

LAPORAN RESMI

Praktikum 3 Percabangan dan Perulangan

Mata Kuliah: Praktek Pemrograman Berbasis Objek



Disusun oleh:

M. Ainur Ramadhan (3122500047)

2 D3 Teknik Informatika B

Dosen Pengampu: Yanuar Risah Prayogi S.Kom., M.Kom.

**PROGRAM STUDI D3 TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

2023/2024

A. TUGAS PENDAHULUAN

1. Sebutkan dan jelaskan berbagai macam sintaks percabangan yang digunakan di Java!

Jawab :

1. If-Else: Mengizinkan eksekusi kode berdasarkan kondisi, dengan dua opsi: jika benar, jalankan bagian "if", jika salah, jalankan bagian "else".
 2. If - Else If - Else: Digunakan untuk memilih tindakan dari beberapa opsi berurutan, berdasarkan kondisi yang sesuai.
 3. Switch: Menentukan tindakan berdasarkan nilai ekspresi tertentu, seperti kasus-kasus dalam sebuah kasus.
 4. Ternary (Conditional Operator): Memberikan nilai berdasarkan kondisi, menghasilkan satu nilai jika benar, dan nilai lain jika salah.
2. Sebutkan dan jelaskan berbagai macam sintaks perulangan yang digunakan di Java!

Jawab :

1. for : Mengulang blok kode dengan jumlah iterasi yang sudah ditentukan sebelumnya.
2. while : Mengulang blok kode selama kondisi tertentu benar (true).
3. do-while : Mengulang blok kode setidaknya sekali sebelum memeriksa kondisi, dan melanjutkan jika kondisi masih benar.
4. foreach : Mengulang elemen-elemen dalam koleksi seperti array.
5. break dan continue : break digunakan untuk menghentikan perulangan secara paksa, continue digunakan untuk melompati iterasi saat ini dan lanjut ke iterasi berikutnya.

B. LATIHAN

1. Menampilkan bilangan factorial

Source Code :

Faktorial.java

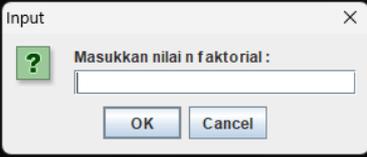
```
import javax.swing.JOptionPane;

public class Faktorial {
    public static void main (String args[]){
        int n;
        int hasil=1;
        String str = JOptionPane.showInputDialog("Masukkan nilai n
faktorial :");
        n = Integer.parseInt(str);
        for (int i = 1; i <= n ; i++)
        {
            hasil *= i;
            System.out.print(i);
            if(i<n){
                System.out.print(" x ");
            }
        }
        System.out.print(" = " + hasil);
    }
}
```

```
}  
}
```

Output

```
C:\Users\USER\OOP\Praktikum\Praktikum 3\Latihan>javac *.java  
C:\Users\USER\OOP\Praktikum\Praktikum 3\Latihan>java Faktorial
```



```
C:\Users\USER\OOP\Praktikum\Praktikum 3\Latihan>java Faktorial  
1 x 2 x 3 x 4 = 24  
C:\Users\USER\OOP\Praktikum\Praktikum 3\Latihan>_
```

2. Menampilkan deret bilangan genap

Source Code

DeretBilGenap.java

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

Output

```
C:\Users\USER\OOP\Praktikum\Praktikum 3\Latihan>java DeretBilGenap  
2 4 8 10 12 14 16 18 20  
C:\Users\USER\OOP\Praktikum\Praktikum 3\Latihan>_
```

C. TUGAS

1. Deret Fibonacci

Source Code

DeretFib.java

```
import javax.swing.JOptionPane;
```

```

public class DeretFib{
    public static void main (String args[]){
        int n ;
        int j = 0;
        int k = 1;
        String str = JOptionPane.showInputDialog("Masukkan berapa n
deret fibbonaci ? ");
        n = Integer.parseInt(str);
        for (int i= 1;i <=n; i++)
        {
            System.out.print(k + " ");
            int next = j + k;
            j = k;
            k = next;
        }
    }
}

```

Output

```

C:\Users\USER\00P\Praktikum\Praktikum 3\Tugas>javac *.java
C:\Users\USER\00P\Praktikum\Praktikum 3\Tugas>java DeretFib

1 1 2 3 5 8 13 21
C:\Users\USER\00P\Praktikum\Praktikum 3\Tugas>

```

Analisa

Kode program tersebut digunakan untuk menghasilkan dan mencetak deret Fibonacci sepanjang "n" angka, di mana "n" dimasukkan oleh pengguna melalui dialog input. Setelah mengimpor pustaka `javax.swing.JOptionPane`, program meminta pengguna untuk memasukkan jumlah angka dalam deret Fibonacci yang diinginkan. Selanjutnya, menggunakan perulangan `for`, program menghitung dan mencetak angka dalam deret Fibonacci. Variabel "j" dan "k" mewakili dua angka terakhir dalam deret, dengan nilai awal 0 dan 1. Dalam setiap iterasi, angka "k" dicetak, kemudian dihitung angka berikutnya dalam deret sebagai penjumlahan dari "j" dan "k". Nilai "j" diperbarui menjadi nilai "k" sebelumnya, dan nilai "k" diperbarui menjadi angka berikutnya dalam deret. Akhirnya, program mencetak deret Fibonacci sesuai dengan jumlah angka yang diminta oleh pengguna. Program ini menghasilkan dan menampilkan deret Fibonacci dengan jumlah angka sesuai input pengguna.

2. Menentukan tahun kabisat Source Code

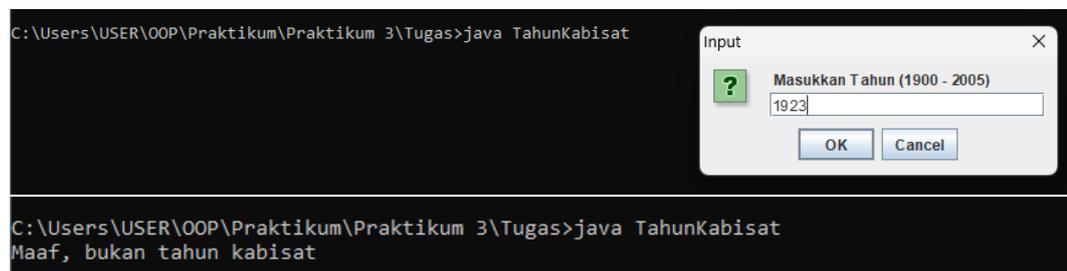
TahunKabisat.java

```
import javax.swing.JOptionPane;

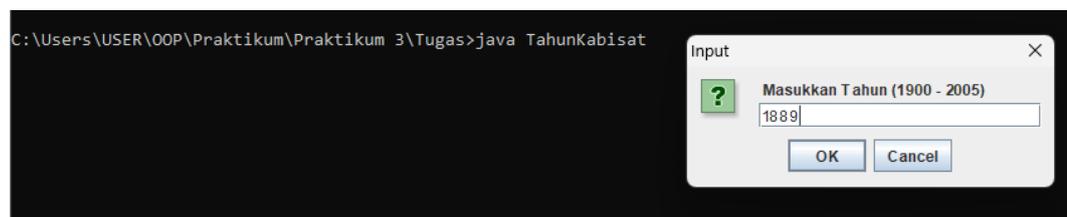
public class TahunKabisat {
    public static void main (String args[]){
        int tahun;
        String str = JOptionPane.showInputDialog(" Masukkan Tahun
(1900 - 2005)");
        tahun = Integer.parseInt(str);
        if (tahun > 2005){
            System.out.println("Maaf, Tahun input diatas 2005");
        } else if (tahun % 400 == 0 || tahun % 4 == 0){
            System.out.println(tahun + " Merupakan tahun kabisat");
        } else if (tahun % 100 == 0 ){
            System.out.println("Maaf, bukan tahun kabisat" );
        } else if (tahun < 1900){
            System.out.println("Maaf, tahun input dibawah 1900");
        } else {
            System.out.println("Maaf, bukan tahun kabisat");
        }
    }
}
```

Output

⇒ Untuk tahun yang bukan tahun kabisat

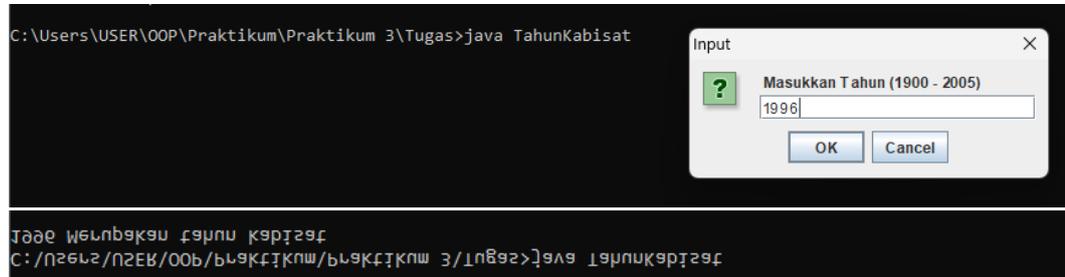


⇒ Untuk tahun yang dibawah tahun 1900

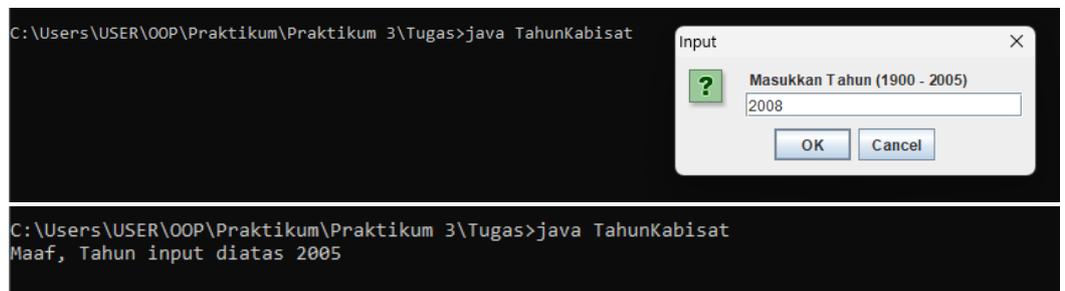


```
C:\Users\USER\OOP\Praktikum\Praktikum 3\Tugas>java TahunKabisat
Maaf, tahun input dibawah 1900
```

⇒ Untuk tahun yang merupakan tahun kabisat



⇒ Untuk tahun yang diatas tahun 2005



Analisa

Kode program tersebut digunakan untuk menentukan apakah tahun yang dimasukkan pengguna melalui dialog input adalah tahun kabisat atau tidak. Setelah mengimpor pustaka `javax.swing.JOptionPane`, program meminta pengguna untuk memasukkan tahun antara 1900 hingga 2005 melalui kotak dialog. Kemudian tahun tersebut diuji menggunakan serangkaian kondisi. Pertama, jika tahun melebihi 2005, program memberikan pesan bahwa tahun melebihi batas. Selanjutnya, program memeriksa apakah tahun tersebut habis dibagi 400 atau 4, jika iya, tahun dianggap kabisat dan hasilnya dicetak. Jika tidak, program memeriksa apakah tahun tersebut habis dibagi 100, jika iya, maka tahun bukan tahun kabisat. Jika tahun berada di bawah 1900, program juga memberikan pesan. Jika tidak ada kondisi yang terpenuhi, program menyimpulkan bahwa tahun bukan tahun kabisat. Program ini memberikan tanggapan berdasarkan aturan kalender tahun kabisat, di mana tahun kabisat habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100.

KESIMPULAN

Dalam bahasa pemrograman Java, penggunaan sintaks percabangan dan perulangan sangat penting untuk mengontrol alur eksekusi program. Sintaks percabangan memungkinkan pengambilan keputusan berdasarkan kondisi tertentu, dengan variasi seperti if-else dan switch. Di sisi lain, sintaks perulangan memungkinkan pengulangan blok kode untuk melakukan tugas

yang sama berulang kali, melalui konsep seperti for, while, dan do-while, serta kemampuan untuk beroperasi pada koleksi menggunakan foreach. Pemahaman tentang sintaks-sintaks ini memungkinkan pengembang untuk membuat program yang lebih dinamis dan efisien, menghasilkan kode yang lebih terstruktur dan sesuai dengan kebutuhan aplikasi.