**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](https://id.wikipedia.org/wiki/Komputer" \o "Komputer).[[1]](https://id.wikipedia.org/wiki/Pemrograman#cite_note-1) Kode ini ditulis dalam berbagai [bahasa pemrograman](https://id.wikipedia.org/wiki/Bahasa_pemrograman" \o "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](https://id.wikipedia.org/wiki/Algoritme" \o "Algoritme), [logika](https://id.wikipedia.org/wiki/Logika" \o "Logika), [bahasa pemrograman](https://id.wikipedia.org/wiki/Bahasa_pemrograman" \o "Bahasa pemrograman), dan pada banyak kasus, pengetahuan-pengetahuan lain seperti [matematika](https://id.wikipedia.org/wiki/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 operan sehingga membentuk ekspresi.

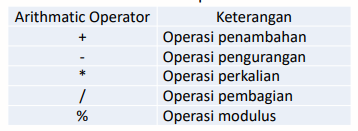
Operator dapat diklasifikasikan menjadi 2 bentuk, yaitu unary operator dan binary operator. Unary operator adalah operator yang hanya melibatkan 1 operan. Sedangkan binary operator adalah operator yang melibatkan 2 operan. Java mempunyai berbagai macam jenis operator yang dapat digolongkan menjadi operator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

**Macam Operator**

Di dalam pemrograman java terdapat 7 operator. Antara lain :

1. Aritmatic Operator

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



1. Increment-Decrement Operator

ncrement – 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 penaikkan dan penurunan nilainya, incrementdecrement operator ini dapat diklasifikasikan menjadi 2 macam, yaitu pre-increment/decrement dan post-increment/decrement

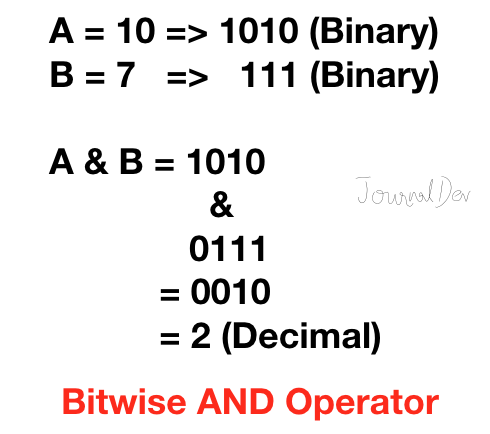




1. BitWise Operator

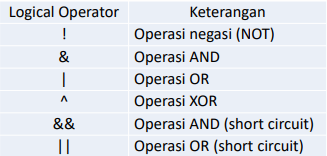
Bitwise operator adalah operator yang dipakai untuk operasi bit pada nilai operan. Yang termasuk bitwise operator bisa dilihat pada Tabel.





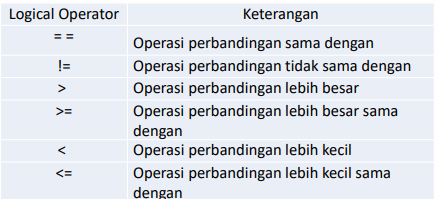
1. Boolean Operator

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



1. Logical Operator

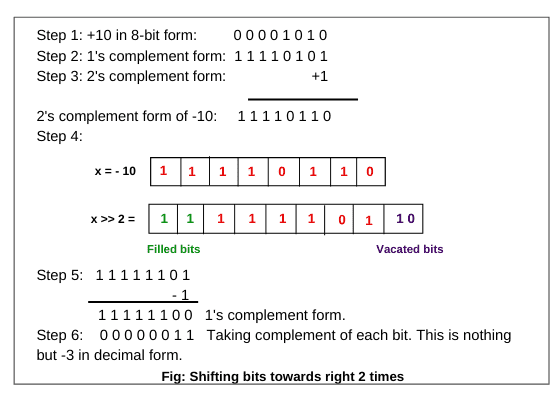
Operator !, &, | dan ^ mempunyai implementasi yang sama sebagaimana ketika ia menjadi bitwise operator. Hanya saja di logical operator, operan yang dilibatkan disini harus bertipe boolean, yang hanya mempunyai nilai true atau false. Logical operator (operator logika) adalah operator yang sering dipakai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



1. 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.





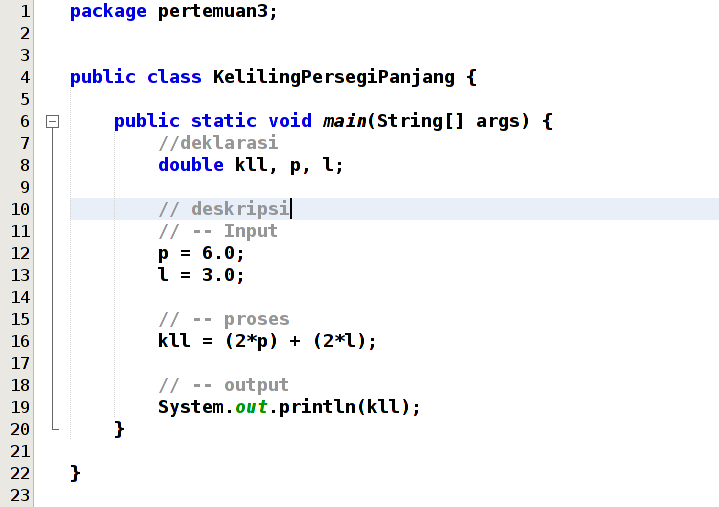
1. Combination Operator

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 .



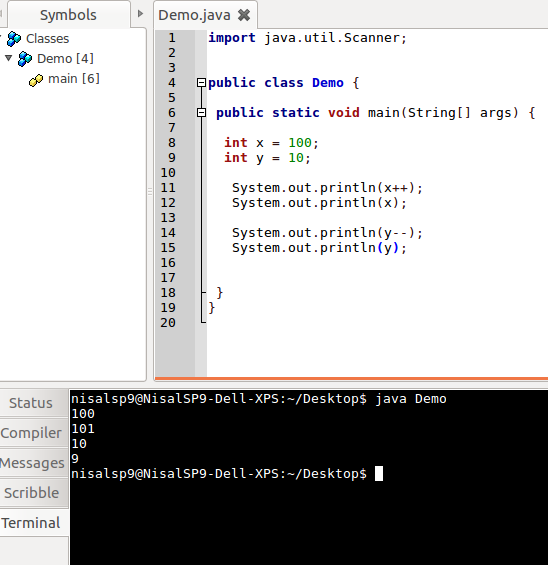
**Contoh Operator**

1. Operator Aritmatika



Outputnya : **18.0**

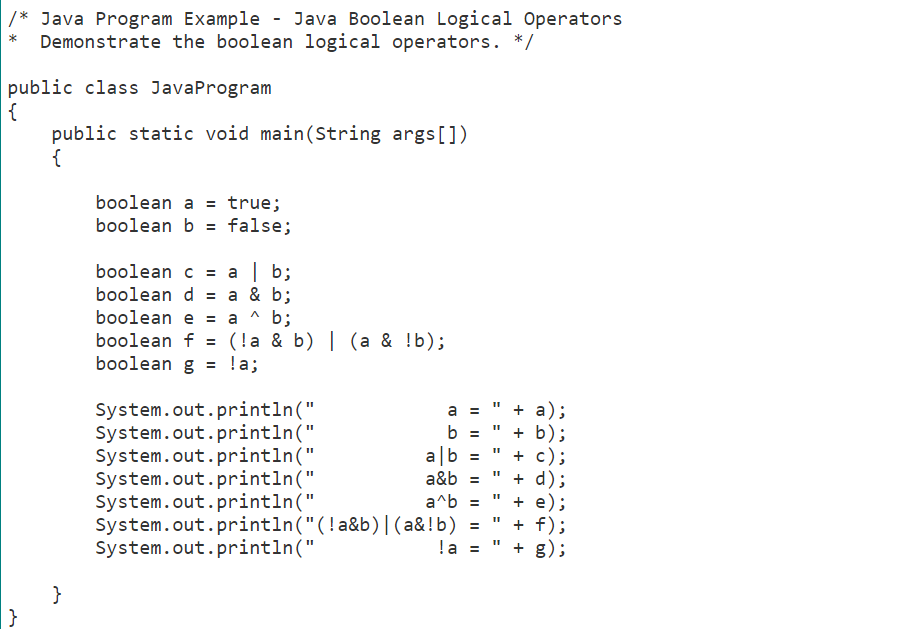
1. Operator Increment-Decrement

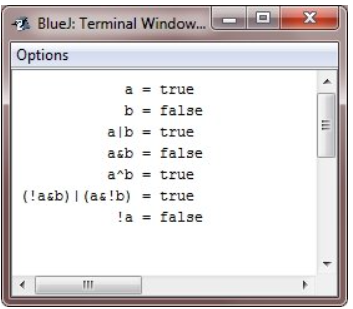


1. Bitwise Operator

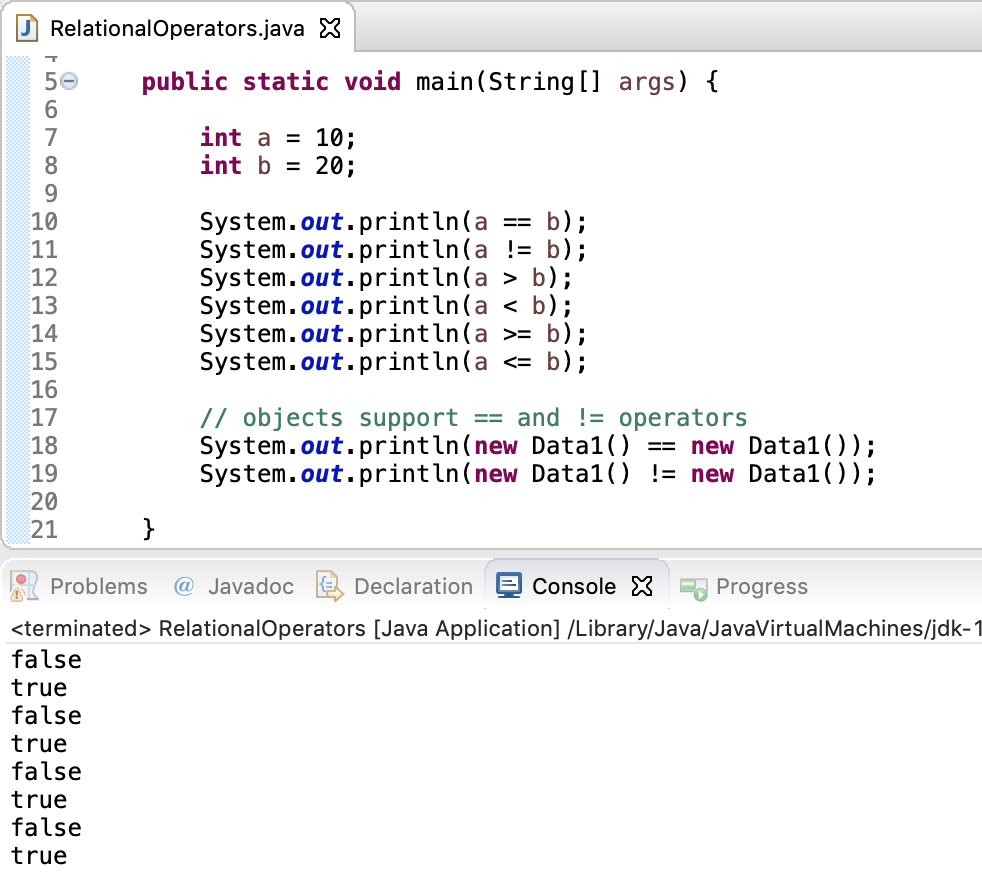


1. Boolean Operator

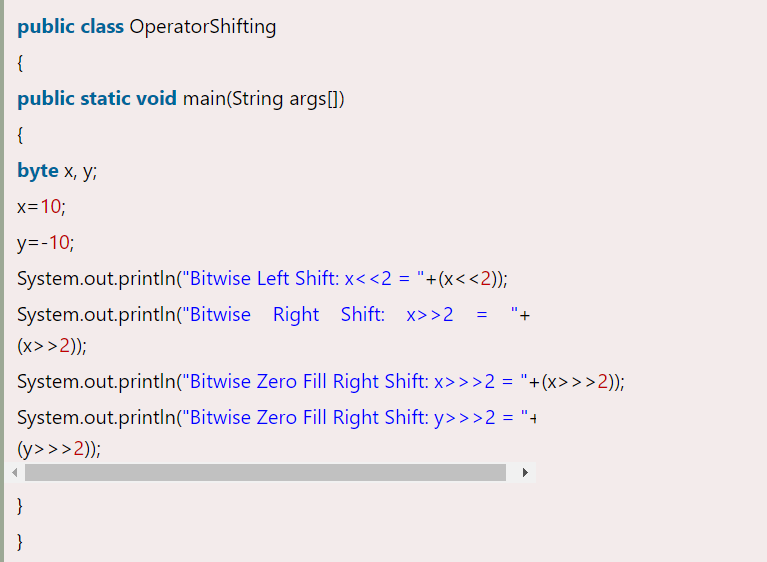


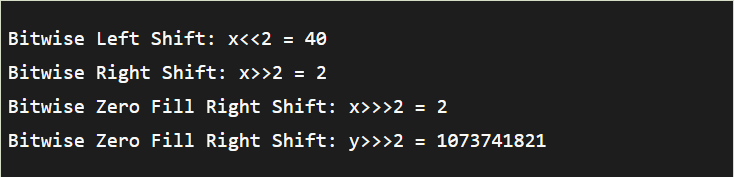


1. Logika operator

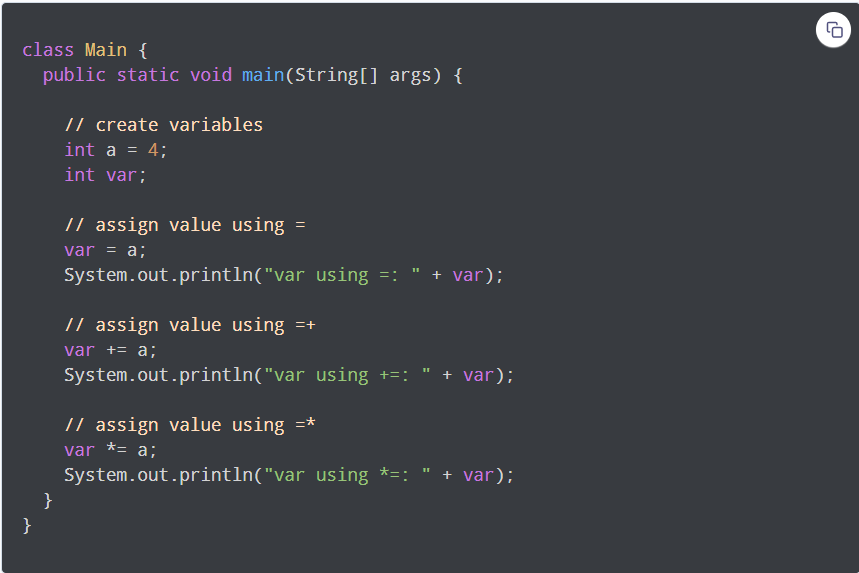


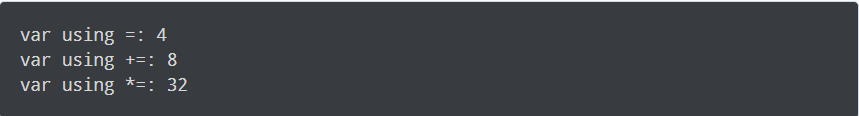
1. Shift operator





1. Combination Operator





**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%2Fjavaseeeedu.blogspot.com%2F2017%2F10%2Fjava-postfix-operators-increment.html&psig=AOvVaw0wUQtVvMTuEPKKQTBwVDsa&ust=1631091294233000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCODhiai_7PICFQAAAAAdAAAAABAD>

Percobaan 1

1. 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++);

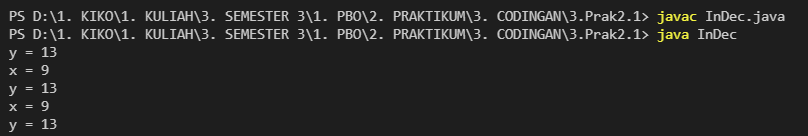
        System.out.println("x = " + x--);

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

    }

}

1. Output



1. 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

1. 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));

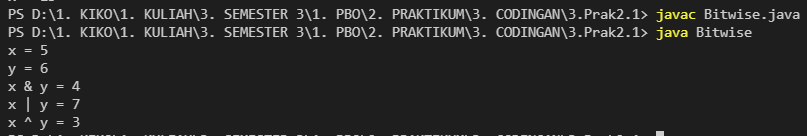
        System.out.println("x | y = " + (x | y));

        System.out.println("x ^ y = " + (x ^ y));

    }

}

1. Output



1. Analisa : pada program diatas terdapat operator bitwise yaitu operator untuk menangani operasi logika bilangan binner 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

1. Listing Program

class BitiseComplement {

    public static void main(String [] args) {

        int x = 8;

        System.out.println("x = " + x);

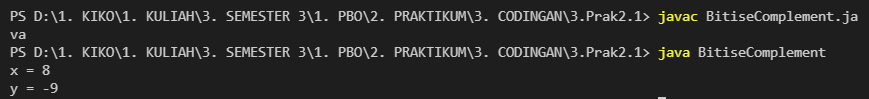
        int y = ~x;

        System.out.println("y = " + y);

    }

}

1. Output



1. 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

1. 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));

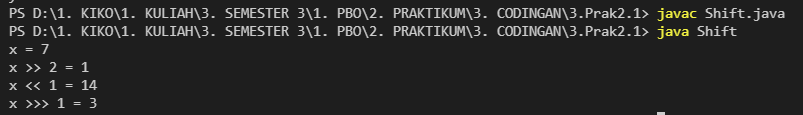
        System.out.println("x << 1 = " + (x << 1));

        System.out.println("x >>> 1 = " + (x >>> 1));

        }

}

1. Output



1. 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

1. 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));

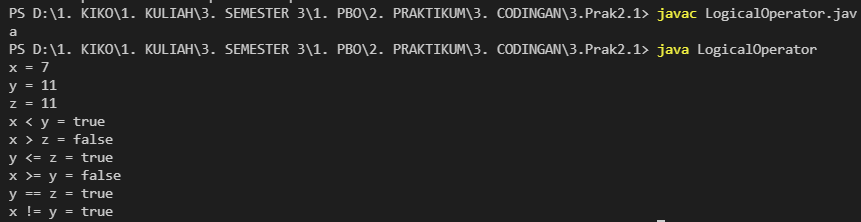
        System.out.println("y == z = " + (y == z));

        System.out.println("x != y = " + (x != z));

    }

}

1. Output



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

Percobaan 6

1. Listing Program

public class BooleanAnd {

    public static void main(String [] args) {

        int a = 5, b = 7;

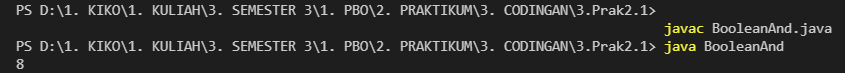
        if ((a < 2) & (b++ < 10)) b += 2;

        System.out.println(b);

    }

}

1. Output



1. 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

1. 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);

    }

}

1. Output



1. 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

1. 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);

    }

}

1. Output



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

Percobaan 9

1. 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);

    }

}

1. Output



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

Percobaan 10

1. Listing Program

class Conditional {

    public static void main(String [] args) {

        int x = 0;

        boolean isEven = false;

        System.out.println("x = " + x);

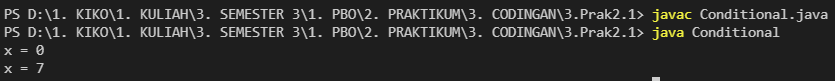
        x = isEven ? 4 : 7;

        System.out.println("x = " + x);

    }

}

1. Output



1. 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

Tugas

1. Menampilkan representasi biner dari bilangan desimal bertipe int
2. Listing program

import java.util.\*;

public class IntToBiner{

    public static void main(String [] args){

        Scanner input = new Scanner (System.in);

        int bil;

        System.out.print("Masukkan nilai desimal: ");

        bil = input.nextInt();

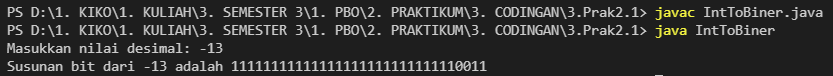
        String biner = Integer.toBinaryString(bil);

        System.out.println("Susunan bit dari "+bil+" adalah %032d" + biner);

    }

}

1. Output



1. Analisa : pada program diatas, terdapat program yang mengkonversikan bilangan decimal ke biner, sehingga hasil diatas merupakan outputnya.
2. Menampilkan representasi biner dari bilangan desimal bertipe byte
3. Listing program

import java.util.\*;

public class ByteToBiner {

    public static void main(String [] args) {

        Scanner input = new Scanner(System.in);

        byte bil;

        System.out.print("Masukkan nilai desimal: ");

        bil = input.nextByte();

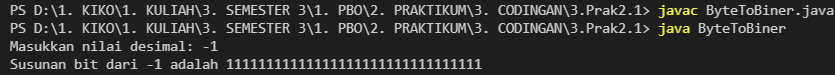
        String biner = Integer.toBinaryString(bil);

        System.out.println("Susunan bit dari " + bil + " adalah " + biner);

    }

}

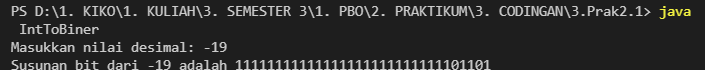
1. Output



1. Analisa : pada program diatas, terdapat program yang mengkonversikan bilangan decimal ke biner, sehingga hasil diatas merupakan outputnya.

Tugas

1. Mencari representasi biner dari suatu bilangan. Tuliskan representasi bit dari nilai –19? Jelaskan.



-19

00000010011 = 19 merupakan bilangan awalnya, kemudian dikurangi 1 bit

00000000001 (dikurangi 1 bit)

00000010010 hasilnya, kemudian 1 di ganti menjadi 0 dan sebaliknya

11111101101 = -19

* Analisa : untuk mengubah mengkonversi bilangan diatas, yaitu dengan mengurangi dengan 1 bit diperoleh 00000010010. Kemudian di convert 0 ke 1 dan sebaliknya, menghasilkan nilai 11111101101.

1. Menganalisa pergeseran bit dari operasi shift Jelaskan apa yang terjadi pada potongan program berikut ini:

byte a=-1;

a=(byte) (a >>> 2);

konversi 1 menjadi -1:

00000001 = 1

00000001 (dikurangi 1 bit)

00000000 = hasil

11111111 = konversi dari 0 ke 1.

Kemudian digeser >>> 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Original Data (-1 desimal) | 11111111 | | | |
| Promote to int gives | 11111111 | 11111111 | 11111111 | 11111111 |
| Shift right unsigned 2 bits gives | 00111111 | 11111111 | 11111111 | 11111111 |
| Truncate to byte gives |  |  |  | 11111111 |
| Expected result was |  |  |  | 00111111 |

* Analisa : seperti nomor yang sebelumnya, bilangan dikonversikan terlebih dahulu lalu mendapatkan hasil 1111111111. Kemudian membuat table yang didalamnya terdapat beberapa step. Kemudian geser ke kanan 2 bit, kemudian pada hasil akhir menggunakan binary paling depan yaitu 00111111

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 persayaratan 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 denga 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;*

*}*

1. 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 jumlah nya. 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

1. 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!");

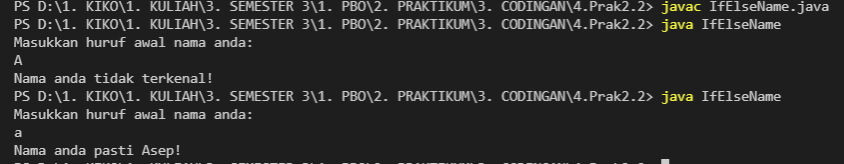
        else

            System.out.println("Nama anda tidak terkenal!");

    }

}

1. Output



1. 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

1. 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:

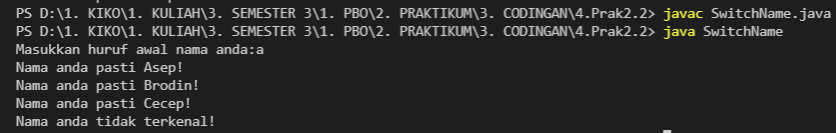
                System.out.println("Nama anda tidak terkenal!");

        }

    }

}

1. Output



1. 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

1. 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:

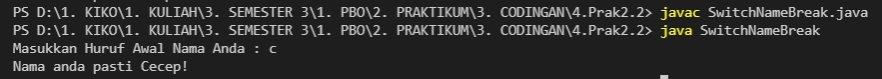
                System.out.println("Nama anda tidak terkenal!");

        }

    }

}

1. Output



1. Analisa : pada program diatas, merupakan progam 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 mncul yang berada dalam lingkup case tsb.

Percobaan 4

1. 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++;

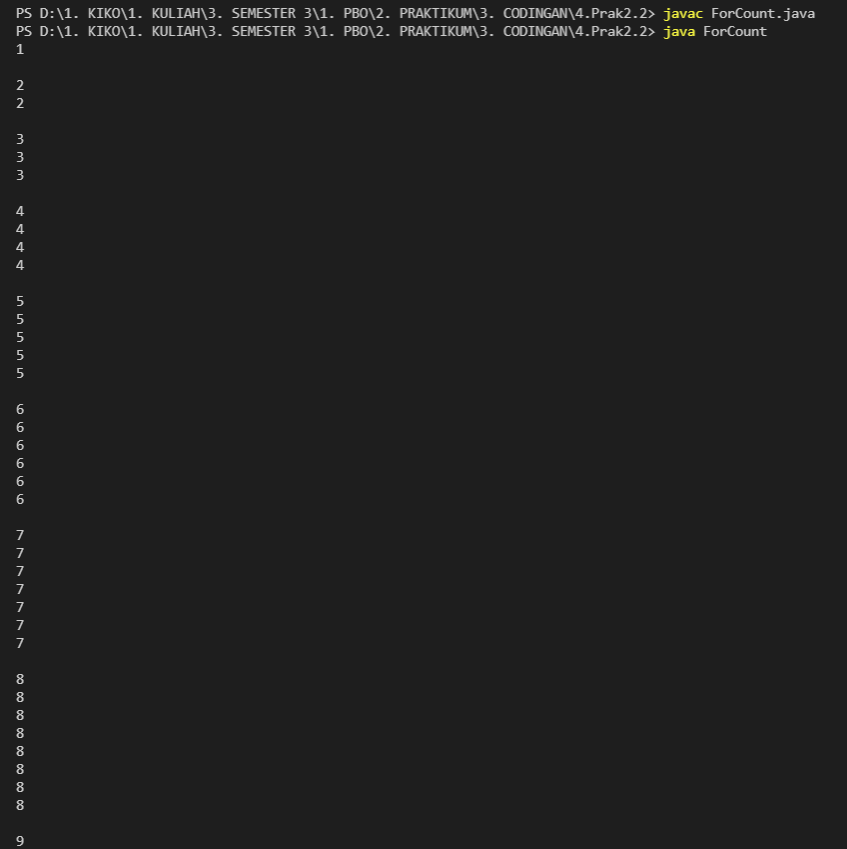
            System.out.println();

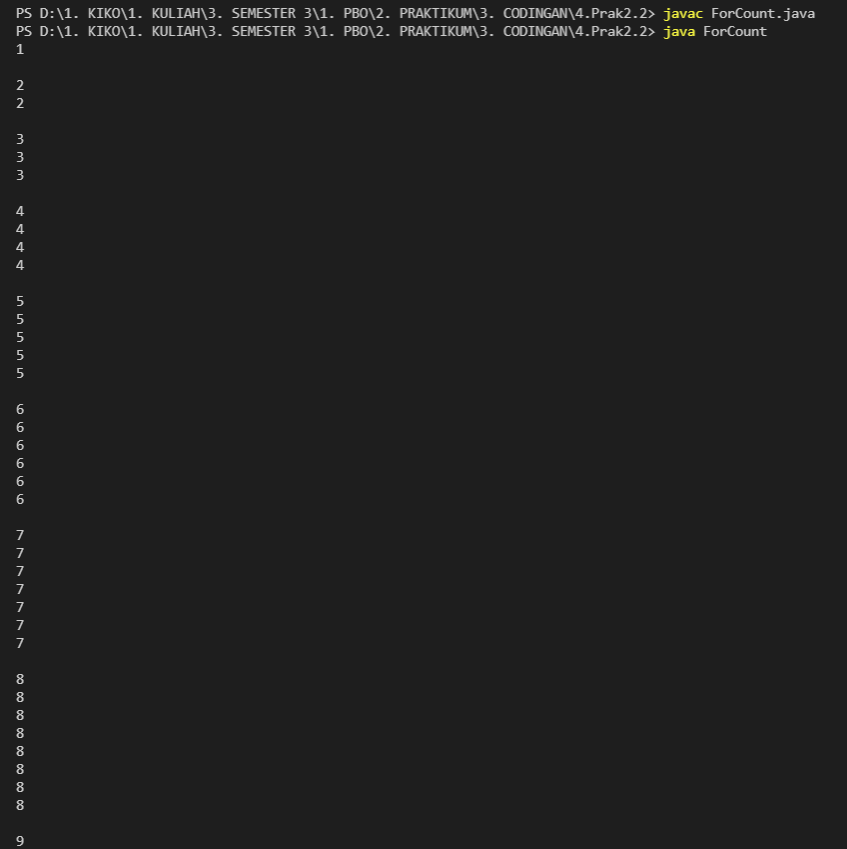
        }

    }

}

1. Output







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

Percobaan 5

1. 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();

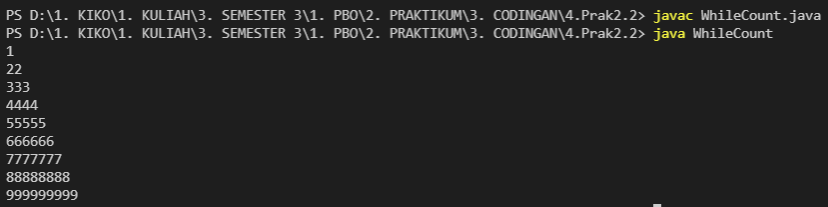
            i++;

        }

    }

}

1. Output



1. Analisa : pada program diatas, terdapat 2 perulangan kedua-duanya menggunakan while, yang didalamnya terdapat kondisi. Jika i < 0 kemudian j < i +1, kemudian hasilnya dicetak. Terdapat 2 increment. Yaitu count dan i

Percobaan 6

1. Listing program

class BreakLoop {

    public static void main(String [] args) {

        int i = 0;

        do {

            System.out.println("Iterasi ke " + i);

            i++;

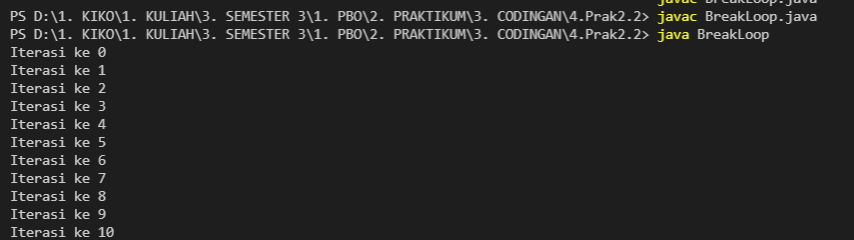
            if (i > 10) break;

        } while (true);

    }

}

1. Output



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

Percobaan 7

1. 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;

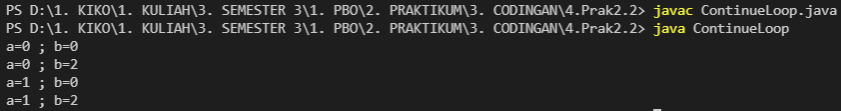
                System.out.println("a=" + a + " ; b=" + b);

            }

    }

}

1. Output



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

Percobaan 8

1. 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);

            }

    }

}

1. Output



1. 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

1. 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;

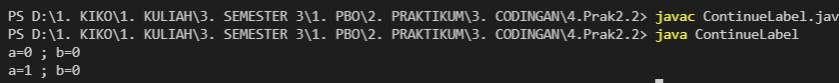
                System.out.println("a=" + a + " ; b=" + b);

            }

    }

}

1. Output



1. 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).
2. 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 ? ");

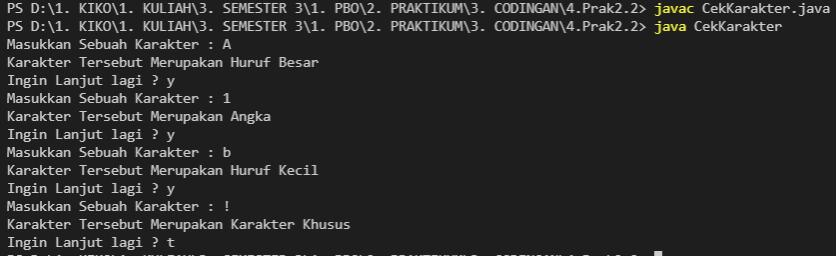
            jawab = input.next().charAt(0);

        }

    }

}

1. Output



1. 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 x 2 x 3 x 4 x 5 =120. Buatlah program untuk menampilkan tabel hasil faktorial dari suatu bilangan yang diinputkan (tampilan bilangan rata kanan)
3. 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\tn!\n");

            System.out.println("================================\n");

            while(x <= bil){

                hasil \*= x;

                System.out.println(+ x + "\t" +hasil);

                x++;

            }

            System.out.print("Ingin Lanjut lagi ? ");

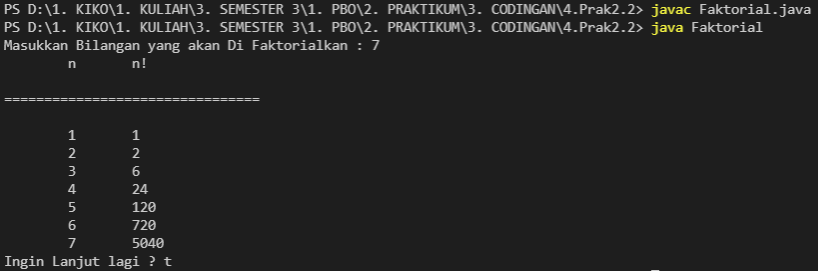
            jawab = input.next().charAt(0);

        }

    }

}

1. Output



1. Analisa : pada program diatas, menggunakan perulangan while keduanya yaitu untuk perulangan yang terdapat kondisi apakah x <= 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 menjalan kan progam lagi atau kah tidak,
2. Buatlah program untuk menampilkan deret Fibonacci Contoh tampilan : Masukkan berapa deret Fibonacci? 8

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

1. 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 ? ");

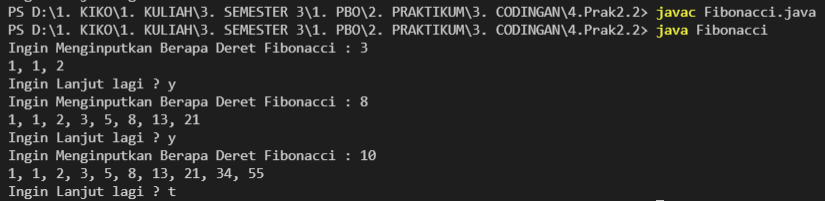
            jawab = input.next().charAt(0);

        }

    }

}

1. Output



1. 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.
2. Buatlah program untuk menampilkan deret bilangan genap dari 2 sampai dengan 20 kecuali kelipatan 6. Contoh tampilan : 2 4 8 10 14 16 20
3. 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+ ", ");

            }

        }

    }

}

1. Output



1. 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 : ax2 + bx + c = 0 , dengan ketentuan sebagai berikut :

D = b2 - 4ac

Jika D = 0 , maka terdapat 2 akar real yang kembar, yaitu : x1 = x2 = -b / 2a

Jika D > 0 , maka terdapat 2 akar real yang berlainan, yaitu : x1 = (-b + √D) / 2a x2 = (-b - √D) / 2a

Jika D < 0 , maka terdapat 2 akar imaginer yang berlainan, yaitu : x1 = -b / 2a + (√D / 2a) i x2 = -b / 2a - (√D / 2a) i

Input : a, b, c (int)

Output : Nilai Determinan serta nilai akar-akar persamaan tsb (x1 dan x2).

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

* 1. 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 + " nilai 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 + " nilai 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 + " nilai X1 = " + x1 + " nilai X2 = " + x2);

            }

            System.out.print("Ingin Lanjut lagi ? ");

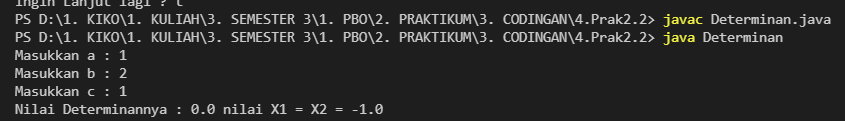
            jawab = input.next().charAt(0);

        }

    }

}

* 1. Output



* 1. 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

1. 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

1. 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 Kabisat");

                }

            }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);

        }

    }

}

1. Output



1. 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 hika 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.