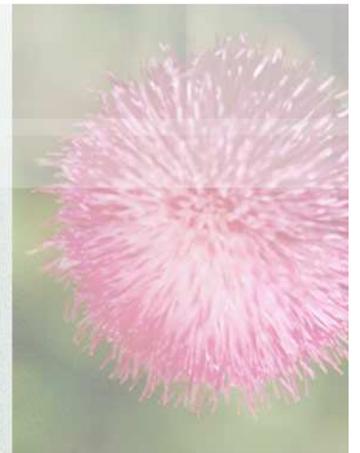




# Flow Control

Tessy Badriyah, SKom. MT.

<http://lecturer.eepis-its.edu/~tessy>



# Tujuan Pembelajaran



- **Mempelajari perintah pencabangan dengan if dan switch**
- **Mempelajari perintah perulangan dengan for, while, dan do-while**
- **Mempelajari fungsi khusus yang mengubah alur kontrol program seperti break dan continue**

# Perintah Kontrol



- **Digunakan untuk merubah urutan eksekusi program**
  - **Perintah kondisional : if, switch**
  - **Perintah perulangan : for, while do-while**
  - **Perintah pencabangan : break, continue, return**



# Statement if-else



- **Bentuk umum**
  - if (kondisi)  
    <statement>;
  - if (kondisi)  
    <statement>;  
else  
    <statement>;
  - if (kondisi)  
    <statement>;  
elseif (kondisi)  
    <statement>;  
.....  
else  
    <statement>;

Hasil pada ekspresi  
Kondisional yaitu logika  
True atau False

# Percobaan 1



- Contoh penggunaan if untuk konversi nilai angka ke nilai huruf

```
Konversi.java *
public class Konversi {
    public static void main(String [] args) {
        int NA;
        String NH;
        NA=78;
        if (NA>80)
            NH="A";
        else if (NA>70)
            NH="AB";
        else if (NA>65)
            NH="B";
        else if (NA>60)
            NH="BC";
        else if (NA>55)
            NH="C";
        else if (NA>40)
            NH="D";
        else
            NH="E";
        System.out.println("Nilai angka = "+NA);
        System.out.println("Nilai huruf = "+NH);
    }
}
```

# Perintah pencabangan dengan switch



- **Cara penulisan :**

```
switch (expr1) {  
    case constant2:  
        statements;  
        break;  
    case constant3:  
        statements;  
        break;  
    default:  
        statements;  
        break;  
}
```



# Percobaan 2



- Penggunaan switch untuk mencari nama hari dalam seminggu

```
Hari.java *
public class Hari {
    public static void main(String [] args) {
        int urutan=2;
        String namahari;
        switch (urutan)
        { case 1 : namahari="Senin"; break;
          case 2 : namahari="Selasa"; break;
          case 3 : namahari="Rabu"; break;
          case 4 : namahari="Kamis"; break;
          case 5 : namahari="Jum'at"; break;
          case 6 : namahari="Sabtu"; break;
          default : namahari="Minggu"; break;
        }
        System.out.println("Urutan : "+urutan+" , nama hari = "+namahari);
    }
}
```

# Perintah perulangan : for

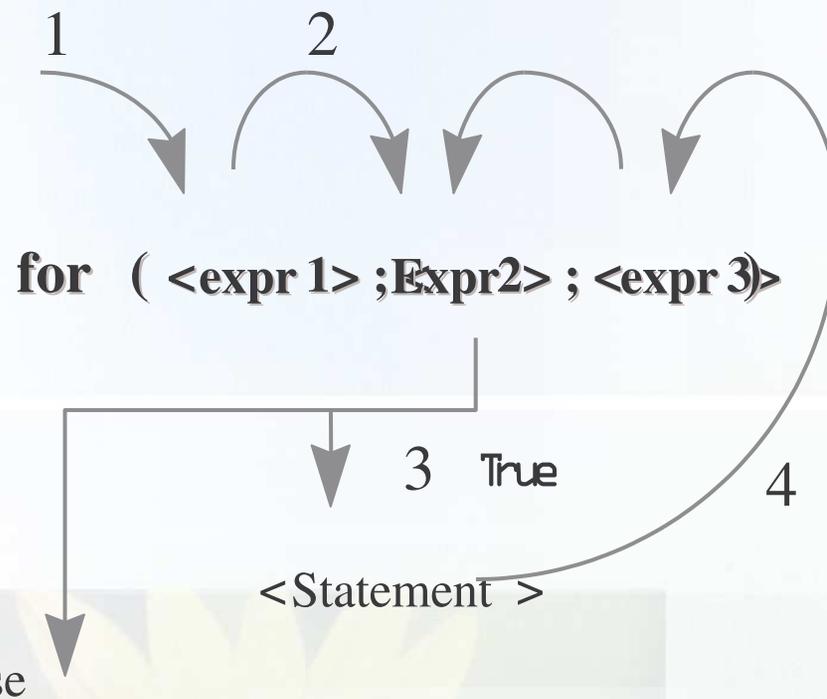


- Mengulang urutan perintah sebanyak yang sudah didefinisikan
- Bentuk umum :  
**for (<expr1>; <expr2>; <expr3>)  
    <statement>**
  - <expr1> inisialisasi variabel kontrol
  - <expr2> dikerjakan selama kondisi ini terpenuhi
  - <expr3> perintah increment/decrement

# Perintah perulangan dengan for



- Urut-urutan eksekusi pada perulangan for :



# Percobaan 3



- **Buat program untuk menampilkan bilangan ganjil yang lebih kecil dari 20 menggunakan perulangan for**

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

```
C:\j2sdk1.4.1_01\bin\java.exe -classpath "C:\j2sdk1.4
1 3 5 7 9 11 13 15 17 19 Finished executing
```



# Percobaan 4



- Menampilkan nilai Faktorial dari suatu bilangan menggunakan perintah for  
 $N! = N \times (N-1) \times (N-2) \times \dots \times 1$   
 $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

```
FaktorialFor.java *
public class FaktorialFor {
    public static void main(String [] args) {
        int N=5;
        int fakt=1;
        for (int i=1; i<=N; i++)
            fakt=fakt*i;
        System.out.println("5! = "+fakt);
    }
}
```

```
C:\j2sdk1.4.1_01\bin\java.e:
5! = 120
Finished executing
```



# Percobaan 5



- Menampilkan deret Fibonacci menggunakan perulangan for

- Deret fibonacci dari suku ke-1 sampai 7

1 1 2 3 5 8 13

$\text{Fibo}(i) = \text{Fibo}(i-1) + \text{Fibo}(i-2)$

```
FiboFor.java *
public class FiboFor {
    public static void main(String [] args) {
        int N=7,i;
        int a=1,b=1,c;
        System.out.println("Fibonacci suku ke-1 sampai 7 : ");
        System.out.print(a+" "+b+" ");
        for (i=3; i<=7; i++) {
            c=a+b;
            a=b; b=c;
            System.out.print(c+" ");
        }
    }
}
```

```
C:\j2sdk1.4.1_01\bin\java.exe -classpath "C:\j2sdk1.4.1_01\bin" FiboFor
Fibonacci suku ke-1 sampai 7 :
1 1 2 3 5 8 13 Finished executing
```

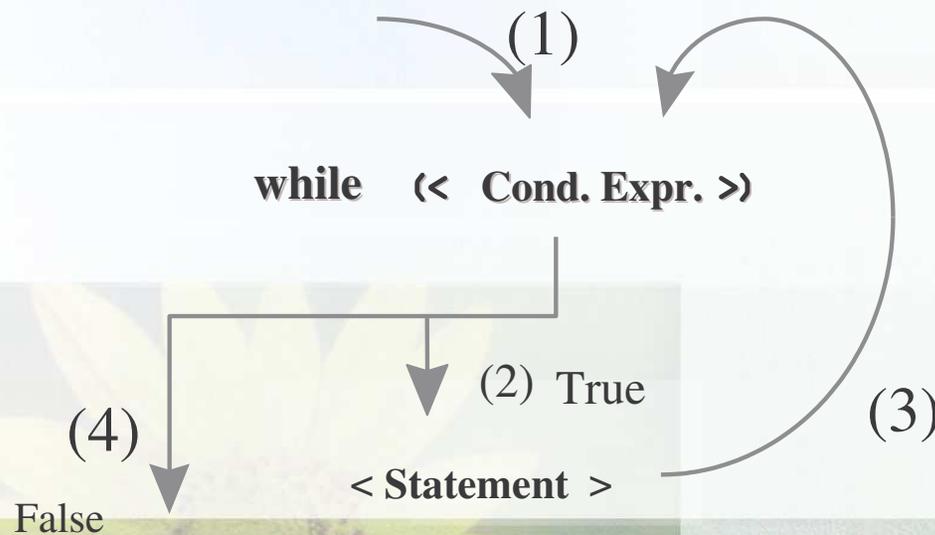


# Perintah perulangan while



- **Bentuk umum :**  
**while (<cond. Expr>)**  
**<statement>;**

- **Urut-urutan eksekusi :**

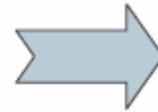


# Perbandingan for dengan while



- **Perbandingan for dengan while**

```
for (i = 0; i < N; ++i)
  s += i;
```



```
i = 0;
while (i < N) {
  s += i;
  ++i;
}
```

# Percobaan 6



- **Buat program untuk menampilkan bilangan ganjil yang lebih kecil dari 20 menggunakan while**

```
Ganjil.java *
public class Ganjil {
    public static void main(String [] args) {
        int i=0;
        while (i<20) {
            if ((i%2)>0)
                System.out.print(i + " ");
            i++;
        }
    }
}
```

```
C:\j2sdk1.4.1_01\bin\java.exe -classpath "C:\j2sdk1.4
1 3 5 7 9 11 13 15 17 19 Finished executing
```



# Percobaan 7



- Menampilkan nilai Faktorial dari suatu bilangan menggunakan perintah while  
 $N ! = N \times (N-1) \times (N-2) \times \dots \times 1$   
 $5 ! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

```
FaktorialWhile.java
public class FaktorialWhile {
    public static void main(String [] args) {
        int i=1,N=5;
        int fakt=1;
        while (i<=N) {
            fakt=fakt*i;
            i++;
        }
        System.out.println("5! = "+fakt);
    }
}
```



# Perintah perulangan do - while



- Setelah menjalankan statement, kemudian kondisi diperiksa apakah menghasilkan nilai true atau false
- Bentuk umum :  
do

**<statements>;**  
**while (<cond. Expr>;**



# Percobaan 8



- **Buat program untuk menampilkan bilangan ganjil yang lebih kecil dari 20 menggunakan while**

```
Ganjil.java
public class Ganjil {
    public static void main(String [] args) {
        int i=0;
        do {
            if ((i%2)>0)
                System.out.print(i + " ");
            i++;
        } while (i<20);
    }
}
```

```
C:\j2sdk1.4.1_01\bin\java.exe -classpath "C:\j2sdk1.4.
1 3 5 7 9 11 13 15 17 19 Finished executing
```



# Percobaan 9



- Menampilkan nilai Faktorial dari suatu bilangan menggunakan perintah **do - while**

$$N ! = N \times (N-1) \times (N-2) \times \dots \times 1$$

$$5 ! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

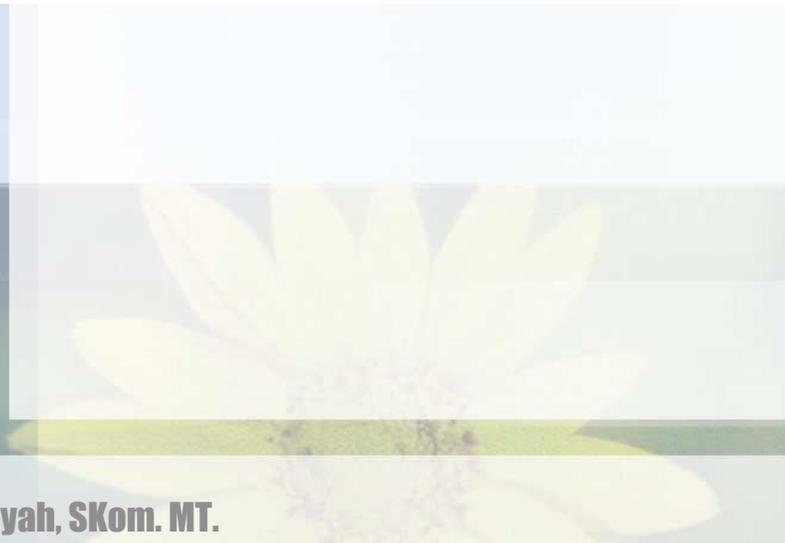
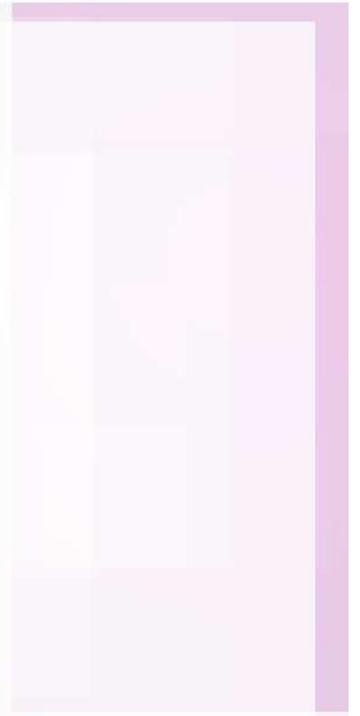
```
FaktorialDoWhile.java
public class FaktorialDoWhile {
    public static void main(String [] args) {
        int i=1,N=5;
        int fakt=1;
        do {
            fakt=fakt*i;
            i++;
        }while (i<=N);
        System.out.println("5! = "+fakt);
    }
}
```



# Perintah khusus yang merubah alur perulangan



- **break [label];**
- **continue [label];**



# Perintah break;



- Untuk keluar dari blok perintah yang diulang
- Bentuk umum perintah break :  
**break [label];**

```
int i;  
while (true) {  
    if (i==3)  
        break;  
    System.out.println("iterasi ke -"+i);  
    ++i;  
}
```

A red arrow starts from the 'break;' statement and points to the right, then turns downwards and then right again, pointing to the closing curly brace of the while loop, indicating that the loop is terminated.

# Perintah break dengan label



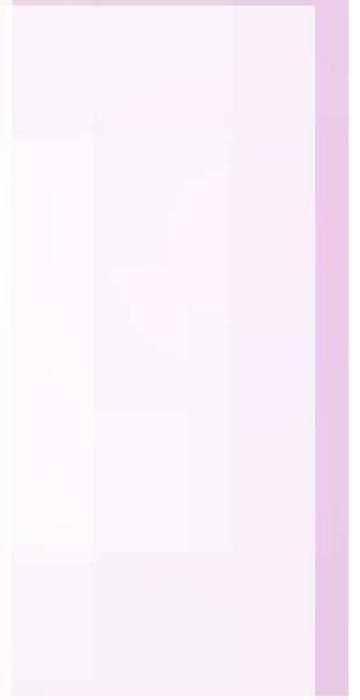
- Dapat digunakan seperti perintah goto
- Contoh penggunaan :

```
namaLabel:  
  St.1 {  
    St.2 {  
      // .....  
      break;  
      // .....  
      break namaLabel;  
    }  
    // .....  
  }  
}
```

# Perintah pencabangan continue



- Untuk berpindah ke awal perulangan
- Bentuk umum :  
`continue [Label];`



# Percobaan 10



- Tampilkan bilangan dari 1 sampai 10 kecuali bilangan 3

```
ContohContinue.java *  
  
public class ContohContinue {  
    public static void main(String [] args) {  
        int i;  
        for (i=1; i<11; i++) {  
            if (i==3)  
                continue;  
            System.out.println(i+" ");  
        }  
    }  
}
```

```
C:\j2sdk1.4.1_01\bin  
1  
2  
4  
5  
6  
7  
8  
9  
10  
Finished executing
```

# Perintah continue dengan label



**namaLabel:**

```
→ St.1 {  
  → St.2 {  
    // .....  
    continue;  
    // .....  
    continue namaLabel;  
  }  
  // .....  
}
```





# TUGAS





1. Tampilkan bilangan fibonacci dengan menggunakan perulangan while
2. Tampilkan bilangan genap dari 1 sampai 10 kecuali bilangan 4, menggunakan :
  - Perulangan for
  - Perulangan while
3. Gunakan perintah break untuk menampilkan bilangan ganjil dari 1 sampai N, gunakan bentuk perulangan :  
while (true) {  
.....  
}





Selesai

24 Maret 2007 dan 4 Mei 2007

