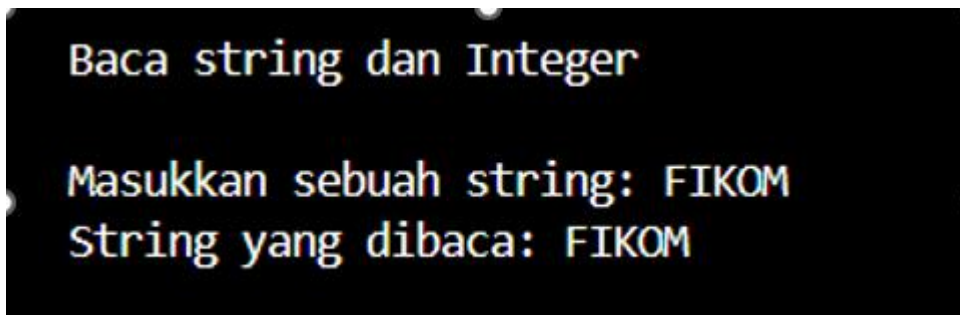


## 1. BacaString

```
2. import java.io.BufferedReader;
3. import java.io.IOException;
4. import java.io.InputStreamReader;
5.
6. public class BacaString {
7.
8.     public static void main(String[] args) throws IOException {
9.         String str;
10.        BufferedReader dataIn = new BufferedReader(new
    InputStreamReader(System.in));
11.
12.        System.out.println("\nBaca string dan Integer \n");
13.        System.out.print("Masukkan sebuah string: ");
14.        str = dataIn.readLine();
15.        System.out.println("String yang dibaca: " + str);
16.    }
17.}
18.
```



```
Baca string dan Integer
Masukkan sebuah string: FIKOM
String yang dibaca: FIKOM
```

**Import Statement:** Program mengimpor kelas-kelas yang digunakan untuk membaca input pengguna. Dalam hal ini, `BufferedReader` digunakan untuk membaca input dari `InputStreamReader`, yang mendapatkan input dari `System.in`.

**Deklarasi Kelas:** Program mendeklarasikan `BacaString` sebagai kelas.

**Metode Utama (Main):** Di sini metode utama `main` dimulai.

**Deklarasi Variabel:** Variabel `str` dibuat untuk menyimpan string yang akan dimasukkan pengguna.

**Membuat Objek `BufferedReader`:** Sebuah konstruktor digunakan untuk membuat objek `BufferedReader`, yang mengambil objek `InputStreamReader` sebagai argumennya.

`InputStreamReader` membaca catatan dari `System.in`, yang merupakan catatan yang berasal dari keyboard.

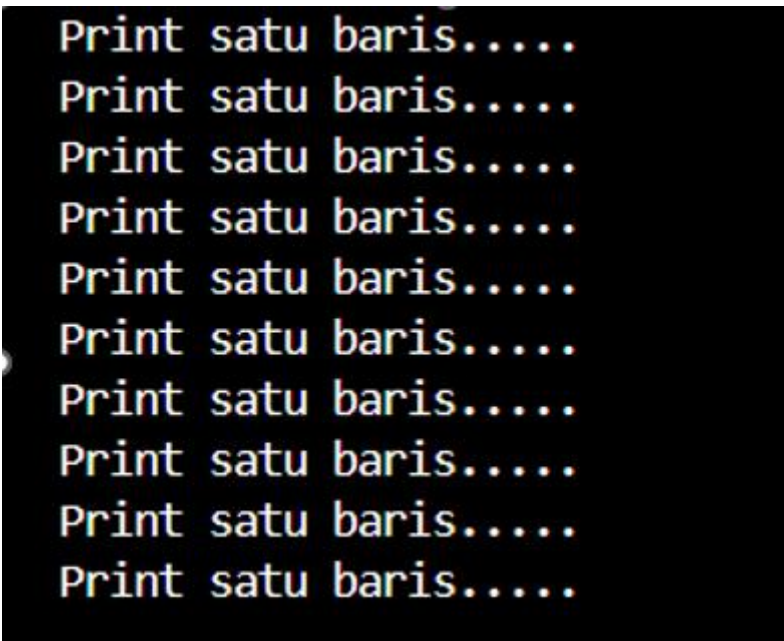
**Membaca Input:** Pesan "Masukkan string:" dicetak untuk meminta pengguna memasukkan string. Kemudian, `readLine()` digunakan untuk membaca input pengguna dan menyimpannya dalam variabel `str`.

**Pencetakan input:** String yang dibaca dicetak kembali ke layar dengan pesan "String yang dibaca:".

**Handling Exception:** Karena `readLine()` dapat menghasilkan `IOException`, deklarasi yang mengeluarkan `IOException` ditambahkan ke metode `main` untuk menangani kesalahan yang mungkin terjadi saat membaca input pengguna. Program ini menunjukkan cara mudah untuk menggunakan `BufferedReader` untuk membaca input pengguna dan mencetaknya kembali.

## 2. ForEver

```
public class ForEver {  
    public static void main(String[] args) {  
        System.out.println("Program akan looping, akhiri dengan ^c");  
        while (true) {  
            System.out.println("Print satu baris.....");  
        }  
    }  
}
```



```
Print satu baris.....  
Print satu baris.....  
Print satu baris.....  
Print satu baris.....  
Print satu baris.....  
Print satu baris.....  
Print satu baris.....  
Print satu baris.....  
Print satu baris.....  
Print satu baris.....
```

contoh sederhana dari cara kita dapat menggunakan konstruksi perulangan while dalam Java untuk membuat loop tak terbatas. Berikut adalah penjelasan singkat dari setiap komponen program:

Deklarasi Kelas: Deklarasi kelas ForEver adalah langkah awal dari program.

Metode Utama (Main): Di sini metode utama main dimulai.

Mencetak Pesan: Sistem.out.println digunakan untuk mencetak pesan "Program akan looping, akhiri dengan ^c", yang memberi tahu pengguna bahwa program akan terus melakukan looping.

Perulangan While: Perulangan while dapat digunakan ketika kondisi while (true) digunakan. Ini menunjukkan bahwa blok kode yang terkandung di dalamnya akan terus dieksekusi selama kondisi benar. Ini karena perulangan tidak pernah berubah karena nilai benar tidak pernah berubah.

Mencetak Pesan dalam Loop: Pesan "Print satu baris....." dicetak secara konsisten dalam perulangan while menggunakan System.out.println().

Loop Tak Terbatas: Karena kondisi perulangan selalu bernilai benar, program akan terus mencetak pesan "Print satu baris." tanpa henti.

Akhiri Program: Untuk mengakhiri program, tekan kombinasi tombol Ctrl dan C pada keyboard. Ini akan memberhentikan eksekusi program yang sedang berjalan.

Program ini digunakan untuk demonstrasi atau debugging untuk melihat bagaimana loop tak terbatas berfungsi atau untuk menguji perilaku program saat melakukan looping tak terbatas. Namun, dalam pengembangan perangkat lunak yang sebenarnya, loop tak terbatas seperti ini harus dihindari karena dapat menyebabkan program gagal atau membuatnya tidak responsif.

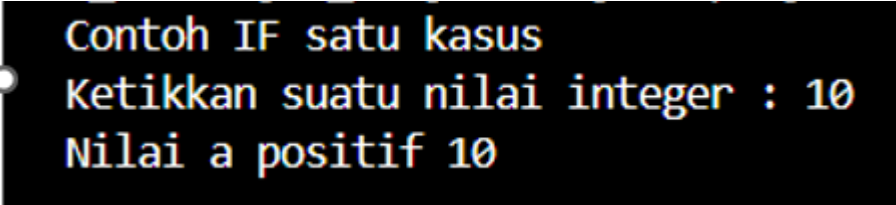
### 3. If1

```
import java.util.Scanner;

public class If1 {

    public static void main(String[] args) {
        Scanner masukan = new Scanner(System.in);
        int a;

        System.out.println("Contoh IF satu kasus");
        System.out.print("Ketikkan suatu nilai integer : ");
        a = masukan.nextInt();
        if (a >= 0) {
            System.out.println("Nilai a positif " + a);
        }
    }
}
```



```
Contoh IF satu kasus
Ketikkan suatu nilai integer : 10
Nilai a positif 10
```

**Import Statement:** Program mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input pengguna.

**Deklarasi Kelas:** Deklarasi kelas If1 digunakan untuk memulai program.

**Metode Utama (Main):** Di sini metode utama main dimulai.

**Deklarasi Variabel:** Variabel masukan adalah objek kelas Scanner yang digunakan untuk membaca input pengguna. Variabel a adalah bilangan bulat yang akan menyimpan nilai yang dimasukkan oleh pengguna.

**Mencetak Pesan:** Sistem.out.println() mencetak pesan "Contoh IF satu kasus" untuk memberi tahu pengguna tentang program yang akan dijalankan, dan pesan "Ketikkan suatu nilai integer:" dicetak untuk meminta pengguna memasukkan nilai integer.

Nilai integer yang dimasukkan oleh pengguna dapat dibaca dan disimpan dalam variabel a dengan menggunakan metode nextInt() dari objek masukan.

**Struktur Pengkondisian If:** Struktur ini memungkinkan pengguna untuk memeriksa nilai yang mereka masukkan. Jika nilai a lebih besar atau sama dengan 0, maka blok kode yang disebutkan dalam if akan dieksekusi.

**Mencetak Output:** Jika kondisi if terpenuhi, yaitu jika nilai a tidak negatif, pesan "Nilai a positif" akan dicetak bersama dengan nilai a menggunakan System.out.println().

Hanya jika pengguna memasukkan nilai yang tidak negatif (0 atau lebih besar dari 0), program ini akan mencetak pesan "Nilai a positif". Jika pengguna memasukkan nilai yang negatif, pesan tidak akan dicetak.

#### 4. If2

```
import java.util.Scanner;

public class If2 {

    public static void main(String[] args) {
        int a;
        Scanner masukan = new Scanner(System.in);

        System.out.println("Contoh IF dua kasus");
        System.out.print("Ketikkan suatu nilai integer: ");
        a = masukan.nextInt();

        if (a >= 0) {
            System.out.println("Nilai a positif " + a);
        } else {
            System.out.println("Nilai a negatif " + a);
        }
    }
}
```

```
Contoh IF dua kasus
Ketikkan suatu nilai integer: -10
Nilai a negatif -10
```

```
Contoh IF dua kasus
Ketikkan suatu nilai integer: 10
Nilai a positif 10
```

Program ini menunjukkan bagaimana struktur pengkondisian if-else digunakan dalam Java. Itu meminta pengguna memasukkan nilai integer dan kemudian menentukan apakah nilai itu positif atau negatif.

Program akan mencetak "nilai positif" bersama nilai jika nilainya lebih besar atau sama dengan 0. Jika nilainya kurang dari 0, maka program akan mencetak "nilai negatif" bersama nilainya.

### 5. If3

```
import java.util.Scanner;

public class If3 {

    public static void main(String[] args) {
        int a;
        Scanner masukan = new Scanner(System.in);

        System.out.println("Contoh IF tiga kasus");
        System.out.print("Ketikkan suatu nilai integer: ");
        a = masukan.nextInt();

        if (a > 0) {
            System.out.println("Nilai a positif " + a);
        } else if (a == 0) {
            System.out.println("Nilai Nol " + a);
        } else {
            System.out.println("Nilai a negatif " + a);
        }
    }
}
```

```
Contoh IF tiga kasus
Ketikkan suatu nilai integer: 20
Nilai a positif 20
```

```
Contoh IF tiga kasus
Ketikkan suatu nilai integer: 0
Nilai Nol 0
```

```
Contoh IF tiga kasus
Ketikkan suatu nilai integer: -20
Nilai a negatif -20
```

penggunaan struktur pengkondisian if-else if-else dalam Java. Aplikasi ini meminta pengguna memasukkan nilai integer dan kemudian memeriksa tiga kondisi potensial berdasarkan nilai tersebut:

Program akan mencetak "nilai a positif" dengan nilai yang dimasukkan jika nilainya lebih besar dari 0, "nilai nol" dengan nilai yang dimasukkan, dan "nilai a negatif" dengan nilai yang dimasukkan jika nilainya sama dengan 0.

Oleh karena itu, berdasarkan nilai yang dimasukkan pengguna, program ini memberikan tiga hasil yang berbeda: nilai positif, nilai nol, atau nilai negatif.

## 6. KasusBoolean

```
public class KasusBoolean {  
  
    public static void main(String[] args) {  
  
        boolean bool;  
  
        bool = true;  
        if (bool) {  
            System.out.print("true\n");  
        } else {  
            System.out.print("false\n");  
        }  
        if (!bool) {  
            System.out.print("salah\n");  
        } else {  
            System.out.print("benar\n");  
        }  
    }  
}
```

```
true  
benar
```

penggunaan struktur pengkondisian if-else dan variabel boolean dalam bahasa pemrograman Java. Struktur pengkondisian if digunakan untuk melakukan pengecekan apakah nilai variabel boolean benar atau tidak; jika nilai bool benar, program akan mencetak "true", jika tidak, maka program akan mencetak "false".

Selanjutnya, pengecekan dilakukan dengan struktur pengkondisian if-else. Program akan mencetak "salah" jika nilai bool tidak benar, yang berarti salah. Jika nilai bool benar, maka program akan mencetak "benar".

Program ini menunjukkan penggunaan variabel boolean dalam pengambilan keputusan dan struktur pengkondisian if-else untuk mengatur alur eksekusi program berdasarkan kondisi variabel boolean.

## 7. KasusSwitch

```
import java.util.Scanner;

public class KasusSwitch {

    public static void main(String[] args) {

        char cc;
        Scanner masukan = new Scanner(System.in);

        System.out.print("Ketikkan sebuah huruf, akhiri dengan RETURN \n");
        cc = masukan.next().charAt(0);
        switch (cc) {
            case 'a': {
                System.out.print("Yang anda ketik adalah a\n");
                break;
            }
            case 'u': {
                System.out.print("Yang anda ketik adalah u\n");
                break;
            }
            case 'e': {
                System.out.print("Yang anda ketik adalah e\n");
                break;
            }
            case 'i': {
                System.out.print("Yang anda ketik adalah i\n");
                break;
            }
            case 'o': {
                System.out.print("Yang anda ketik adalah o\n");
                break;
            }
            default:
                System.out.print("Yang anda ketik adalah huruf mati\n");
        }
    }
}
```

```
Ketikkan sebuah huruf, akhiri dengan RETURN
R
Yang anda ketik adalah huruf mati
```

penggunaan struktur pengkondisian switch-case dalam Java. Aplikasi meminta pengguna memasukkan huruf dan kemudian memeriksa huruf tersebut.

Variabel cc adalah tipe data char yang digunakan untuk menyimpan huruf yang dimasukkan oleh pengguna.

Untuk membaca input pengguna, objek masukan kelas Scanner mencetak pesan "Ketikkan sebuah huruf, akhiri dengan RETURN".

Satu karakter yang dimasukkan oleh pengguna dan disimpan dalam variabel cc dapat dibaca dengan menggunakan metode next() dari objek masukan.

Nilai yang tersimpan dalam variabel cc dapat diperiksa dengan menggunakan struktur pengkondisian case switch. Nilai yang disimpan dalam variabel cc akan dievaluasi oleh setiap case dan mencetak pesan yang sesuai dengan nilai tersebut. Jika nilai cc tidak cocok dengan salah satu case, bagian default akan dieksekusi dan mencetak pesan bahwa huruf yang dimasukkan adalah "huruf mati".

## 8. Konstant

```
import java.util.Scanner;

public class Konstant {

    public static void main(String[] args) {

        final float PHI = 3.1415f;
        float r;
        Scanner masukan = new Scanner(System.in);

        System.out.print("Jari-jari lingkaran = ");
        r = masukan.nextFloat();

        System.out.println("Luas lingkaran = " + (PHI * r * r));
        System.out.println("Akhir program");
    }
}
```

```
Jari-jari lingkaran = 100
Luas lingkaran = 31415.0
Akhir program
```



penggunaan luas lingkaran dengan konstanta dalam Java. Setiap komponen program dijelaskan di bawah ini:

Deklarasi Konstanta: Konstanta PHI adalah konstanta bertipe float dengan nilai 3.1415. Ini akan digunakan sebagai nilai phi dalam rumus untuk menghitung luas lingkaran.

Deklarasi Variabel: Variabel r adalah tipe data floating point yang digunakan untuk menyimpan nilai jari-jari lingkaran yang dimasukkan pengguna.

Objek Pencarian: Objek masukan kelas Pencarian digunakan untuk membaca input pengguna.

Membaca Input: Pesan "Jari-jari lingkaran = " dicetak untuk memberikan instruksi kepada pengguna. Selanjutnya, nilai jari-jari lingkaran yang dimasukkan oleh pengguna dan disimpan dalam variabel r dihitung dengan metode nextFloat() dari objek input.

Menghitung dan Mencetak Luas Lingkaran: Rumus  $\text{PHI} * r * r$  digunakan untuk menghitung luas lingkaran, dan kemudian hasilnya dicetak bersama dengan pesan "Luas lingkaran =" dengan menggunakan System.out.println().

Pencetakan Pesan Akhir Program: Pesan "Akhir program" dicetak untuk menunjukkan bahwa program telah selesai.

## 9.Max2

```
import java.util.Scanner;

public class Max2 {

    public static void main(String[] args) {

        int a, b;
        Scanner masukan = new Scanner(System.in);
        /* Program */
        System.out.print("Maksimum dua bilangan : \n");
        System.out.print("Ketikkan dua bilangan, pisahkan dg RETURN :\n");
        a = masukan.nextInt();
        b = masukan.nextInt();
        System.out.println("Ke dua bilangan : a = " + a + " b = " + b);
        if (a >= b) {
            System.out.println("Nilai a yang maksimum " + a);
        } else /* a > b */ {
            System.out.println("Nilai b yang maksimum: " + b);
        }
    }
}
```

```
Maksimum dua bilangan :  
Ketikkan dua bilangan, pisahkan dg RETURN :  
100  
200  
Ke dua bilangan : a = 100 b = 200  
Nilai b yang maksimum: 200
```

program Java sederhana yang meminta pengguna memasukkan dua angka dan mencetak angka yang lebih besar di antara keduanya. Setiap komponen program dijelaskan di bawah ini:

Deklarasi Variabel: Variabel a dan b ditetapkan sebagai bilangan bulat untuk menyimpan dua angka yang dimasukkan oleh pengguna.

Objek Pencarian: Objek masukan kelas Pencarian digunakan untuk membaca input pengguna.

Pencetakan pesan petunjuk: Pesan "Maksimum dua bilangan" dan "Ketikkan dua bilangan, pisahkan dengan RETURN" dicetak untuk memberi pengguna petunjuk.

Membaca Input: Pengguna dapat membaca dua bilangan yang dimasukkan dan disimpan dalam variabel a dan b secara berurutan dengan menggunakan metode nextInt() dari objek masukan.

Untuk memberikan konfirmasi kepada pengguna, jumlah yang dimasukkan oleh pengguna dicetak kembali.

Struktur Pengkondisian If: Struktur pengkondisian if-else digunakan untuk mengevaluasi hubungan antara dua angka yang dimasukkan oleh pengguna. Jika bilangan a lebih besar atau sama dengan b, maka a dianggap sebagai nilai maksimum dan dicetak; sebaliknya, jika bilangan a lebih kecil atau sama dengan b, maka b dianggap sebagai nilai maksimum dan dicetak.

## 10. PriFor

```
import java.util.Scanner;  
  
public class PriFor {  
    public static void main(String[] args) {  
        int i, N;  
        Scanner masukan = new Scanner(System.in);  
  
        System.out.print("Baca N, print 1 s/d N ");  
        System.out.print("N = ");  
        N = masukan.nextInt();  
  
        for (i = 1; i <= N; i++) {  
            System.out.println(i);  
        }  
        System.out.println("Akhir program \n");  
    }  
}
```

```
Baca N, print 1 s/d N N = 5
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
Akhir program
```

penggunaan loop for dalam Java. Aplikasi ini membaca nilai N pengguna dan kemudian mencetak angka dari 1 hingga N.

Singkatnya, ini adalah penjelasan:

Program meminta pengguna untuk memasukkan nilai N, yang disimpan dalam variabel N. Kemudian program menggunakan loop untuk mengiterasi dari 1 hingga nilai N, mencetak nilai i setiap iterasi, dan setelah loop selesai, program mencetak pesan "Akhir program". Dengan demikian, output program akan mencetak angka dari 1 hingga nilai N, dan pesan "Akhir program" adalah pesan akhir dari program.

## 11. PrintIterasi

```
import java.util.Scanner;

public class PrintIterasi {

    public static void main(String[] args) {

        int N;
        int i;
        Scanner masukan = new Scanner(System.in);

        System.out.print("Nilai N > 0 = ");
        N = masukan.nextInt();
        i = 1;
        System.out.print("Print i dengan ITERATE : \n");
        for (;;) {
            System.out.println(i);
            if (i == N)
                break;
            else {
                i++;
            }
        }
    }
}
```

```
Nilai N > 0 = 5
Print i dengan ITERATE :
1
2
3
4
5
```

sederhana menggunakan perulangan for untuk mencetak nilai dari 1 hingga N, di mana N adalah bilangan bulat yang dimasukkan pengguna. Setiap komponen program dijelaskan di bawah ini:

Deklarasi Variabel: Untuk mengatur iterasi perulangan dan menyimpan nilai yang dimasukkan oleh pengguna, variabel N dan i dideklarasikan sebagai bilangan bulat.

Objek Pencarian: Objek masukan kelas Pencarian digunakan untuk membaca input pengguna.

Membaca Input: Pesan "Nilai N > 0 =" dicetak untuk meminta pengguna memasukkan nilai yang lebih besar dari 0. Nilai ini kemudian disimpan dalam variabel N dengan menggunakan metode nextInt() dari objek input.

Inisialisasi Variabel: Variabel i akan digunakan sebagai iterator dalam perulangan dan diinisialisasi dengan nilai 1.

Perulangan for: Perulangan ini digunakan untuk mencetak nilai i dari 1 hingga N. Perulangan ini akan berlanjut sampai nilai i mencapai nilai N.

Nilai i dicetak setiap kali menggunakan System.out.println(i).

Pemeriksaan Batas Perulangan: Dalam perulangan, pemeriksaan if memeriksa apakah nilai i sama dengan N. Jika nilainya sama, maka perulangan dihentikan dengan pernyataan break.

## 12. PrintRepeat

```
import java.util.Scanner;

public class PrintRepeat {
    public static void main(String[] args) {
        int N;
        int i;
        Scanner masukan = new Scanner(System.in);

        System.out.print("Nilai N > 0 = ");
        N = masukan.nextInt();

        i = 1;
        System.out.print("Print i dengan REPEAT: \n");
        do {
            System.out.print(i + "\n");
```

```

        i++;
    } while (i <= N);
}
}

```

```

Nilai N > 0 = 6
Print i dengan REPEAT:
1
2
3
4
5
6

```

penggunaan loop do-while dalam Java. Program ini meminta pengguna untuk memasukkan nilai N dan kemudian menggunakan loop do-while untuk mencetak angka dari 1 hingga N. Singkatnya, ini adalah penjelasan:

Program meminta pengguna untuk memasukkan nilai N, yang disimpan dalam variabel N. Program mencetak angka dari 1 hingga N dengan menggunakan loop do-while. Pada setiap iterasi, program mencetak nilai i, kemudian menambahkan nilai i dengan 1. Iterasi dilakukan selama nilai i kurang dari atau sama dengan N. Akibatnya, output program adalah angka dari 1 hingga nilai N.

### 13. PrintWhile

```

import java.util.Scanner;

public class PrintWhile {
    public static void main(String[] args) {
        int N;
        int i;
        Scanner masukan = new Scanner(System.in);

        System.out.print("Nilai N > 0 = ");
        N = masukan.nextInt();

        i = 1;
        System.out.print("Print i dengan WHILE: \n");
        while (i <= N) {
            System.out.println(i);
            i++;
        };
    }
}

```

```
Nilai N > 0 = 7
Print i dengan WHILE:
1
2
3
4
5
6
7
```

penggunaan loop while dalam Java. Program ini meminta pengguna untuk memasukkan nilai N dan kemudian menggunakan loop while untuk mencetak angka dari 1 hingga N.

Singkatnya, ini adalah penjelasan:

Program meminta pengguna untuk memasukkan nilai N, yang disimpan dalam variabel N.

Program mencetak angka dari 1 hingga N dengan menggunakan loop while. Pada setiap iterasi, program mencetak nilai i, kemudian menambahkan nilai i dengan 1. Iterasi dilakukan selama nilai i kurang dari atau sama dengan N. Akibatnya, output program adalah angka dari 1 hingga nilai N.

#### 14. PrintWhile1

```
import java.util.Scanner;

public class PrintWhile1 {
    public static void main(String[] args) {
        int N;
        int i = 1;
        Scanner masukan = new Scanner(System.in);

        System.out.print("Nilai N > 0 = ");
        N = masukan.nextInt();

        System.out.print("Print i dengan WHILE (ringkas): \n");
        while (i <= N) {
            System.out.println(i++);
        } /* (i > N) */
    }
}
```

```
Nilai N > 0 = 8
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7
8
```

Dalam aplikasi ini, pengguna diminta untuk memasukkan nilai N dan kemudian menggunakan loop while untuk mencetak angka dari 1 hingga N. Berikut adalah penjelasan singkatnya:

Program meminta pengguna untuk memasukkan nilai N, yang kemudian disimpan dalam variabel N. Variabel i diinisialisasi dengan nilai 1, dan program menggunakan loop while untuk mencetak nilai i dan kemudian meningkatkannya menggunakan operator ++ pada setiap iterasi. Iterasi berlanjut selama nilai i kurang dari atau sama dengan nilai N, dan loop berhenti jika nilai i melebihi nilai N.

Setelah iterasi selesai, program menyelesaikan eksekusi.

Program ini menghasilkan daftar angka dari 1 hingga N.

#### 15. PrintXinterasi

```
import java.util.Scanner;

public class PrintXinterasi {
    public static void main(String[] args) {
        int Sum = 0;
        int x;
        Scanner masukan = new Scanner(System.in);

        System.out.print("Masukkan nilai x (int), akhiri dg 999: ");
        x = masukan.nextInt();

        if (x == 999) {
            System.out.print("Kasus kosong\n");
        } else {
            Sum = x;
            for (;;) {
                System.out.print("Masukkan nilai x (int), akhiri dg 999: ");
                x = masukan.nextInt();

                if (x == 999)
                    break;
                else {
```

```

        Sum = Sum + x;
    }
}
System.out.println("Hasil penjumlahan = " + Sum);
}
}

```

```

Masukkan nilai x (int), akhiri dg 999: 100
Masukkan nilai x (int), akhiri dg 999: 200
Masukkan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 300

```

Dalam aplikasi ini, seseorang harus memasukkan nilai x berulang kali sampai nilai 999 ditemukan; kemudian, menjumlahkan semua nilai x yang dimasukkan kecuali 999. Ini adalah penjelasan yang singkat:

Program meminta pengguna untuk memasukkan nilai x yang berupa bilangan bulat dan akan terus meminta hingga pengguna memasukkan 999. Jika pengguna langsung memasukkan 999, program akan mencetak "Kasus kosong" dan berhenti. Jika pengguna memasukkan nilai lain selain 999, nilai tersebut akan ditambahkan ke variabel Sum. Program akan terus meminta pengguna untuk memasukkan nilai x hingga pengguna memasukkan 999. Setelah pengguna memasukkan 999, program akan mence

#### 16. PrintXRepeat

```

import java.util.Scanner;

public class PrintXRepeat {
    public static void main(String[] args) {
        int Sum;
        int x;
        Scanner masukan = new Scanner(System.in);

        System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");
        x = masukan.nextInt();

        if (x == 999) {
            System.out.print("Kasus kosong \n");
        } else {
            Sum = 0;
            do {
                Sum = Sum + x;
                System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");
                x = masukan.nextInt();
            } while (x != 999);
            System.out.println("Hasil penjumlahan = " + Sum);
        }
    }
}

```



```
Masukkan nilai x (int), akhiri dg 999 : 20
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 30
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 60
```

Dalam aplikasi ini, seseorang harus memasukkan nilai x berulang kali sampai nilai 999 ditemukan; kemudian, menjumlahkan semua nilai x yang dimasukkan kecuali 999. Ini adalah penjelasan yang singkat:

Program meminta pengguna untuk memasukkan nilai x yang berupa bilangan bulat dan akan terus meminta hingga pengguna memasukkan 999. Jika pengguna langsung memasukkan 999, program akan mencetak "Kasus kosong" dan berhenti. Jika pengguna memasukkan nilai lain selain 999, nilai tersebut akan ditambahkan ke variabel Sum. Program akan terus meminta pengguna untuk memasukkan nilai x hingga pengguna memasukkan 999. Setelah pengguna memasukkan 999, program akan berhenti

#### 17 PrintXWhile

```
import java.util.Scanner;

public class PrintXWhile {
    public static void main(String[] args) {
        int Sum;
        int x;
        Scanner masukan = new Scanner(System.in);

        Sum = 0;
        System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");
        x = masukan.nextInt();

        while (x != 999) {
            Sum = Sum + x;
            System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");
            x = masukan.nextInt();
        }

        System.out.println("Hasil penjumlahan = " + Sum);
    }
}
```

```
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 20
Masukkan nilai x (int), akhiri dg 999 : 30
Masukkan nilai x (int), akhiri dg 999 : 40
Masukkan nilai x (int), akhiri dg 999 : 50
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 150
```

Dalam aplikasi ini, seseorang harus memasukkan nilai x berulang kali sampai nilai 999 ditemukan; kemudian, menjumlahkan semua nilai x yang dimasukkan kecuali 999. Ini adalah penjelasan yang singkat:

Program meminta pengguna untuk memasukkan nilai x yang berupa bilangan bulat dan akan terus meminta hingga pengguna memasukkan 999. Setelah pengguna memasukkan nilai x, nilai tersebut akan ditambahkan ke variabel Sum. Setelah pengguna memasukkan nilai 999, program akan mencetak hasil penjumlahan dari semua nilai x yang dimasukkan oleh pengguna kecuali 999. Output program

#### 18 SubProgram

```
import java.util.Scanner;

public class SubProgram {
    public static int maxab(int a, int b) {

        return ((a >= b) ? a : b);
    }

    public static void tukar(int a, int b) {

        int temp;
        temp = a;
        a = b;
        b = temp;
        System.out.println("Ke dua bilangan setelah tukar: a = " + a + " b = "
+ b);
    }

    public static void main(String[] args) {

        int a, b;
        Scanner masukan = new Scanner(System.in);

        System.out.println("Maksimum dua bilangan");
        System.out.println("Ketikkan dua bilangan, pisahkan dg RETURN : ");
        a = masukan.nextInt();
        b = masukan.nextInt();

        System.out.println("Ke dua bilangan : a = " + a + " b = " + b);
    }
}
```

```

        System.out.println("Maksimum = " + (maxab(a, b)));
        System.out.println("Tukar kedua bilangan...");
        tukar(a, b);
    }
}

```

```

Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
10
20
Ke dua bilangan : a = 10 b = 20
Maksimum = 20
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 20 b = 10

```

Dua fungsi program subprogram ini adalah maxab, yang digunakan untuk menemukan nilai maksimum dari dua angka, dan tukar, yang digunakan untuk menukar nilai kedua angka tersebut. Ini adalah penjelasan yang singkat:

Fungsi maxab menerima dua parameter bilangan bulat a dan b, lalu mengembalikan bilangan terbesar di antara keduanya. Fungsi tukar menerima dua parameter bilangan bulat a dan b, lalu menukar nilai keduanya. Program utama meminta pengguna untuk memasukkan dua bilangan bulat, kemudian mencetak bilangan yang dimasukkan; kemudian, program menggunakan fungsi maxab untuk mencetak nilai maksimum dari dua bilangan tersebut. Selanjutnya, program memanggil fungsi tukar untuk menukar n dari dua bilangan bulat bilangan dan mengubah nilai kedua angka tersebut.

19 Tempair

```

import java.util.Scanner;

public class Tempair {

    public static void main(String[] args) {

        int T;
        Scanner masukan=new Scanner(System.in);
        System.out.print ("Contoh IF tiga kasus \n");
        System.out.print ("Temperatur (der. C) = ");
        T=masukan.nextInt();
        if (T < 0) {
            System.out.print ("Wujud air beku \n"+ T);
        }else if ((0 <= T) && (T <= 100)){

```

```
System.out.print ("Wujud air cair \n"+ T);  
}else if (T > 100){  
System.out.print ("Wujud air uap/gas \n"+ T);  
};  
}  
}
```

```
Contoh IF tiga kasus  
Temperatur (der. C) = 120  
Wujud air uap/gas  
120
```

Contoh penggunaan struktur kontrol if dengan tiga kasus untuk menentukan jenis air berdasarkan suhu dalam derajat Celsius adalah Tempair. Berikut adalah penjelasan program yang singkat:

Program meminta pengguna untuk memasukkan suhu dalam derajat Celsius, yang disimpan dalam variabel T. Jika suhu kurang dari 0, program akan mencetak "Wujud air beku" dengan nilai suhunya, jika suhu antara 0 dan 100, program akan mencetak "Wujud air cair" dengan nilai suhunya, dan jika suhu lebih dari 100, program akan mencetak "Wujud air uap/gas" dengan nilai suhunya.

Tanda; setelah blok else pada akhir struktur kontrol tidak mempengaruhi hasil program; namun, karena blok tersebut tidak memiliki pernyataan lain setelahnya, blok tersebut dapat dihilangkan.

Akibatnya, program Ini menunjukkan wujud air berdasarkan suhu yang dimasukkan pengguna.