obyek



Nama: Rizki Wahyu Nurcahyani Fajarwati
Kelas: D4 Teknik Informatika B
NRP: 3120600042

**PROGRAM STUDI TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2021**

Operator

Tugas Pendahuluan

Buatlah makalah yang berisi tentang berbagai macam operator dengan disertai contoh penggunaan dan outputnya.

Pendahuluan

Latar Belakang

Pemrograman adalah proses menulis, menguji dan memperbaiki (*debug*), dan memelihara kode yang membangun suatu program komputer.^[1] Kode ini ditulis dalam berbagai bahasa pemrograman. Tujuan dari pemrograman adalah untuk memuat suatu program yang dapat melakukan suatu perhitungan atau 'pekerjaan' sesuai dengan keinginan si pemrogram. Untuk melakukan pemrograman, diperlukan keterampilan dalam algoritme, logika, bahasa pemrograman, dan pada banyak kasus, pengetahuan-pengetahuan lain seperti matematika.

Dalam pemrograman tentunya terdapat komponen di dalamnya salah satunya yaitu operator. operator merupakan simbol-simbol yang digunakan untuk melakukan sebuah operasi tertentu dan lebih banyak digunakan dalam fungsi matematika. Operator memiliki banyak sekali jenisnya, yaitu operator aritmatika, logika, bitwise, relasional dan juga masih banyak lagi. Pada kali ini saya akan membahas sedikit mengenai macam-macam operator

Rumusan Masalah

1. Apakah itu operator?
2. Apa saja macam-macam operator yang ada?
3. Bagaimana contoh penggunaan operator pada program?

Tujuan

1. Mendeskripsikan tentang operator
2. Menjelaskan mengenai macam-macam operator
3. Mengetahui dan memahami contoh penggunaan operator pada program

Pembahasan

Pengertian Operator

Ekspresi adalah statement yang mengembalikan suatu nilai sedangkan Operator adalah suatu symbol yang biasanya digunakan dalam ekspresi. Operator akan melakukan suatu operasi terhadap operand sesuai dengan fungsinya, seperti penjumlahan, pengurangan, perkalian, dan sebagainya. Operator merupakan pengendali pada operasi yang akan dilakukan pada beberapa operand sehingga membentuk ekspresi.

Operator dapat diklasifikasikan menjadi 2 bentuk, yaitu unary operator dan binary operator. Unary operator adalah operator yang hanya melibatkan 1 operand. Sedangkan binary operator adalah operator yang melibatkan 2 operand. Java mempunyai berbagai macam jenis operator yang dapat digolongkan menjadi operator aritmatika, increment-decrement, bitwise, boolean, logik, shift (geser), penugasan, kombinasi dan kondisi

Macam Operator

Di dalam pemrograman java terdapat 7 operator. Antara lain :

1. Arithmetic Operator

Arithmetic operator (operator aritmatika) adalah operator yang berfungsi untuk operasi aritmatika.

Arithmetic Operator	Keterangan
+	Operasi penambahan
-	Operasi pengurangan
*	Operasi perkalian
/	Operasi pembagian
%	Operasi modulus

2. Increment-Decrement Operator

Increment – decrement operator adalah operator yang berguna untuk menaikkan 1 nilai (increment) dan menurunkan 1 nilai (decrement). Yang termasuk increment-decrement operator bisa dilihat pada Tabel. Berdasarkan urutan eksekusi penaikan dan penurunan nilainya, increment decrement operator ini dapat diklasifikasikan menjadi 2 macam, yaitu pre-increment/decrement dan post-increment/decrement

Increment-Decrement Operator	Keterangan
++	increment
--	decrement

Operator	Contoh	Penjelasan
Pre-increment	<code>++a</code>	Tambah a sebanyak 1 angka, lalu tampilkan hasilnya
Post-increment	<code>a++</code>	Tampilkan nilai a, lalu tambah a sebanyak 1 angka
Pre-decrement	<code>--a</code>	Kurangi a sebanyak 1 angka, lalu tampilkan hasilnya
Post-decrement	<code>a--</code>	Tampilkan nilai a, lalu kurangi a sebanyak 1 angka

3. BitWise Operator

Bitwise operator adalah operator yang dipakai untuk operasi bit pada nilai operan. Yang termasuk bitwise operator bisa dilihat pada Tabel.

Bitwise Operator	Keterangan
<code>~</code>	Operasi complement
<code>&</code>	Operasi AND
<code> </code>	Operasi OR
<code>^</code>	Operasi XOR

A = 10 => 1010 (Binary)

B = 7 => 111 (Binary)

A & B = 1010

&

0111

= 0010

= 2 (Decimal)

JournalDev

Bitwise AND Operator

4. Boolean Operator

Boolean operator (operator boolean) adalah operator yang mengharuskan operannya bertipe boolean (true atau false). Yang termasuk boolean operator bisa dilihat pada Tabel

Logical Operator	Keterangan
!	Operasi negasi (NOT)
&	Operasi AND
	Operasi OR
^	Operasi XOR
&&	Operasi AND (short circuit)
	Operasi OR (short circuit)

5. Logical Operator

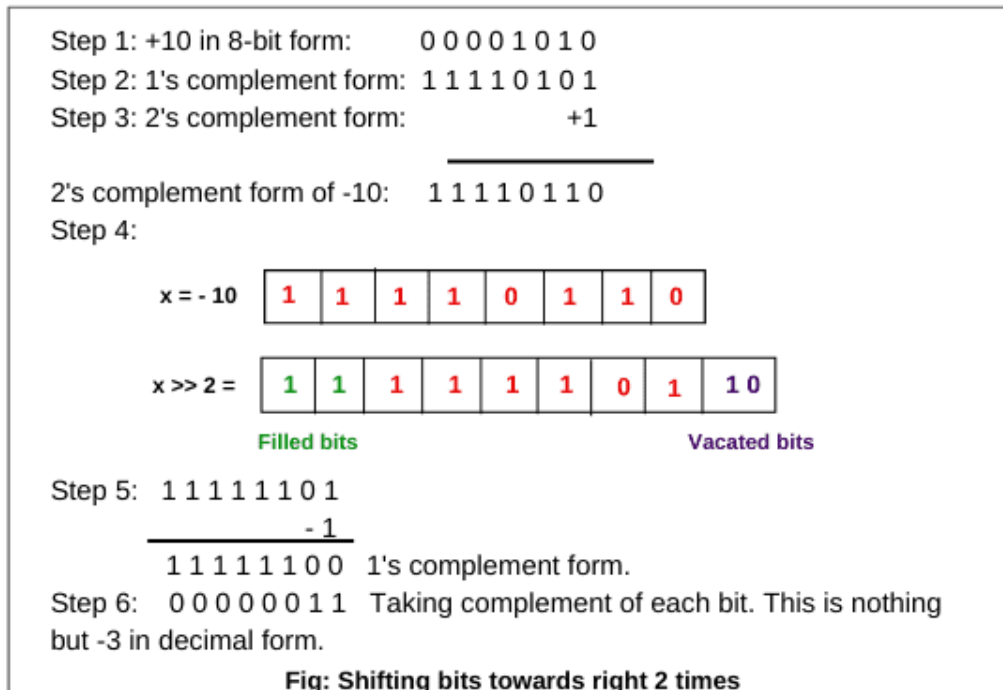
Operator `!`, `&`, `|` dan `^` mempunyai implementasi yang sama sebagaimana ketika ia menjadi bitwise operator. Hanya saja di logical operator, operan yang dilibatkan disini harus bertipe boolean, yang hanya mempunyai nilai true atau false. Logical operator (operator logika) adalah operator yang sering dipakai untuk operasi perbandingan dan selalu menghasilkan suatu nilai bertipe boolean (true atau false). Yang termasuk logical operator bisa dilihat pada Tabel

Logical Operator	Keterangan
<code>==</code>	Operasi perbandingan sama dengan
<code>!=</code>	Operasi perbandingan tidak sama dengan
<code>></code>	Operasi perbandingan lebih besar
<code>>=</code>	Operasi perbandingan lebih besar sama dengan
<code><</code>	Operasi perbandingan lebih kecil
<code><=</code>	Operasi perbandingan lebih kecil sama dengan

6. Shift Operator

Shift operator (operator geser) adalah operator yang berfungsi untuk menggeser susunan bit pada suatu nilai. Yang termasuk dalam shift operator dapat dilihat pada Tabel.

Shift Operator	Keterangan
<code>>></code>	right shift
<code>>>></code>	unsigned right shift
<code><<</code>	left shift



7. Combination Operator

Combination operator (operator kombinasi) adalah operator yang terdiri dari gabungan 2 operator. Biasanya combination operator ini dipakai untuk mempersingkat waktu penulisan program. Yang termasuk operator combination bisa dilihat pada Tabel .

Combination Operator	Keterangan
+=	Gabungan dari operator = dan +
-=	Gabungan dari operator = dan -
*=	Gabungan dari operator = dan *
/=	Gabungan dari operator = dan /
%=	Gabungan dari operator = dan %
>>=	Gabungan dari operator = dan >>
>>>=	Gabungan dari operator = dan >>>
<<=	Gabungan dari operator = dan <<
&=	Gabungan dari operator = dan &
=	Gabungan dari operator = dan
^=	Gabungan dari operator = dan ^

Contoh Operator

1. Operator Aritmatika

```

1 package pertemuan3;
2
3
4 public class KelilingPersegiPanjang {
5
6     public static void main(String[] args) {
7         //deklarasi
8         double kll, p, l;
9
10        // deskripsi|
11        // -- Input
12        p = 6.0;
13        l = 3.0;
14
15        // -- proses
16        kll = (2*p) + (2*l);
17
18        // -- output
19        System.out.println(kll);
20    }
21 }
22
23

```

Outputnya : **18.0**

2. Operator Increment-Decrement

```

1 import java.util.Scanner;
2
3
4 public class Demo {
5
6     public static void main(String[] args) {
7
8         int x = 100;
9         int y = 10;
10
11        System.out.println(x++);
12        System.out.println(x);
13
14        System.out.println(y--);
15        System.out.println(y);
16
17    }
18 }
19
20

```

```

naisalp9@NaisalSP9-Dell-XPS:~/Desktop$ java Demo
100
101
10
9
naisalp9@NaisalSP9-Dell-XPS:~/Desktop$

```

3. Bitwise Operator

BitwiseAndExample.java

```
public class BitwiseAndExample
{
    public static void main(String[] args)
    {
        int x = 9, y = 8;
        // bitwise and
        // 1001 & 1000 = 1000 = 8
        System.out.println("x & y = " + (x & y));
    }
}
```

Output

```
x & y = 8
```

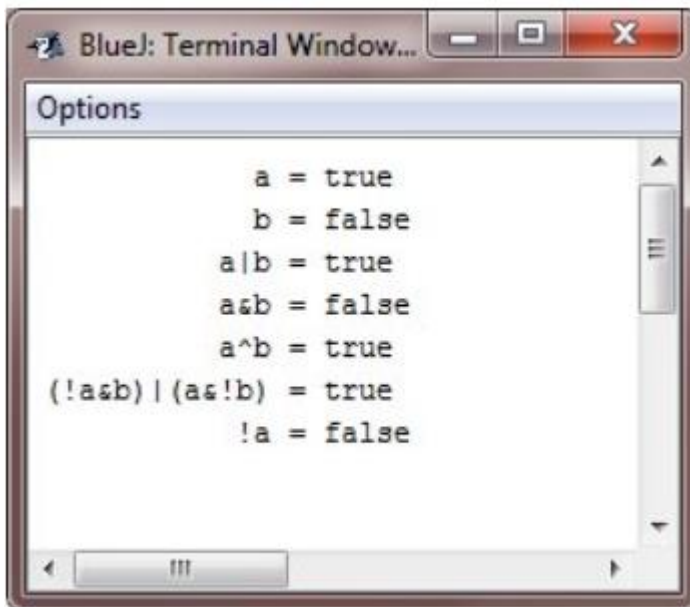
4. Boolean Operator

```
/* Java Program Example - Java Boolean Logical Operators
 * Demonstrate the boolean logical operators. */

public class JavaProgram
{
    public static void main(String args[])
    {
        boolean a = true;
        boolean b = false;

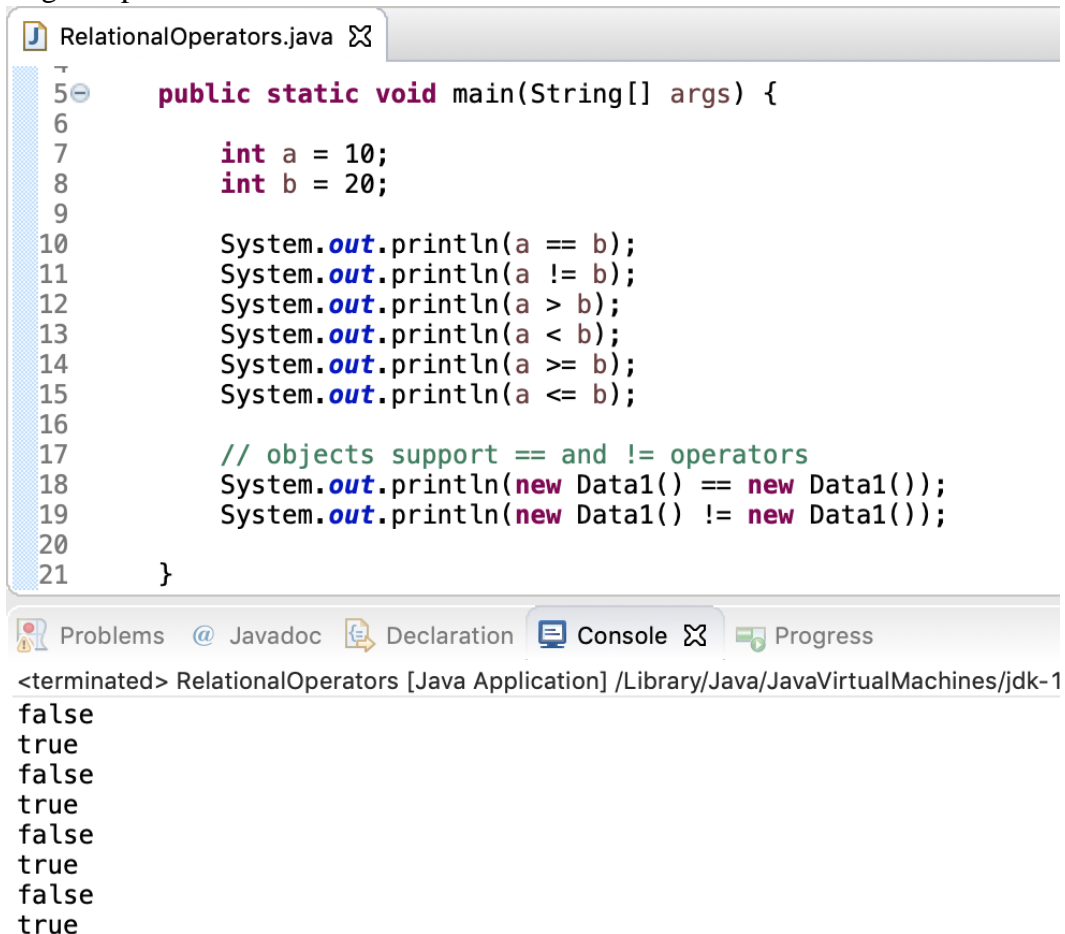
        boolean c = a | b;
        boolean d = a & b;
        boolean e = a ^ b;
        boolean f = (!a & b) | (a & !b);
        boolean g = !a;

        System.out.println("          a = " + a);
        System.out.println("          b = " + b);
        System.out.println("          a|b = " + c);
        System.out.println("          a&b = " + d);
        System.out.println("          a^b = " + e);
        System.out.println("(!a&b)|(a&!b) = " + f);
        System.out.println("          !a = " + g);
    }
}
```

```
Options
    a = true
    b = false
    a|b = true
    a&b = false
    a^b = true
    (!a&b) | (a&!b) = true
    !a = false
```

5. Logika operator



```
RelationalOperators.java
5 public static void main(String[] args) {
6
7     int a = 10;
8     int b = 20;
9
10    System.out.println(a == b);
11    System.out.println(a != b);
12    System.out.println(a > b);
13    System.out.println(a < b);
14    System.out.println(a >= b);
15    System.out.println(a <= b);
16
17    // objects support == and != operators
18    System.out.println(new Data1() == new Data1());
19    System.out.println(new Data1() != new Data1());
20
21 }
```

<terminated> RelationalOperators [Java Application] /Library/Java/JavaVirtualMachines/jdk-1

```
false
true
false
true
false
true
false
true
```

6. Shift operator

```
public class OperatorShifting
{
    public static void main(String args[])
    {
        byte x, y;
        x=10;
        y=-10;
        System.out.println("Bitwise Left Shift: x<<2 = "+(x<<2));
        System.out.println("Bitwise Right Shift: x>>2 = "+
(x>>2));
        System.out.println("Bitwise Zero Fill Right Shift: x>>>2 = "+(x>>>2));
        System.out.println("Bitwise Zero Fill Right Shift: y>>>2 = "+
(y>>>2));
    }
}
```

```
Bitwise Left Shift: x<<2 = 40
Bitwise Right Shift: x>>2 = 2
Bitwise Zero Fill Right Shift: x>>>2 = 2
Bitwise Zero Fill Right Shift: y>>>2 = 1073741821
```

7. Combination Operator

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

        // create variables
        int a = 4;
        int var;

        // assign value using =
        var = a;
        System.out.println("var using =: " + var);

        // assign value using +=
        var += a;
        System.out.println("var using +=: " + var);

        // assign value using *=
        var *= a;
        System.out.println("var using *=: " + var);
    }
}
```

```
var using =: 4
var using +=: 8
var using *=: 32
```

Penutup

Kesimpulan

Operator merupakan komponen yang diperlukan dalam sebuah pemrograman, setiap operator memiliki fungsi yang berbeda-beda. Operator memiliki banyak macamnya, yang dibedakan menurut fungsinya. Dengan adanya operator, memudahkan seseorang dalam mengerjakan pemrogramannya

Daftar Pustaka

PetaniKode, diakses pada (September, 7 2021)

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.petanikode.com%2Fjava-operator%2F&psig=AOvVaw3jIq2WAMKoh3-SCtoaEIjQ&ust=1631090135631000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCNjI-8e67PICFQAAAAAdAAAAABAD>

Modul Praktikum Pemrograman Berbasis Obyek "Operator", diakses pada (September, 7 2021)

Wikipedia, diakses pada (September, 7 2021) <https://id.wikipedia.org/wiki/Pemrograman>

Blog Agung Trianggoro, diakses pada (September, 7 2021) <https://agung-trianggoro.blogspot.com/2019/10/makalah-tipe-data-variable-operator.html>

DuniaIlkom, diakses pada (September, 7 2021) <https://www.duniailkom.com/tutorial-belajar-java-jenis-jenis-operator-increment-decrement-bahasa-java/>

W3resource, diakses pada (September, 7 2021)

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.w3resource.com%2Fjava-tutorial%2Fjava-logical-operators.php&psig=AOvVaw0xtG95xeWhi90e6kZqJzLj&ust=1631091446489000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCNDm4bm_7PICFQAAAAAdAAAAABAJ

Developerhelps, diakses pada (September, 7 2021)

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.developerhelps.com%2Fjava-bitwise-operators%2F&psig=AOvVaw0fdpAWMDBwo6hFW9Os_bmZ&ust=1631091374838000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCPjn55W_7PICFQAAAAAdAAAAABAD

Blog javasedu, diakses pada (September, 7 2021)

https://www.google.com/url?sa=i&url=http%3A%2F%2Fjavaseeedu.blogspot.com%2F2017%2F10%2Fjava-postfix-operators-increment.html&psig=AOvVaw0wUQtVvMTuEPKKQTBwVDsa&ust=1631091294233000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCODhiai_7PICFQAAAAAdAAAAABAD

Percobaan 1

a. Listing Program

```
class InDec {
    public static void main(String [] args) {
        int x = 8, y = 13;

        System.out.println("y = " + y);
        System.out.println("x = " + ++x);
        System.out.println("y = " + y++);
        System.out.println("x = " + x--);
        System.out.println("y = " + --y);
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac InDec.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java InDec
y = 13
x = 9
y = 13
x = 9
y = 13
```

- c. Analisa : pada program diatas, terdapat percobaan decrement dan increment. Terdapat pre-increment yaitu tambahkan nilai x terlebih dahulu kemudian ditampilkan hasilnya. Lalu ada post increment yaitu menampilkan hasilnya terlebih dahulu kemudian baru di tambahkan. Untuk decrementnya sama seperti incrementnya. Bedanya decrement itu dikurangi, jika increment itu ditambah.

Percobaan 2

a. Listing Program

```
class Bitwise {
    public static void main(String [] args) {
        int x = 5, y = 6;

        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("x & y = " + (x & y));
        System.out.println("x | y = " + (x | y));
        System.out.println("x ^ y = " + (x ^ y));
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac Bitwise.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java Bitwise
x = 5
y = 6
x & y = 4
x | y = 7
x ^ y = 3
```

- c. Analisa : pada program diatas terdapat operator bitwise yaitu operator untuk menangani operasi logika bilangan biner dalam bentuk bit. Bilangan decimal akan di terjemahkan ke dalam biner, kemudian di operatorkan dengan bilangan yang lain. Lalu hasilnya akan di translate lagi ke decimal.

Percobaan 3

a. Listing Program

```
class BitiseComplement {
    public static void main(String [] args) {
        int x = 8;
        System.out.println("x = " + x);
        int y = ~x;
        System.out.println("y = " + y);
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac BitiseComplement.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java BitiseComplement
x = 8
y = -9
```

- c. Analisa : pada program diatas merupakan program yang menggunakan operator yang digunakan untuk operasi komplemen. Jadi bilangan yang dimasukkan akan di komplemenkan baru kemudian ditampilkan. Menggunakan symbol (~).

Percobaan 4

a. Listing Program

```
public class Shift {
    public static void main (String [] args) {
        int x = 7;
        System.out.println("x = " + x);
        System.out.println("x >> 2 = " + (x >> 2));
        System.out.println("x << 1 = " + (x << 1));
        System.out.println("x >>> 1 = " + (x >>> 1));
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac Shift.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java Shift
x = 7
x >> 2 = 1
x << 1 = 14
x >>> 1 = 3
```

- c. Analisa : pada program diatas, Shift operator (operator geser) adalah operator yang berfungsi untuk menggeser susunan bit pada suatu nilai. Diatas terlihat jika bilangan diubah ke biner, kemudian di geser, maka hasilnya seperti output diatas.

Percobaan 5

a. Listing Program

```
class LogicalOperator {
    public static void main(String [] args) {
        int x = 7, y = 11, z = 11;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("z = " + z);
        System.out.println("x < y = " + (x < y));
        System.out.println("x > z = " + (x > z));
        System.out.println("y <= z = " + (y <= z));
        System.out.println("x >= y = " + (x >= y));
        System.out.println("y == z = " + (y == z));
        System.out.println("x != y = " + (x != z));
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac LogicalOperator.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java LogicalOperator
x = 7
y = 11
z = 11
x < y = true
x > z = false
y <= z = true
x >= y = false
y == z = true
x != y = true
```

- c. Analisa : pada program diatas, terdapat berbagai macam operator logika. Terdapat <, >, >=, <=, ==, !=. hasilnya true or false. Seperti contoh 7 < 11 maka hasilnya true.

Percobaan 6

a. Listing Program

```
public class BooleanAnd {
    public static void main(String [] args) {
        int a = 5, b = 7;
        if ((a < 2) & (b++ < 10)) b += 2;
        System.out.println(b);
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac BooleanAnd.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java BooleanAnd
8
```

- c. Analisa : pada program diatas, hasilnya 8 dikarenakan adanya increment, meskipun statement yang ada di if tidak dijalankan. Bernilai false dikarenakan salah satu kondisinya false. Namun masih mengecek nilai yang ada di samping sehingga nilai b di increment

Percobaan 7

a. Listing Program

```
public class ShortCircuitBooleanAnd {
    public static void main(String [] args) {
        int a = 5, b = 7;
        if ((a < 2) && (b++ < 10)) b += 2;
        System.out.println(b);
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac ShortCircuitBooleanAnd.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java ShortCircuitBooleanAnd
7
```

- c. Analisa : pada program diatas, beda dengan program sebelumnya adalah, saat bernilai false, kondisi yang disampingnya masih di cek, sedangkan pada program ini Ketika kondisi sudah false. Maka kondisi kedua langsung false, dan langsung di skip.

Percobaan 8

a. Listing Program

```
public class BooleanOr {
    public static void main(String [] args) {
        int a = 5, b = 7;
        if ((a > 2) | (b++ < 10)) b += 2;
        System.out.println(b);
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac BooleanOr.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java BooleanOr
10
```

- c. Analisa : pada program diatas, menggunakan OR dimana outputnya menjadi 10 dikarenakan kedua pernyataan benar, kemudian ada increment, dan juga ada penambahan 2.

Percobaan 9

a. Listing Program

```
public class ShortCircuitBooleanOr {
    public static void main(String [] args) {
        int a = 5, b = 7;
        if ((a > 2) || (b++ < 10)) b += 2;
        System.out.println(b);
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac ShortCircuitBooleanOr.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java ShortCircuitBooleanOr
9
```

- c. Analisa : pada program diatas, output menjadi 9 dikarenakan sebelumnya sudah ditambahkan 2.

Percobaan 10

a. Listing Program

```
class Conditional {
    public static void main(String [] args) {
        int x = 0;
        boolean isEven = false;
        System.out.println("x = " + x);
        x = isEven ? 4 : 7;
        System.out.println("x = " + x);
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> javac Conditional.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\3.Prak2.1> java Conditional
x = 0
x = 7
```

- c. Analisa : pada program diatas, mengapa bernilai 7, dikarenakan pada sebelumnya IsEven sudah bernilai false, dan 4 : 7 itu berarti true : false, karena false maka yang ditampilkan adalah 7. Namun jika sebelumnya bernilai true, maka akan ditampilkan 4

Kondisional & Perulangan

Tugas Pendahuluan

1. Sebutkan dan jelaskan berbagai macam sintaks percabangan yang digunakan di Java!
Percabangan di dalam java terdapat 3 macam, yaitu if, if else, dan switch.

- If merupakan percabangan yang menyatakan pernyataan atau statement yang akan di eksekusi jika kondisi atau persyaratan terpenuhi. Teman dari if sendiri adalah else, jadi nilai yang ada di dalam if itu benar atau salah, jika tidak memenuhi biasanya akan dilempar ke else. Contoh sintaksnya adalah :

```
If(ekspresi_boolean){  
    Pernyataan 1;  
}else {  
    Pernyataan 2;  
}
```

- Else if/if else merupakan percabangan yang mirip dengan if, namun bedanya disini bisa terdapat kondisi lebih dari 1, yang mana kondisi tersebut masuk ke dalam else if. Jika kondisi pertama salah, maka akan di lempar ke kondisi ke 2, begitu sampai jika tidak ada yang cocok, akan dilempar ke else, untuk bentuk syntaxnya :

```
if (ekspresi_boolean1) {  
    Pernyataan1;  
} else if (ekspresi_boolean2) {  
    Pernyataan2;  
} else {  
    Pernyataan3;  
}
```

- Switch dipakai pada saat kita ingin memberikan kondisi dengan beberapa syarat yang identik yang masing-masing mempunyai pernyataan yang berbeda-beda. Pada Java, nilai yang dilewatkan pada switch harus bertipe int, short, byte atau char. Untuk sintaksnya sebagai berikut :

```
Switch (ekspresi){  
    Case nilai1 :  
        Pernyataan1;  
        Break;  
    Case nilai2 :  
        Pernyataan2;  
        Break;  
    Default :  
        Pernyataan3;  
}
```

2. Sebutkan dan jelaskan berbagai macam sintaks perulangan yang digunakan di Java!
Perulangan di dalam Java terdapat 3 macam, yaitu for, while dan do-while.

- Perulangan for digunakan saat kita mengetahui jumlah pasti untuk berhenti.syntaxnya:

```
For(inisialisasi; kondisi; iterasi){  
    Pernyataan1;  
}
```

- Perulangan while, perulangan ini terdapat pengecekan di awal masuk loopingnya, jadi jika tidak sesuai, maka pernyataan yang ada di dalam looping while tidak akan dijalankan. Looping ini digunakan saat kita belum mengetahui jumlahnya. Untuk syntaxnya:

```
While(kondisi){  
    Pernyataan;  
}
```

- Perulangan do-while, perulangan ini langsung menuju ke pernyataan pada do, dikarenakan pengecekannya berada di akhir, yaitu di while. Kemudian jika tidak cocok dengan while, akan keluar, namun pernyataan pada do sudah di proses. Syntaxnya :

```
Do{  
    Pernyataan;  
}while(kondisi);
```

Percobaan 1

a. Listing program

```
class IfElseName {
    public static void main(String [] args) {
        char firstInitial = 'a';
        System.out.println("Masukkan huruf awal nama anda: ");
        try {
            firstInitial = (char) System.in.read();
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }
        if (firstInitial == 'a')
            System.out.println("Nama anda pasti Asep!");
        else if (firstInitial == 'b')
            System.out.println("Nama anda pasti Brodin!");
        else if (firstInitial == 'c')
            System.out.println("Nama anda pasti Cecep!");
        else
            System.out.println("Nama anda tidak terkenal!");
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac IfElseName.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java IfElseName
Masukkan huruf awal nama anda:
A
Nama anda tidak terkenal!
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java IfElseName
Masukkan huruf awal nama anda:
a
Nama anda pasti Asep!
```

- c. Analisa : pada program diatas, terdapat percabangan if else, terdapat pengecekan, apakah first initial yang dimasukkan sebelumnya masuk ke dalam if, atau else if. Jika tidak terdapat maka akan pergi ke else, yang dimana elsenya terdapat pernyataan jika nama tidak terkenal, namun bila ada di salah satu kondisi, maka pernyataan yang ada akan di proses.

Percobaan 2

a. Listing program

```
public class SwitchName {
    public static void main(String [] args) {
        char firstInitial = 'a';
        System.out.println("Masukkan huruf awal nama anda:");
        try {
            firstInitial = (char) System.in.read();
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }
        switch (firstInitial) {
            case 'a':
                System.out.println("Nama anda pasti Asep!");
            case 'b':
                System.out.println("Nama anda pasti Brodin!");
            case 'c':
                System.out.println("Nama anda pasti Cecep!");
            default:
                System.out.println("Nama anda tidak terkenal!");
        }
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac SwitchName.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java SwitchName
Masukkan huruf awal nama anda:a
Nama anda pasti Asep!
Nama anda pasti Brodin!
Nama anda pasti Cecep!
Nama anda tidak terkenal!
```

- c. Analisa : pada program diatas, menggunakan percabangan switch. Mengapa outputnya tercetak semua, dikarenakan tidak adanya break yang menghentikan jika suatu pernyataan sudah diproses, pernyataan yang ada dibawahnya tidak perlu diproses.

Percobaan 3

a. Listing program

```
public class SwitchNameBreak {
    public static void main(String [] args) {
        char firstInitial = 'a';
        System.out.print("Masukkan Huruf Awal Nama Anda : ");
        try {
            firstInitial = (char) System.in.read();
        } catch (Exception e) {
            System.out.println("Error : " + e.toString());
        }
        switch (firstInitial) {
            case 'a':
                System.out.println("Nama Anda pasti Asep!");
                break;
            case 'b':
                System.out.println("Nama anda pasti Brodin!");
                break;
            case 'c':
                System.out.println("Nama anda pasti Cecep!");
                break;
            default:
                System.out.println("Nama anda tidak terkenal!");
        }
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac SwitchNameBreak.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java SwitchNameBreak
Masukkan Huruf Awal Nama Anda : c
Nama anda pasti Cecep!
```

- c. Analisa : pada program diatas, merupakan program yang sama dengan nomor percobaan sebelumnya, namun bedanya disini terdapat break. Yang berarti keluar dari percabangannya. Langsung ke step selanjutnya. Dan tentunya tidak memproses pernyataan di bawahnya. Sehingga pernyataanya hanya muncul yang berada dalam lingkup case tsb.

Percobaan 4

a. Listing program

```
class ForCount {
    public static void main(String [] args){
        int count = 1;
        for(int i = 0; i < 9; i++){
            for(int j = 0; j < i+1; j++){
                System.out.println(count);
            }count++;
            System.out.println();
        }
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac ForCount.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java ForCount
1
2
2
3
3
3
4
4
4
4
5
5
5
5
5
6
6
6
6
6
6
7
7
7
7
7
7
7
7
8
8
8
8
8
8
8
8
9
9
9
9
9
9
9
9
9
```

c. Analisa : pada program diatas, menggunakan perulangan for. Dikarenakan mengetahui jumlah untuk berhentinya.

Percobaan 5

a. Listing program

```
class WhileCount {
    public static void main(String [] args) {
        int count = 1;
        int i = 0;
        while (i < 9) {
            int j = 0;
            while (j < i + 1) {
                System.out.print(count);
                j++;
            }
            count++;
            System.out.println();
            i++;
        }
    }
}
```

b.

```
22
333
4444
55555
666666
7777777
88888888
99999999
```

- c. Analisa : pada program diatas, terdapat 2 perulangan kedua-duanya menggunakan while, yang didalamnya terdapat kondisi. Jika $i < 9$ kemudian $j < i + 1$, kemudian hasilnya dicetak. Terdapat 2 increment. Yaitu count dan i

Percobaan 6

a. Listing program

```
class BreakLoop {
    public static void main(String [] args) {
        int i = 0;
        do {
            System.out.println("Iterasi ke " + i);
            i++;
            if (i > 10) break;
        } while (true);
    }
}
```

b.

```
Iterasi ke 2
Iterasi ke 3
Iterasi ke 4
Iterasi ke 5
Iterasi ke 6
Iterasi ke 7
Iterasi ke 8
Iterasi ke 9
Iterasi ke 10
```

- c. Analisa : pada program diatas, terdapat perulangan do while, dimana pernyataan di dalamnya di kerjakan dahulu baru kemudian di cek di while. Terdapat if, jika $i > 10$ maka break, yang mana berarti keluar dari looping dan menuju proses selanjutnya

Percobaan 7

a. Listing program

```
class ContinueLoop {
    public static void main(String [] args) {
        int a, b;
        for (a = 0; a < 2; a++)
            for (b = 0; b < 3; b++) {
                if (b == 1) continue;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac ContinueLoop.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java ContinueLoop
a=0 ; b=0
a=0 ; b=2
a=1 ; b=0
a=1 ; b=2
```

- c. Analisa : pada program diatas, menggunakan looping for, dilengkapi dengan continue pada looping ke 2, yang mana jika $b == 1$ maka akan di skip, langsung pergi ke perulangannya Kembali. Dikarenakan ini continue bukan break, maka program tidak akan keluar dari perulangan, namun hanya melewati nya saja.

Percobaan 8

a. Listing program

```
public class BreakLabel {
    public static void main(String [] args) {
        int a, b;
        Mulai:
        for (a = 0; a < 2; a++)
            for (b = 0; b < 3; b++) {
                if (b == 1)
                    break Mulai;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac BreakLabel.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java BreakLabel
a=0 ; b=0
```

- c. Analisa : pada program diatas, merupakan program dengan perulangan for dan percabangan if, serta terdapat break, yang dimana break mulai itu, keluar dari program yang dimulai dengan kata mulai. Jadi sejak mulai, itu tidak di proses.

Percobaan 9

a. Listing program

```
public class ContinueLabel {
    public static void main(String [] args) {
        int a, b;
        Mulai: for (a = 0; a < 2; a++)
            for (b = 0; b < 3; b++) {
                if (b == 1)
                    continue Mulai;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}
```

b.

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java ContinueLabel
a=0 ; b=0
a=1 ; b=0
```

- c. Analisa : pada program diatas, merupakan program dengan perulangan for dan percabangan if, serta terdapat continue, yang dimana continue mulai itu, melewati dari program yang dimulai dengan kata mulai.

Latihan soal

1. Buatlah program untuk menentukan kelompok suatu karakter yang dimasukkan melalui keyboard. Kelompok karakter tersebut adalah huruf kecil, huruf besar, angka, dan karakter khusus (tanda baca, operator dan sebagainya).

a. Listing program

```
import java.util.*;

public class CekKarakter {
    public static void main(String[] args) {
        char karakter = 'a', jawab = 'y';

        Scanner input = new Scanner(System.in);
        while (jawab == 'Y' || jawab == 'y') {
            System.out.print("Masukkan Sebuah Karakter : ");
            try {
                karakter = (char) System.in.read();
            } catch (Exception e) {
                System.out.print("Error : " + e.toString());
            }
            if (karakter >= 65 && karakter <= 96)
                System.out.println("Karakter Tersebut Merupakan Huruf Besar");
            else if (karakter >= 97 && karakter <= 122)
                System.out.println("Karakter Tersebut Merupakan Huruf Kecil");
            else if (karakter >= '0' && karakter <= '9')
                System.out.println("Karakter Tersebut Merupakan Angka");
            else
                System.out.println("Karakter Tersebut Merupakan Karakter Khusus");

            System.out.print("Ingin Lanjut lagi ? ");
            jawab = input.next().charAt(0);
        }
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac CekKarakter.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java CekKarakter
Masukkan Sebuah Karakter : A
Karakter Tersebut Merupakan Huruf Besar
Ingin Lanjut lagi ? y
Masukkan Sebuah Karakter : 1
Karakter Tersebut Merupakan Angka
Ingin Lanjut lagi ? y
Masukkan Sebuah Karakter : b
Karakter Tersebut Merupakan Huruf Kecil
Ingin Lanjut lagi ? y
Masukkan Sebuah Karakter : !
Karakter Tersebut Merupakan Karakter Khusus
Ingin Lanjut lagi ? t
```

- c. Analisa : pada program diatas, terjadi pengecekan, apakah bilangan tersebut masuk kedalam huruf besar, kecil, angka atau karakter khusus. Menggunakan if else sebagai caranya. Kemudian terdapat perulangan while yang akan terus berulang jika user selalu menjawab y atau Y
2. Bilangan bulat faktorial n, ditulis dengan n! adalah dihasilkan dari mengalikan dari 1 sampai dengan n. Contohnya $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$. Buatlah program untuk menampilkan tabel hasil faktorial dari suatu bilangan yang diinputkan (tampilan bilangan rata kanan)
 - a. Listing program

```

import java.util.*;

public class Faktorial {
    public static void main(String [] args){
        int bil, hasil = 1, x = 1;
        char jawab = 'y';

        Scanner input = new Scanner(System.in);

        while(jawab == 'y' || jawab == 'Y'){
            System.out.print("Masukkan Bilangan yang akan Di Faktorialkan : ");
            bil = input.nextInt();
            System.out.println("\tn\t\n!");
            System.out.println("=====\n");
            while(x <= bil){
                hasil *= x;
                System.out.println(+ x + "\t" + hasil);
                x++;
            }
            System.out.print("Ingin Lanjut lagi ? ");
            jawab = input.next().charAt(0);
        }
    }
}

```

b. Output

```

PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac Faktorial.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java Faktorial
Masukkan Bilangan yang akan Di Faktorialkan : 7
n      n!
=====
1      1
2      2
3      6
4      24
5      120
6      720
7      5040
Ingin Lanjut lagi ? t

```

- c. Analisa : pada program diatas, menggunakan perulangan while keduanya yaitu untuk perulangan yang terdapat kondisi apakah $x \leq bil$ jika iya maka pernyataan di dalamnya akan di proses, namun jika tidak maka akan keluar dari perulangan tsb, dan menjalankan step berikutnya, lalu perulangan yang kedua yaitu bertanya apakah ingin menjalankan program lagi atau kah tidak,

3. Buatlah program untuk menampilkan deret Fibonacci Contoh tampilan : Masukkan berapa deret Fibonacci? 8

8 deret Fibonacci = 1 1 2 3 5 8 13 21

a. Listing program

```
import java.util.*;

public class Fibonacci {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int pertama = 1, kedua = 1, ketiga = 0, tampsatu = 0, tampdua = 0,
        bil, x;
        char jawab = 'y';

        while (jawab == 'y' || jawab == 'Y') {
            System.out.print("Ingin Menginputkan Berapa Deret Fibonacci :
");
            bil = input.nextInt();
            if(bil == 1){
                System.out.print(pertama);
            }else{
                System.out.print(+pertama+ " , " +kedua);
            }
            tampsatu = pertama;
            tampdua = kedua;
            for (x = 1; x < bil-1 ; x++) {
                if( bil > 2){
                    ketiga = tampsatu + tampdua;
                    System.out.print(", " +ketiga);
                    tampsatu = tampdua;
                    tampdua = ketiga;
                }
            }
            System.out.print("\nIngin Lanjut lagi ? ");
            jawab = input.next().charAt(0);
        }
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac Fibonacci.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java Fibonacci
Ingin Menginputkan Berapa Deret Fibonacci : 3
1, 1, 2
Ingin Lanjut lagi ? y
Ingin Menginputkan Berapa Deret Fibonacci : 8
1, 1, 2, 3, 5, 8, 13, 21
Ingin Lanjut lagi ? y
Ingin Menginputkan Berapa Deret Fibonacci : 10
1, 1, 2, 3, 5, 8, 13, 21, 34, 55
Ingin Lanjut lagi ? t
```

- c. Analisa : pada program diatas, terdapat percabangan if else, dan juga terdapat looping for, percabangannya digunakan untuk menampilkan bilangan pertama dan kedua, untuk perulangannya digunakan untuk menampilkan bilangan berikutnya di ikuti dengan perhitungan bilangannya, saya menggunakan variable tambahan 2, untuk menampung bilangan yang ada, agar bilangan tsb tidak berganti.

4. Buatlah program untuk menampilkan deret bilangan genap dari 2 sampai dengan 20 kecuali kelipatan 6. Contoh tampilan : 2 4 8 10 14 16 20

a. Listing program

```
public class BilGenap {
    public static void main(String [] args){
        int i;
        for(i = 2; i <= 20; i++){
            if(i%2 == 0){
                if(i%6 == 0){
                    continue;
                }
                System.out.print(i+ " , ");
            }
        }
    }
}
```

b. Output

```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac BilGenap.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java BilGenap
2, 4, 8, 10, 14, 16, 20,
```

- c. Analisa : pada program diatas, menggunakan perulangan for dikarenakan tau kapan akan berhentinya. Dan juga menggunakan pengecekan if, apakah bilangan tersebut bila dibagi dengan 2 sisa baginya 0, jika iya akan di cek ckembali, jika dibagi 6 hasil baginya 0 akan di lewati atau continue. Dikarenakan pada perintah kelipatan 6 itu di skip. Sehingga output yang muncul hanyalah bilangan genap kecuali kelipatan 6, maks sampai bilangan 20.

Tugas

1. Menghitung nilai determinan dan mencari akar persamaan kuadrat

Buatlah program untuk menghitung determinan dan mencari akar-akar dari persamaan kuadrat : $ax^2 + bx + c = 0$, dengan ketentuan sebagai berikut :

$$D = b^2 - 4ac$$

Jika $D = 0$, maka terdapat 2 akar real yang kembar, yaitu : $x_1 = x_2 = -b / 2a$

Jika $D > 0$, maka terdapat 2 akar real yang berlainan, yaitu : $x_1 = (-b + \sqrt{D}) / 2a$ $x_2 = (-b - \sqrt{D}) / 2a$

Jika $D < 0$, maka terdapat 2 akar imajiner yang berlainan, yaitu : $x_1 = -b / 2a + (\sqrt{D} / 2a) i$
 $x_2 = -b / 2a - (\sqrt{D} / 2a) i$

Input : a, b, c (int)

Output : Nilai Determinan serta nilai akar-akar persamaan tsb (x_1 dan x_2).

Petunjuk : Gunakan `Math.pow(x,0.5)` untuk mencari akar dari x.

a. Listing Program

```
import java.util.*;

public class Determinan {
    public static void main(String[] args) {
        int a, b, c;
        Float D, x1, x2;
        char jawab = 'y';
        Scanner input = new Scanner(System.in);

        while (jawab == 'y' || jawab == 'Y') {
            System.out.print("Masukkan a : ");
            a = input.nextInt();
            System.out.print("Masukkan b : ");
            b = input.nextInt();
            System.out.print("Masukkan c : ");
            c = input.nextInt();

            D = (float) Math.pow(b, 2)-4*a*c;

            if (D == 0) {
                x1 = x2 = (float)-b / (2 * a);
                System.out.println("Nilai Determinannya : " + D + " nil
ai X1 = X2 = " + x1);
            } else if (D < 0) {
                x1 = (-b / (2 * a) + (float) Math.pow(D,0.5) / 2*a) ;
                x2 = (-b / (2 * a) - (float) Math.pow(D,0.5) / 2*a);
                System.out.println("Nilai Determinannya : " + D + " nil
ai X1 = " + x1 + " i dan nilai X2 = " + x2 + " i");
            } else{
                x1 = (-b + (float) Math.pow(D,0.5)) / (2 * a);
                x2 = (-b - (float) Math.pow(D,0.5)) / (2 * a);
                System.out.println("Nilai Determinannya : " + D + " nil
ai X1 = " + x1 + " nilai X2 = " + x2);
            }
            System.out.print("Ingin Lanjut lagi ? ");
            jawab = input.next().charAt(0);
        }
    }
}
```

b. Output

```
Ingin Lanjut lagi ? c
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac Determinan.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java Determinan
Masukkan a : 1
Masukkan b : 2
Masukkan c : 1
Nilai Determinannya : 0.0 nilai X1 = X2 = -1.0
```

- c. Analisis : pada program diatas, menghitung nilai determinan beserta akar x1 dan x2nya, menggunakan pengecekan if else, apabila hasil D itu > 0 atau < 0 atau == 0. Jika memenuhi syarat maka statement yang ada dibawahnya akan dijalankan. Kemudian akan ditampilkan berapa hasilnya

2. Menentukan tahun kabisat

Buatlah program untuk menentukan suatu tahun kabisat atau bukan dimana tahun dibatasi mulai dari tahun 1900 sampai dengan tahun 2005.

Contoh tampilan:

Masukkan tahun (1900-2005) : 1923

1923 bukan tahun kabisat

Masukkan tahun (1900-2005) : 1898

Maaf, tahun input di bawah 1900

Masukkan tahun (1900-2005) : 1996

1996 adalah tahun kabisat

Masukkan tahun (1900-2005) : 2008

Maaf, tahun input diatas 2005

a. Listing program

```
import java.util.*;

public class Kabisat {
    public static void main(String [] args){
        int tahun;
        char jawab = 'y';
        Scanner Input = new Scanner(System.in);

        while(jawab == 'y' || jawab == 'Y'){
            System.out.print("Masukkan tahun (1900-2005) : ");
            tahun = Input.nextInt();

            if(tahun >= 1900 && tahun <=2005){
                if((tahun % 400 == 0 || tahun % 100 != 0) && tahun % 4 ==
0){
                    System.out.println("Tahun " +tahun+ " merupakan tahun
kabisat");
                }else{
                    System.out.println("Tahun " +tahun+ " Bukan Tahun Kabi
sat");
                }
            }else if(tahun > 2005){
                System.out.println("Inputan Anda salah, diatas 2005");
            }else{
                System.out.println("Inputan Anda salah, dibawah 1900");
            }
            System.out.print("Ingin Lanjut lagi ? ");
            jawab = Input.next().charAt(0);
        }
    }
}
```

b. Output


```
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> javac Kabisat.java
PS D:\1. KIKO\1. KULIAH\3. SEMESTER 3\1. PBO\2. PRAKTIKUM\3. CODINGAN\4.Prak2.2> java Kabisat
Masukkan tahun (1900-2005) : 1923
Tahun 1923 Bukan Tahun Kabisat
Ingin Lanjut lagi ? y
Masukkan tahun (1900-2005) : 1898
Inputan Anda salah, dibawah 1900
Ingin Lanjut lagi ? y
Masukkan tahun (1900-2005) : 1996
Tahun 1996 merupakan tahun kabisat
Ingin Lanjut lagi ? y
Masukkan tahun (1900-2005) : 2008
Inputan Anda salah, diatas 2005
Ingin Lanjut lagi ? t
```

- c. Analisis : pada program diatas, terdapat pengecekan tahun kabisat. Untuk kriteria yang pertama yaitu tahun harus lebih dari 1900 dan kurang dari 2005. Menggunakan if else, sebagai bantuannya. Kemudian jika tahun tersebut lolos pada awal pengecekan kemudian akan di cek Kembali. Jika tidak lolos maka tahun tersebut bukanlah tahun kabisat, jika lolos, merupakan tahun kabisat.