

PERINTAH MASUKAN

PERTEMUAN 3

a).Scanner

Class Scanner digunakan secara ekstensif untuk memasukan data dari keyboard. Tidak seperti perintah masukan lainnya, data-data yang dimasukan misalnya berupa angka, tidak perlu dilakukan konversi dari string ke integer atau tipe data lainnya. Untuk menggunakan Class Scanner, harus menyertakan package java.util.

contoh penggunaan Scanner

```
3 import java.util.Scanner;
4 /**
5  *
6  * @author rony
7  */
8 public class LatScanner {
9     public static void main(String args[])
10 {
11     //membuat objek baru
12     Scanner input = new Scanner(System.in);
13
14     //deklarasi variabel
15
16     String nama;
17     int n2;
18     double n1, n3;
19
20     System.out.print("Masukan Nama Anda : ");
21     nama = input.nextLine();
22     System.out.print("Masukan Nilai 1 : ");
23     n1 = input.nextDouble();
24     System.out.print("Masukan Nilai 2 : ");
25     n2 = input.nextInt();
26
27     n3 = n1 + n2;
28     System.out.println("\nNama Anda : " + nama);
29
30     System.out.println("Nilai Anda : " + n3);
31 }
32 }
```

contoh penggunaan Scanner(lanjutan)

```
2 import java.util.Scanner;
3 /**
4  *
5  * @author rony
6  */
7 public class ClassScanner {
8     public String nama;
9     public double n1;
10    public int n2,n3;
11    Scanner input = new Scanner(System.in);
12
13    public String getnama(){
14        return nama;
15    }
16
17    public void inputScanner(){
18
19        System.out.print("Masukan Nama Anda : ");
20        nama = input.nextLine();
21        System.out.print("Masukan Nilai 1 : ");
22        n1 = input.nextDouble();
23        System.out.print("Masukan Nilai 2 : ");
24        n2 = input.nextInt();
25    }
26
27    public double rata(){
28
29        return ((n1 + n2)/2);
30    }
31 }
```

Berikut penggunaan Scanner dan pembuatan Class yang digunakan di dalam main Class, untuk masukan data dari keyboard.

1.buatlah sebuah Class dengan nama **ClassScanner.java**

contoh penggunaan Scanner(lanjutan)

```
1 import java.util.Scanner;
2
3 /**
4  *
5  * @author rony
6  */
7
8 public class AksiScanner {
9     public static void main(String args[])
10
11 {
12     //membuat objek baru
13     ClassScanner scan=new ClassScanner();
14
15     scan.inputScanner();
16     System.out.println("\nNama Anda : " + scan.getnama());
17     System.out.println("Nilai Anda : " + scan.rata());
18
19 }
20
21 }
22
```

Kode program
ClassScanner yang telah di
buat diatas adalah sebuah
class yang kan digunakan
didalam main Class sesuai
dengan konsep dari
pemrograman berorientasi
Objek
kemudian buatlah sebuah
main class dengan nama
AksiScanner

contoh penggunaan Scanner(lanjutan)

Sedangkan pada kode program AksiProgram.Java merupakan sebuah main class yang dalam penulisan kode programnya hanya perlu memanggil method yang telah dibuat di dalam ClassScanner

Berikut output dari kode program AksiScanner.java

```
Output - LatihanJava (run)
run:
Masukan Nama Anda : Rony
Masukan Nilai 1 : 90
Masukan Nilai 2 : 80

Nama Anda :Rony
Nilai Anda : 85.0
BUILD SUCCESSFUL (total time: 11 seconds)
```

b.)BufferedReader

```
1 import java.io.*;
2 /**
3  *
4  * @author rony
5  */
6 public class LatBuffered {
7     public static void main(String args[]) throws Exception
8     {
9         //membuat objek baru
10        InputStreamReader keyreader = new InputStreamReader(System.in);
11        BufferedReader input = new BufferedReader(keyreader);
12        //deklarasi variabel
13        String kata1, kata2;
14
15        System.out.print("Masukan Kata Pertama : ");
16        kata1 = input.readLine();
17        System.out.print("Masukan Kata Kedua : ");
18        kata2 = input.readLine();
19        System.out.println("\nHasil Input String " + kata1 + " " + kata2);
20    }
21 }
22
23
24
25
```

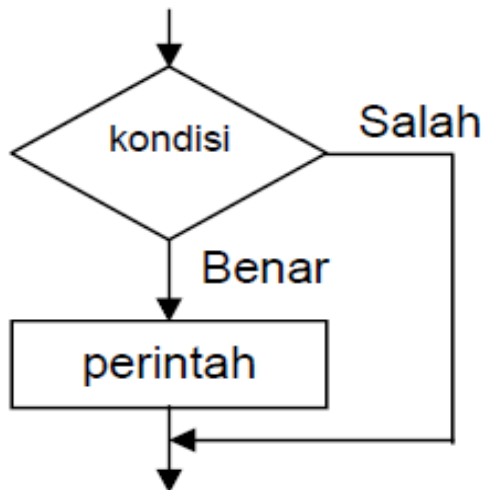
InputStreamReader digunakan membaca arus byte stream dan mengkonversi byte-byte ke dalam nilai-nilai bilangan bulat yang merepresentasikan karakter-karakter Unicode.

Kelas BufferedReader membaca masukan Stream karakter dan penyangga tersebut untuk efisiensi.

PENYELEKSIAN KONDISI PEMROGRAMAN JAVA

A. Pernyataan If

Pernyataan if mempunyai pengertian, “ Jika kondisi bernilai benar, maka perintah akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan”. Dari pengertian tersebut dapat dilihat dari diagram alir berikut:



A. Pernyataan If (lanjutan)

Penulisan kondisi harus didalam tanda kurung dan berupa ekspresi relasi dan penulisan pernyataan dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong. Jika pemakaian if diikuti dengan pernyataan majemuk, bentuk penulisannya sebagai berikut :

```
if (kondisi)  
pernyataan;
```

Jika lebih dari satu pernyataan hasu diapit dengan tanda kurung kurawal

```
if (kondisi)  
{  
pernyataan;  
.....  
}
```

Contoh dalam Kode Program

```
1 import java.util.*;
2 /**
3  *
4  * @author rony
5  */
6 public class LatIf
7 {
8     public static void main(String[] args)
9     {
10         double tot_beli, potongan=0, jum_bayar=0;
11         Scanner input = new Scanner(System.in);
12
13         System.out.print("Total Pembelian Rp. ");
14         tot_beli = input.nextDouble();
15         if (tot_beli >= 50000)
16             potongan = 0.2 * tot_beli;
17         System.out.println("Besarnya Potongan Rp. " + potongan);
18         jum_bayar = tot_beli - potongan;
19         System.out.println("Jumlah yang harus dibayarkan Rp. " + jum_bayar);
20     }
21 }
22
23
24
25
26
27
28
```

Contoh dalam Kode Program(lanjutan)

```
2 package KondisiIf;
3
4 /**
5  *
6  * @author rony
7  */
8 public class IfClass {
9     public double TotBeli,potongan;
10
11     public void setTotalBeli(double a){
12         TotBeli=a;
13     }
14
15     public double getPotongan(){
16         if (TotBeli >= 50000){
17             potongan = 0.2 * TotBeli;
18         }
19         return potongan;
20     }
21
22     public double JumlahBayar()
23     {
24         return (TotBeli-potongan);
25     }
26 }
```

Sekarang kita pahami lagi konsep OOP dengan memisahkan kode program menjadi beberapa class dengan method yang ada pada class yang di buat. Buatlah sebuah class dengan nama IfClass.java dengan kode program seperti di bawah ini

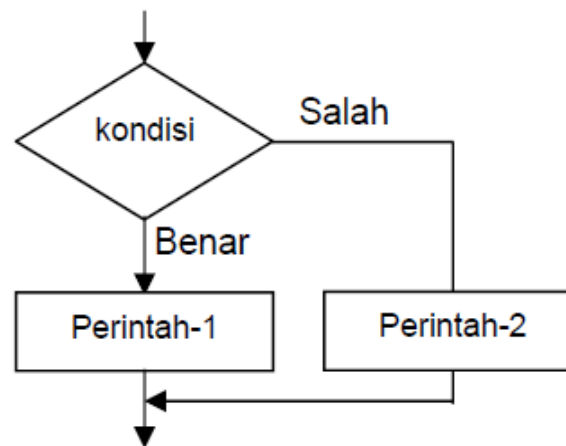
Contoh dalam Kode Program(lanjutan)

```
1
2 package KondisiIf;
3
4 import java.util.Scanner;
5
6 /**
7  *
8  * @author rony
9  */
10 public class IfClassAksi {
11     public static void main(String[] args)
12     {
13         double tot beli;
14         Scanner input = new Scanner(System.in);
15         IfClass fungsiif=new IfClass();
16
17         System.out.print("Total Pembelian Rp. ");
18         fungsiif.TotBeli = input.nextDouble();
19
20         System.out.println("Besarnya Potongan Rp. " + fungsiif.getPotongan());
21         System.out.println("Jumlah yang harus dibayarkan Rp. " +fungsiif.JumlahBayar());
22     }
23 }
```

Kemudian buatlah sebuah class sebagai class utama dengan nama IfClassAksi.java dengan kode program di bawah ini

B. Pernyataan IF-ELSE

Pernyataan if-else mempunyai pengertian, “ Jika kondisi bernilai benar, maka perintah-1 akan dikerjakan dan jika tidak memenuhi syarat maka akan mengerjakan perintah-2”. Dari pengertian tersebut dapat dilihat dari diagram alir berikut :



Pernyataan IF-ELSE (lanjutan)

```
if (kondisi)  
pernyataan-1;  
else  
pernyataan-1;
```

Perintah-1 dan perintah-2 dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong. Jika pemakaian if-else diikuti dengan pernyataan majemuk, bentuk penulisannya sebagai berikut:

```
if (kondisi)  
{  
perintah-1;  
...  
}  
else  
{  
perintah-2;  
}
```

Jika lebih dari satu pernyataan harus diapit dengan tanda kurung kurawal.

Contoh Kode Program IF-Else

```
3
4 import java.util.Scanner;
5
6 /**
7  *
8  * @author rony
9  */
10 public class LatIfElse {
11     public static void main(String[] args) {
12         Scanner input = new Scanner(System.in);
13         String keterangan,nama;
14         int nilai;
15
16         System.out.print("Masukan Nama Siswa:");
17         nama = input.nextLine();
18         System.out.print("Masukan Nilai Akhir :");
19         nilai = input.nextInt();
20         if(nilai > 70){
21             // perintah jika kondisi true
22             keterangan="lulus";
23         }else{
24             // perintah jika kondisi false
25             keterangan="Gagal";
26         }
27         System.out.println("Hasil Akhir");
28         System.out.println("=====");
29         System.out.println("Nama Siswa                :"+nama);
30         System.out.println("Nilai Akhir Yang di dapat :"+nilai);
31         System.out.println("Keterangan                :"+keterangan);
32         System.out.println("=====");
33     }
34 }
```


Contoh Kode Program IF-Else(lanjutan)

```
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author rony
8  */
9 public class IfElseClass {
10     public String nama,ket;
11     public int nilAakhir;
12     Scanner input = new Scanner(System.in);
13
14     public void setInputData(){
15         System.out.print("Masukan Nama Siswa:");
16         nama = input.nextLine();
17         System.out.print("Masukan Nilai Akhir :");
18         nilAakhir = input.nextInt();
19     }
20     public String getKeterangan(){
21         if(nilAakhir > 70){
22             // perintah jika kondisi true
23             ket="lulus";
24         }else{
25             // perintah jika kondisi false
26             ket="Gagal";
27         }
28         return ket;
29     }
30
31 }
32
```

Sekarang mari kita rubah kode program terstruktur diatas dengan konsep class dan main class. Buatlah sebuah class dengan nama IfElseClass.java dengan kode program sebagai berikut

Contoh Kode Program IF-Else(lanjutan)

```
3
4 /**
5  *
6  * @author rony
7  */
8 public class IfElseAksi {
9     public static void main(String[] args) {
10
11         //pembuatan objek baru dari class IfElseClass.java
12         IfElseClass ifclass=new IfElseClass();
13
14         ifclass.setInputData();
15         ifclass.getKeterangan();
16         System.out.println("Hasil Akhir");
17         System.out.println("=====");
18         System.out.println("Nama Siswa           :"+ifclass.nama);
19         System.out.println("Nilai Akhir Yang di dapat :"+ifclass.nilaiAkhir);
20         System.out.println("Keterangan           :"+ifclass.getKeterangan());
21         System.out.println("=====");
22     }
23 }
24
```

Setelah IfElseClass di buat maka kita akan membuat main class dengan nama IfElseAksi.java dengan kode program sebagai berikut.

If-Else Majemuk

Bentuk dari if-else bertingkat sebenarnya mirip dengan nested if, keuntungan penggunaan if- else bertingkat dibanding dengan nested if adalah penggunaan bentuk penulisan yang lebih sederhana.

Bentuk Penulisannya

```
if (syarat)
{
... perintah;
... perintah;
}
else if (syarat)
{
... perintah;
... perintah;
}
else
{
... perintah;
... perintah;
}
```

Contoh If-Else Majemuk

Berikut contoh penggunaan pernyataan if – else majemuk, bisa anda ikuti seperti dibawah ini.

```
3 import java.util.Scanner;
4 /**
5  *
6  * @author rony
7  */
8 public class IfElseMajemuk {
9     public static void main(String[] args){
10         int pendapatan;
11         double jasa, komisi, total;
12         //deklarasi objek
13         Scanner input = new Scanner(System.in);
14         System.out.print("Masukan Pendapatan Sales Rp. ");
15         pendapatan = input.nextInt();
16
17         if (pendapatan >= 0 && pendapatan <= 200000) {
18             jasa=10000;
19             komisi=0.1*pendapatan;
20         }
21         else if(pendapatan<=500000) {
22             jasa=20000;
23             komisi=0.15*pendapatan;
24         }
25         else {
26             jasa=30000;
27             komisi=0.2*pendapatan;
28         }
29         /* menghitung total */
30         total = komisi+jasa;
31         System.out.println("\nUang JasaRp. " + (int) jasa);
32         System.out.println("Uang Komisi Rp. " + (int) komisi);
33         System.out.println("=====");
34         System.out.println("Uang Total Rp. " + (int) total);
35     }
36 }
```

Pernyataan switch - case

Bentuk dari switch - case merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternatif.

```
switch (ekspresi integer)
{
case konstanta-1 :
... perintah;
... perintah;
break;
case konstanta-2 :
... perintah;
... perintah;
break;
.....
.....
default :
... perintah;
... perintah;
}
```

Pernyataan switch -case ini memiliki kegunaan sama seperti if – else bertingkat, tetapi penggunaannya hanya untuk memeriksa data yang bertipe primitif integer saja. Bentuk penulisan perintah ini sebagai berikut :

Contoh Program switch - case

Setiap cabang akan dijalankan jika syarat nilai konstanta tersebut dipenuhi dan default akan dijalankan jika semua cabang di atasnya tidak terpenuhi.

Contoh :

```
import java.util.*;
class LatScan4
{
    public static void main(String[] args)
    {
        int kd;
        String nb="";
        Scanner input=new Scanner(System.in);
        System.out.print("masukan kode barang[1..3]:");
        kd=input.nextInt();

        switch(kd)
        {
            case 1:
                nb="Alat Olah raga";
                break;
            case 2:
                nb="Alat Elektronik";
                break;
            case 3:
                nb="Alat Alat Masak";
                break;
            default:
                nb=" Anda Salah Kode !!!";
                break;
        }
        System.out.println("\nNama Barang :"+nb);
    }
}
```

Pernyataan break menunjukkan bahwa perintah siap keluar dari switch. Jika pernyataan ini tidak ada, maka program akan diteruskan ke cabang – cabang yang lainnya.

Berikut contoh penggunaan pernyataan switch - case, bisa anda ikuti seperti dibawah samping.