

*Modul 3:
Kendali program dan teknik
penyimpanan data*

Setelah mengikuti mata kuliah ini mahasiswa dapat membuat program java dengan konsep kondisional, loop dan mengelola data dengan array.

Pengantar :

Dalam modul ini akan diuraikan beberapa topik bahasan yaitu :

- 1) Statemen if .. dan if .. else, block program : {..}
- 2) statemen switch .. *case* .. *break*
- 3) loop *for* dan *while*
- 4) konsep array

1. Statemen kondisional

Statemen kondisional dalam JAVA memiliki 4 macam syntax :

- `If (kondisi) statemen ;`

atau

- `If (kondisi) {blok statemen} ;`

atau

- `If (kondisi) statemen1 else statemen2;`

atau

- `If (kondisi) {blok statemen -1} else {blok statemen-2} ;`

BLOK STATEMEN : kumpulan statemen yang berada dalam blok { }

Contoh Program 1

```
public class Sales
{ public static void main(String[] args)
  {
    String NAMA,PESAN;
    double JUAL,BONUS;

    NAMA="BAMBANG";
    JUAL=99000;
    if (JUAL>100000)
      {BONUS=0.1*JUAL; PESAN="SELAMAT !!";}
    else
      {BONUS=0; PESAN="ANDA HARUS LEBIH GIAT LAGI";}

    System.out.println("Saudara "+NAMA );
    System.out.println("Bonus Anda sebesar : "+BONUS );
    System.out.println(PESAN);
  }
}
```

Hasil program

Saudara BAMBANG

Bonus Anda sebesar : 0

ANDA HARUS LEBIH GIAT LAGI

Contoh program 2

- Program menentukan nilai A,B,C,D,E

```
class NilaiHuruf
{ int nilai=75; char Huruf;
  public static void main (String [] arg)
  {if (nilai<40)Huruf='E';
    else if (nilai<55) Huruf='D';
    else if (nilai<70) Huruf = 'C';
    else if (nilai <80) Huruf='B';
    else Huruf='A';
    System.out.println ("Nilai angka = "+nilai+" Huruf :
    "+Huruf);

    } // akhir main
} //akhir definisi class
```

Hasil program :

Nilai angka = 75 Huruf : B

2. switch .. case .. break

Syntax :

```
switch ( variabelSeleksi )
{ case nilai1:
    { statemen1;
      ....
      [break; ] }
  case nilai2:
    { statemen2
      ....
      [break; ] }
  [default :]
    { statemen_n;
      ...
      [break; ] }
}
```

Jika variabelSeleksi nilainya sama dengan nilai1 maka statemen1 dijalankan

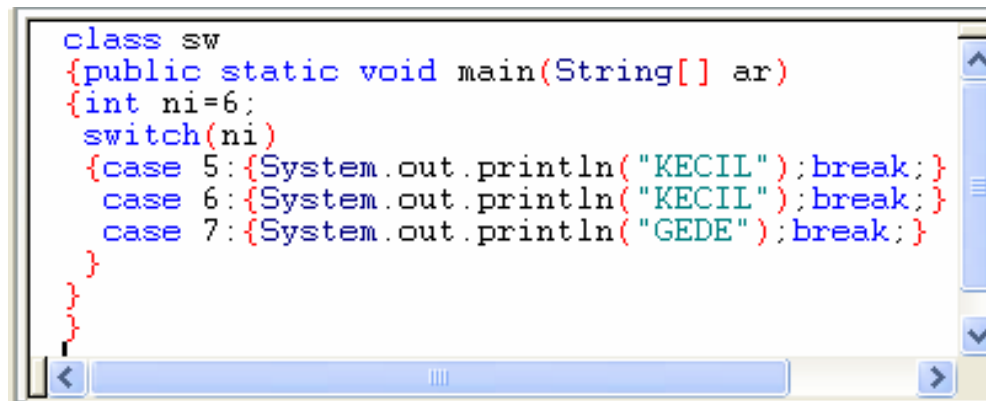
Jika nilainya sama dengan nilai2 maka statemen2 dijalankan

Jika statemen break ditemukan maka kontrol program akan keluar dari **switch**

Hasil contoh aplikasi switch (1)

Edit program dengan editor teks

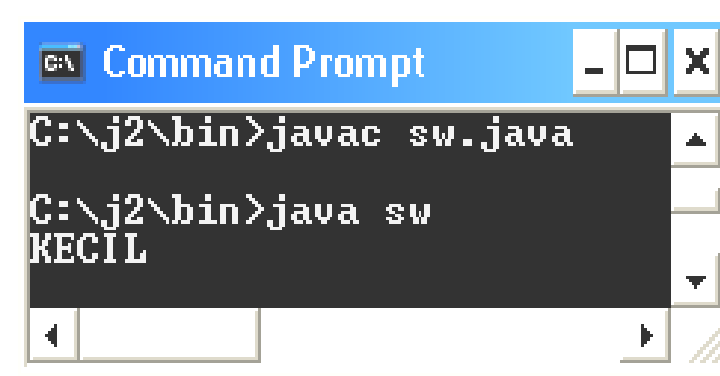
Text-Pad :



```
class sw
{public static void main(String[] ar)
{int ni=6;
switch(ni)
{case 5: {System.out.println("KECIL");break;}
case 6: {System.out.println("KECIL");break;}
case 7: {System.out.println("GEDE");break;}
}
}
```

Compilasi dengan javac.exe

Eksekusi dengan java.exe



```
Command Prompt
C:\j2\bin>javac sw.java
C:\j2\bin>java sw
KECIL
```


Pembahasan break

Jika dibelakang deretan statemen tidak dipasang statemen **break**; maka setelah evaluasi statemen yang sesuai **case**, alur program tidak loncat keluar tetapi mengevaluasi kembali statemen **case** yang dibawahnya

Sebagai contoh jika statemen **break** dibelakang **case** 6 dihilangkan, maka alur program akan mengevaluasi **case** dibawahnya

Hasil program seperti disebelah ini

```
class sw
{public static void main(String[] ar)
{int ni=6;
switch(ni)
{case 5:{System.out.println("KECIL");break;}
case 6:{System.out.println("KECIL");}
case 7:{System.out.println("GEDE");break;}
}}
}
```

```
C:\ Command Prompt
C:\j2\bin>javac sw.java
C:\j2\bin>java sw
KECIL
GEDE
```

3. Loop

Pada prinsipnya secara logika ada dua macam jenis perulangan (loop), yaitu :

1. **Perulangan terhitung** : yaitu proses perulangan dimana kendali ulangnya adalah counter. Proses diulang selama nilai counter bergerak dari nilai awal ke nilai akhir. Kendali loop ini adalah **for** dengan syntax :

```
for (counter = nilaiawal; kondisi; perubahan counter)
{ body of loop}
```

Contoh :

```
for(int i=0; i<5; i++)
    System.out.println(i+" ");
```

Hasil:

```
0 1 2 3 4
```

Contoh-contoh loop for

```
for(int i=0;i<10;i+=2)
    System.out.print(i+" ");
```

Hasil : 0 2 4 6 8

```
for(int i=10;i>0;i-=3)
    System.out.print(i+" ");
```

Hasil : 10 7 4 1

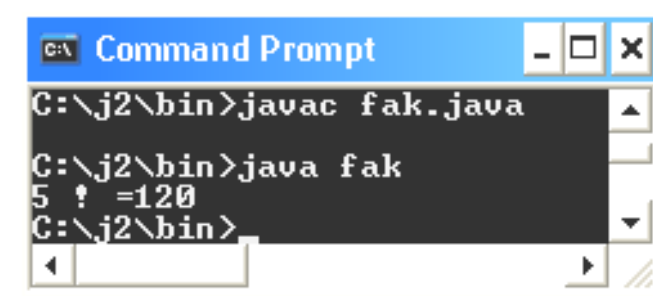
Program mencari n faktorial :

```
int f=1;
for(int i=1;i<=n;i++) f *=i;
```

Program faktorial
 $n! = 1 * 2 * \dots * n$

```
class fak
{public static void main(String[] ar)
{ int f=1; int n=5;
  for(int i=1;i<=n; i++) f*=i;
  System.out.print(n+" ! =" +f);
}
}
```

Hasil program :



```
C:\> Command Prompt
C:\j2\bin>javac fak.java
C:\j2\bin>java fak
5 ! =120
C:\j2\bin>
```

Perulangan while(.){} dan do {}while(.)

2. **Perulangan tak hingga** : yaitu proses perulangan dimana kendali ulangnya adalah kondisi. Proses diulang selama nilai kondisi benar masih dipenuhi. Kendali loop ini adalah **while** dengan syntax :

```
while(kondisi)
    { body of loop }
```

Contoh :

```
int n=5;
while(n>0)
    { System.out.println(n+" "); n--;} 
```

Hasil:

5 4 3 2 1

Dalam body of loop **while** ada bagian yang merubah kondisi sehingga suatu saat kondisi itu menjadi tidak lagi benar, dan proses berhenti.

perulangan do{ } while(..)

Perulangan while juga dapat mengambil bentuk dimana pengecekan kondisi dilakukan setelah blok statemen dieksekusi. Syntax adalah :

```
do { body of loop } while(kondisi)
```

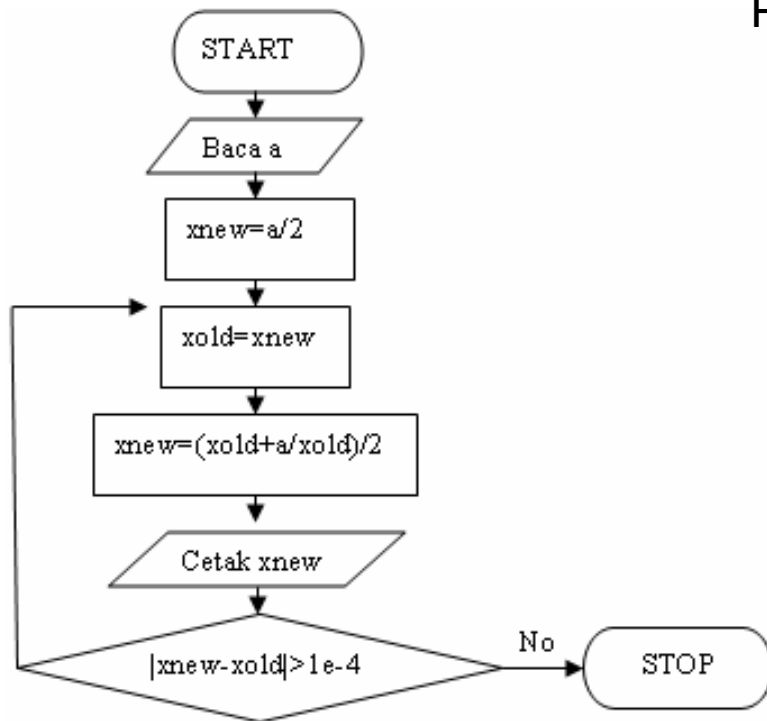
Contoh :

```
int n=5;  
do{System.out.println(n+" "); n--;}  
while(n>0;
```

Hasil:

```
5 4 3 2 1
```

Contoh loop while untuk mencari akar suatu bilangan real



Programnya :

```
public class Akar
{ public static void main(String[] args)
  { double a=26;
    double xnew=a/2;
    double xold;
    do {
      xold=xnew;
      xnew=(xold+a/xold)/2;
      System.out.println("xnew : "+xnew);
    }
    while (Math.abs(xnew-xold)>1e-4);
    System.out.println("AKAR DARI "+a+" = "+xnew);
  }
}
```

Hasil programnya:

```
C:\j2\bin>java Akar
xnew : 7.5
xnew : 5.483333333333333
xnew : 5.112487335359676
xnew : 5.099037252729456
xnew : 5.099019513623642
AKAR KUADRAT DARI 26.0 = 5.09901951362
```

The screenshot shows a Windows Command Prompt window with the title 'C:\ Command Prompt'. The user has executed the command 'java Akar'. The output displays the iterative values of 'xnew' and the final result: 'AKAR KUADRAT DARI 26.0 = 5.09901951362'.

4. Array

Array adalah sederetan data atau objek yang sejenis yang diakses menggunakan indeks

Indeks dimulai dari 0 sampai n-1, dimana n adalah jumlah elemen

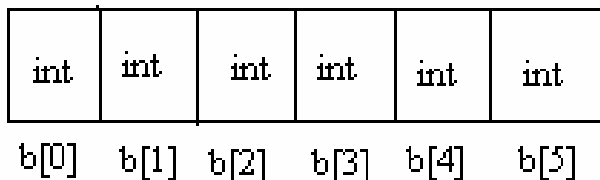
Syntax definisi adalah :

```
TipeData [] nama_array=new TipeData[n];
```

Contoh :

```
int [] b=new int[5];
```

artinya b adalah array of int dengan cacah maksimal elemen 5



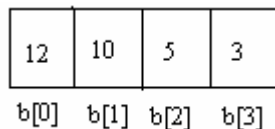
Kedalam tiap-tiap elemen b[i] selanjutnya dapat dimasukkan nilai int dengan berbagai cara yang mungkin

Array 1 Dimensi

- Array dapat berupa variabel atau berupa konstanta.
- Array dapat berupa array dari int, long, float, String atau bahkan objek
- Definisi array 1 dimensi untuk konstanta :

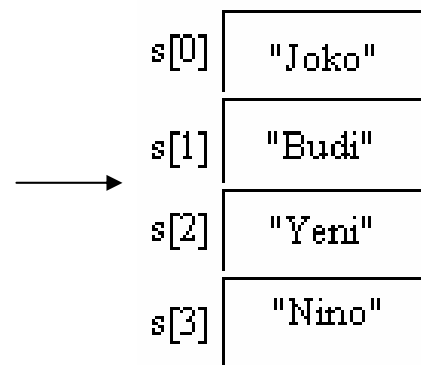
```
int [ ] b = { 12, 10, 5, 3 } ;
```

adalah mendefinisikan array 1 dimensi dari int yang berelemen data 12, 10, 5 dan 3 seperti gambar berikut



Contoh definisi konstanta String:

```
String [ ] s  
= { "Joko", "Budi", "Yeni", "Nino" }
```

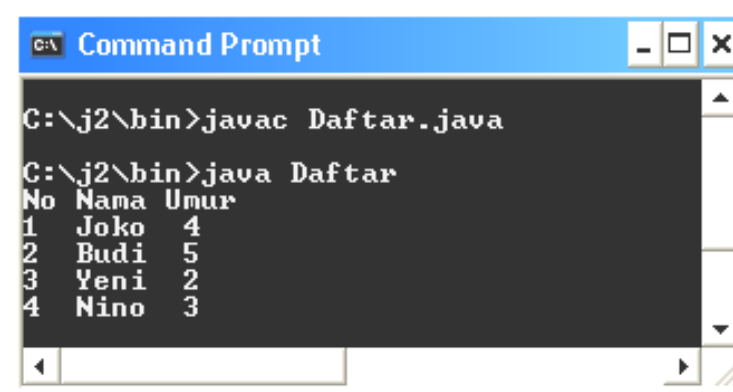


Contoh program array 1 dim

- Program mendefinisikan array berisi nama dan umur

```
class Daftar
{public static void main(String[] ar)
{ int[] um={4,5,2,3};
String []nm={"Joko","Budi","Yeni","Nino"};
System.out.println("No Nama Umur");
for(int i=0;i<4;i++)
System.out.println((i+1)+" "+nm[i]+" "+um[i]);
}
```

- Hasil program :



```
C:\> Command Prompt
C:\j2\bin>javac Daftar.java
C:\j2\bin>java Daftar
No Nama Umur
1 Joko 4
2 Budi 5
3 Yeni 2
4 Nino 3
```

Array 2 dimensi (matrik)

- Array 2 dimensi didefinisikan dengan :
- TipeData [][] namaArray=new TipeData[n][m]
mendefinisikan array 2 dim n baris m kolom
- **Contoh :**

```
int [][]b= new int[3][4];
```

Mendefinisikan array int 3 baris 4 kolom

b[0][1]	int	int	int	int
b[1][1]	int	int	int	int
b[2][1]	int	int	int	int

Contoh : constanta array String 2 dimensi

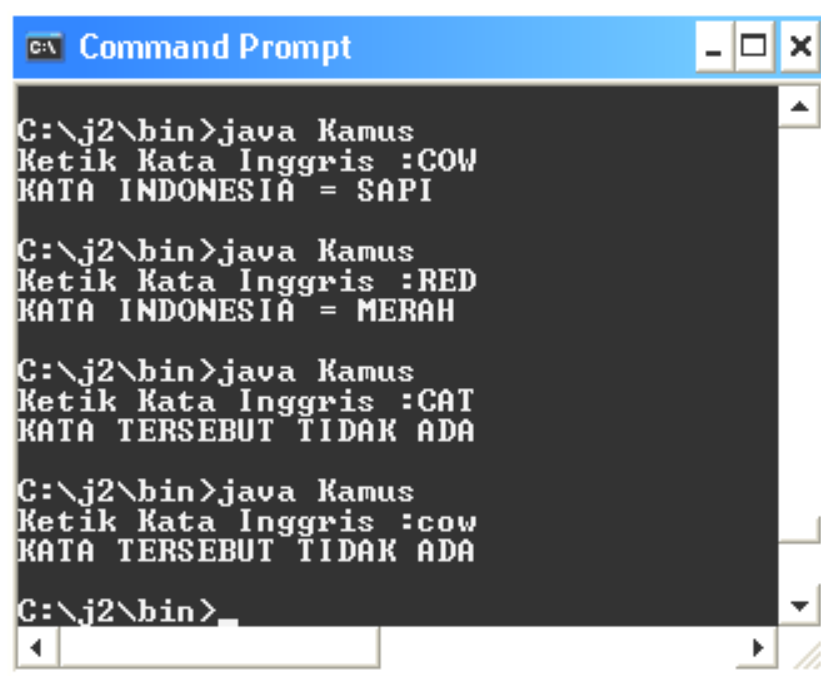
```
String[][] Kata={ {"ONE",  
"SATU"}, {"COW", "SAPI"}, {"RUN",  
"LARI"}, {"RED", "MERAH"}, {"MAN",  
"ORANG"}};
```

	Kolom 0	kolom 1
baris 0	ONE	SATU
baris 1	COW	SAPI
baris 0	RUN	LARI
baris 1	RED	MERAH
baris 0	MEN	ORANG

Contoh aplikasi array 2 dimensi untuk program kamus.

```
import java.io.*;
class Kamus
{ public static void main(String args[])throws Exception
  { String[][] Kata={ {"ONE", "SATU"},
    {"COW", "SAPI"}, {"RUN", "LARI"},
    {"RED", "MERAH"}, {"MAN", "ORANG"} };
    String KataDicari;
    int i, k;
    KataDicari=bacaString("Ketik Kata Inggris :");
    //cari kata
    k=-1;
    for (i=0; i<5; i++)
      if (Kata[i][0].equals(KataDicari) ) k=i;
    //cek ketemu atau tidak , jika k=-1 tidak ketemu
    if (k==(-1))
      System.out.println("KATA TERSEBUT TIDAK ADA");
    else
      System.out.println("KATA INDONESIA = "+Kata[k][1]);
  }
  static String bacaString(String s)throws Exception
  {System.out.print(s);
    int n=0; String hasil="";
    while (n!=13){n=System.in.read();
      | if(n!=13)hasil+=(char)n;}
    return hasil;
  }
}
```

Hasil program kamus



```
C:\> Command Prompt
C:\j2\bin>java Kamus
Ketik Kata Inggris :COW
KATA INDONESIA = SAPI

C:\j2\bin>java Kamus
Ketik Kata Inggris :RED
KATA INDONESIA = MERAH

C:\j2\bin>java Kamus
Ketik Kata Inggris :CAT
KATA TERSEBUT TIDAK ADA

C:\j2\bin>java Kamus
Ketik Kata Inggris :cow
KATA TERSEBUT TIDAK ADA

C:\j2\bin>
```

- Hasil kamus jika diinputkan kata dengan huruf kecil ternyata hasilnya masih :
KATA TERSEBUT TIDAK ADA
- Hal ini karena pencocokan string adalah apa adanya dengan yang ada dalam array
- Untuk mengatasi itu pencocokan dapat dilakukan dengan masing-masing string dirubah menjadi huruf kecil semua

Rangkuman

- Perulangan terdiri dari dua jenis : terhingga dan takhingga
- Perulangan jenis terhingga dikendalikan dengan for
- Perulangan jenis tak hingga menggunakan kendali while
- Kondisi umum menggunakan kendali if
- Kondisi untuk data data diskrit menggunakan kendali switch
- Data sejenis yang berjumlah banyak dan akan diakses secara massal dapat disimpan dalam array
- Array untuk satu deretan data disebut dengan array 1 dimensi : didefinisikan dengan `Tipe_Data [] NamVar=new Tipe_data[ukuran]`
- Array dua dimensi didefinisikan dengan `Tipe_Data [] [] NamVar=new Tipe_data[ukuran][ukuran]`

Latihan :

- Ubahlah program Kamus seperti yang dijelaskan dalam contoh menjadi program kamus yang dapat dieksekusi terus tanpa setiap kali eksekusi harus dengan > java Kamus, tetapi dialognya

>java Kamus <enter>

Ketik kata inggris : COW <enter>

KATA INDONESIA = SAPI

Mau main lagi <Y/T> ?:

Jika dijawab Y atau y maka akan ditanya lagi kata inggris , jika dijawab T atau t program selesai.

- Program juga harus dapat menerima input dalam huruf kecil dengan jawaban yang benar.