

Pertemuan 3 dan 4.

I/O Stream, Percabangan, Perulangan dan Konversi Tipe Data pada Java.

A. Penggunaan Input Output Stream

Perintah Masukan

Untuk melakukan perintah masukan, kita akan menggunakan 3 (tiga) cara, yaitu `InputStream`, `BufferedInputStream` dan `Scanner`.

1) `InputStream`

`InputStream` adalah subclass `Object`, yang menjadi landasan untuk class-class yang biasa digunakan untuk membaca data dan menampilkan kelayar. Untuk penggunaan `InputStream` ini, harus menyertakan package **`java.io`**.

Berikut penggunaan `InputStream`, untuk masukan data dari keyboard.

```
masukan_inputstream.java *
import java.io.*;
class masukan_inputstream
{
    public static void main(String[] args)
    {
        String kata="";
        boolean akhir=false;
        int huruf;
        System.out.print("Masukan kata-kata anda:");
        while(!akhir)
        {
            try
            {
                huruf=System.in.read();
                if(huruf < -1 || huruf == '\n')
                {
                    akhir = true;
                    kata = kata + (char) huruf;
                }
            }
            catch (IOException e)
            {
                System.err.println("Mengalami Salah?");
                akhir=true;
            }
        }
        System.out.println("Kalimat yang anda ketikan = " + kata);
    }
}
```

Hasilnya :

```
General Output
-----Configuration: <Default>-----
Masukan kata-kata anda:STMIK Nusa Mandiri
Kalimat yang anda ketikan = STMIK Nusa Mandiri

Process completed.
```

2) InputStreamReader dan BufferedReader

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. Pada penggunaannya harus mempunyai Reader Object untuk membuat versi buffered. Berikut konstrutor yang digunakan untuk membuat BufferedReader.

- `BufferedReader(Reader)` digunakan untuk membuat penyangga karakter stream yang berhubungan dengan Reader Objek yang ditetapkan.
- `BufferedReader (Reader, int)` digunakan untuk membuat penyangga karakter stream yang berhubungan dengan Reader Objek yang ditetapkan dan dengan penyangga ukuran integer.

Penyangga Karakter Stream dapat dibaca menggunakan metoda `read()` dan `read(char[], int, int)` untuk menguraikan `FileReader`, serta dapat juga membaca baris dari teks dengan menggunakan `readLine ()`.

metoda `readLine ()`, digunakan untuk kembalikan String objek yang berisi next line dari teks pada stream, tidak termasuk karakter atau karakter-karakter yang merepresentasikan end-of-line. Jika akhir stream dicapai, maka nilai string yang ingin dikembalikan bernilai dengan null. end-of-line ditandai dengan beberapa pernyataan berikut :

- newline karakter (`'\n'`)
- carriage return character (`'\r'`)
- carriage return character yang diikuti oleh satu newline (`"\n\r"`)

Berikut penggunaan `InputStreamReader` dan `Buffered`, untuk masukan data dari keyboard.

```

inputbuffer_reader.java *
import java.io.*;
class inputbuffer_reader
{
    public static void main(String args[]) throws Exception
    {
        //membuat objek baru
        InputStreamReader keyreader = new InputStreamReader(System.in);
        BufferedReader input = new BufferedReader(keyreader);

        //deklarasi variabel
        String s1, s2;
        System.out.print("Masukan String Pertama : ");
        s1 = input.readLine();
        System.out.print("Masukan String Kedua : ");
        s2 = input.readLine();
        System.out.println("\nHasil Input String " + s1 + " " + s2);
    }
}

```

Hasilnya :

```

General Output
Masukan String Pertama : Nusa Mandiri
Masukan String Kedua : Family

Hasil Input String Nusa Mandiri Family

Process completed.

```

3) 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**.

Untuk membaca baris dari text yang diinputkan dengan menggunakan metoda `next()`. Supaya string yang dibaca utuh termasuk spasi, dengan menggunakan `nextLine()`. Untuk membaca data berupa nilai integer atau tipe data angka lainnya, sebagai contoh bisa menggunakan metoda, seperti dibawah ini :

- `nextInt()` untuk membaca nilai integer
- `nextDouble()` untuk membaca nilai double.
- `nextFloat()` untuk membaca nilai float.

Berikut penggunaan Scanner, untuk masukan data dari keyboard.

```
scanner.java
import java.util.Scanner;
class scanner
{
    public static void main(String args[])
    {
        //membuat objek baru
        Scanner input = new Scanner(System.in);
        //deklarasi variabel
        String nama;
        int n2;
        double n1, n3;
        System.out.print("Masukan Nama Anda : ");
        nama = input.nextLine();
        System.out.print("Masukan Nilai 1 : ");
        n1 = input.nextDouble();
        System.out.print("Masukan Nilai 2 : ");
        n2 = input.nextInt();
        n3 = n1 + n2;
        System.out.println("\nNama Anda : " + nama);
        System.out.println("Nilai Anda : " + n3);
    }
}
```

Hasilnya :

```
General Output
-----Configuration: <Default>-----
Masukan Nama Anda : Nurmalasari
Masukan Nilai 1 : 80
Masukan Nilai 2 : 90

Nama Anda :Nurmalasari
Nilai Anda : 170.0

Process completed.
```

Perintah Keluaran

Pada saat melakukan aktivitas dasar pemrograman, pasti tidaklah terlepas dari menampilkan data atau hasil kelayar. Dalam hal ini kita butuh sebuah standard output yaitu stream yang digunakan untuk mengirimkan keluaran kelayar. Stream tersebut yaitu System.out. Pada saat kita

akan menampilkan kelayar, kita butuh method print atau println. Perbedaan antara print dan println yaitu :

- `System.out.print(Statement);` , maka informasi yang ditampilkan dilayar tidak pindah baris.
- `System.out.println(Statement);` , maka informasi yang ditampilkan dilayar pindah baris (line new). Statement bisa berupa pesan dan argument atau variabel. Statement biasanya diapit dengan tanda kutip ganda (" "), untuk memisahkan antara statement dan argumen atau variabel dipisah dengan tanda plus (+).

Berikut contoh program sederhana perintah keluaran.

```

Output.java
class Output
{
public static void main(String[] args)
{
// deklarasi variabel
int a, b, c;
// memberikan nilai
a = 3;
b = 5;
// proses
c = a + b;
// cetak variabel
System.out.println("Hasil Perhitungan");
System.out.print("Hasilnya adalah = " + a);
System.out.print(" + " + b);
System.out.println(" = " + c);
}
}

```

Hasilnya :

```

General Output
-----Configuration: <Default>-----
Hasil Perhitungan
Hasilnya adalah = 3 + 5 = 8

Process completed.

```

B. Percabangan

- 1) **If – Else** : Pernyataan if mempunyai pengertian, “ Jika kondisi bernilai benar, maka perintah-1 akan dikerjakan dan jika tidak memenuhi syarat maka akan mengerjakan perintah-2”.

```

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

```

Contoh penggunaan if-else :

```

if_else.java *
import java.util.*;
class if_else
{
    public static void main(String[] args)
    {
        double tot_beli, potongan=0, jum_bayar=0;
        Scanner input = new Scanner(System.in);
        System.out.print("Total Pembelian Rp. ");
        tot_beli = input.nextDouble();
        if (tot_beli >= 50000)
            potongan = 0.2 * tot_beli;
        else
            potongan = 0.05 * tot_beli;
        System.out.println("Besarnya Potongan Rp. " + potongan);
        jum_bayar = tot_beli - potongan;
        System.out.println("Jumlah yang harus dibayarkan Rp. " + jum_bayar);
    }
}

```

Hasilnya :

```

General Output
-----Configuration: <Default>-----
Total Pembelian Rp. 50000
Besarnya Potongan Rp. 10000.0
Jumlah yang harus dibayarkan Rp. 40000.0

Process completed.

```

- 2) **Nested If** : Nested if merupakan pernyataan if berada didalam pernyataan if yang lainnya. Bentuk penulisan pernyataan Nested if adalah :

```

if(syarat)
    if(syarat)
        ... perintah;
    else
        ... perintah;
else
    if(syarat)
        ... perintah;
    else
        ... perintah;

```

Contoh (nama file : nestedif)

```

import java.util.*;
class nestedif
{
    public static void main(String[] args)
    {
        int pendapatan;
        double jasa, komisi, total;
        // deklarasi objek
        Scanner input = new Scanner(System.in);
        System.out.print("Masukan pendapatan sales Rp.");
        pendapatan=input.nextInt();
        if (pendapatan>=0 && pendapatan <=200000)
        {
            jasa=10000;
            komisi=0.1*pendapatan;
        }
        else{
            if(pendapatan<=500000)
            {
                jasa=20000;
                komisi=0.15*pendapatan;
            }
            else
            {
                jasa=30000;
                komisi=0.2*pendapatan;
            }
        }
    }
}

```

```

    }
    }
    //menghitung total
    total=komisi+jasa;

    System.out.println("\n Uang jasa Rp." +(int)jasa);
    System.out.println("Uang Komisi Rp. " +(int) komisi);
    System.out.println("=====");
    System.out.println("Uang Total Tp. " +(int) total);
}
}

```

Hasilnya:

```

General Output
-----Configuration: <Default>-----
Masukan pendapatan sales Rp.25000

    Uang jasa Rp.10000
    Uang Komisi Rp. 2500
=====
Uang Total Tp. 12500

Process completed.

```

3) **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.

```

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

```


Contoh :

```
import java.util.*;
class ifMajemuk
{
    public static void main(String[] args)
    {
        int pendapatan;
        double jasa, komisi, total;
        //deklarasi objek
        Scanner input = new Scanner(System.in);
        System.out.print("Masukan Pendapatan Sales Rp. ");
        pendapatan = input.nextInt();
        if (pendapatan >= 0 && pendapatan <= 200000)
        {
            jasa=10000;
            komisi=0.1*pendapatan;
        }
        else if(pendapatan<=500000)
        {
            jasa=20000;
            komisi=0.15*pendapatan;
        }
        else
        {
            jasa=30000;
            komisi=0.2*pendapatan;
        }

        total = komisi+jasa;
        System.out.println("\nUang Jasa Rp. " + (int) jasa);
        System.out.println("Uang Komisi Rp. " + (int) komisi);
        System.out.println("=====");
        System.out.println("Uang Total Rp. " + (int) total);
    }
}
```

Hasilnya :

```
General Output
-----Configuration: <Default>-----
Masukan Pendapatan Sales Rp. 100000

Uang Jasa Rp. 10000
Uang Komisi Rp. 10000
=====
Uang Total Rp. 20000

Process completed.
```

- 4) **Switch Case :** Bentuk dari switch - case merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternatif. 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 :

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

    .....

    default :
        ... perintah;
        ... perintah;
```

Contoh switch case :

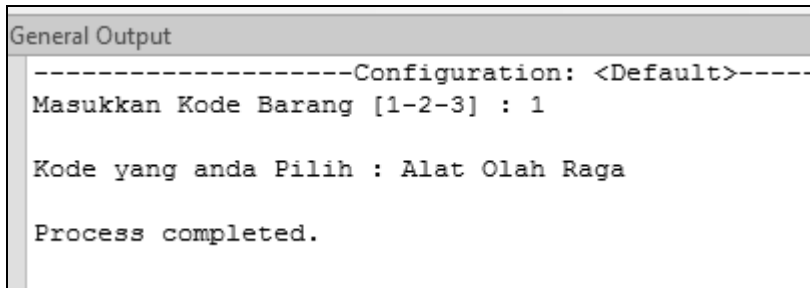
```
import java.util.*;
class SwitchCase
{
    public static void main(String[] args)
    {
        int kode;
        String nmbarang;
```

```

System.out.print("Masukkan Kode Barang [1-2-3] : ");
kode = input.nextInt();
switch(kode)
{
    case 1 :
        nmbarang = "Alat Olah Raga";
        break;
    case 2 :
        nmbarang = "Alat Elelctronik";
        break;
    case 3 :
        nmbarang = "Alat Masak";
        break;
    default:
        nmbarang = "Anda Salah Memasukan kode";
        break;
}
System.out.println("\nKode yang anda Pilih : " + nmbarang);
}
}

```

Hasilnya :



```

General Output
-----Configuration: <Default>-----
Masukkan Kode Barang [1-2-3] : 1

Kode yang anda Pilih : Alat Olah Raga

Process completed.

```

Latihan percabangan

1. Buatlah program untuk menghitung nilai rata-rata dari seorang siswa, dengan ketentuan sebagai berikut :
 - a. Nama Siswa, Nilai Pertandingan I, Nilai Pertandingan II, Nilai Pertandingan III diinput.
 - b. Nilai Rata-rata merupakan hasil dari Nilai Pertandingan I, II dan III dibagi dengan 3.
 - c. Ketentuan Mendapat Hadiah dari pertandingan :
 - Jika Nilai Rata-Rata ≥ 85 , maka mendapat hadiah Seperangkat Komputer P4

- Jika Nilai Rata-Rata ≥ 70 , maka mendapat hadiah Seperangkat Uang sebesar Rp.500,000
- Jika Nilai Rata-Rata < 70 , maka mendapat hadiah Hiburan

d. Tampilan yang diinginkan sebagai berikut :

Layar Masukan

PROGRAM HITUNG NILAI RATA-RATA

Nama Siswa : ... <diinput>

Nilai Pertandingan I : ... <diinput>

Nilai Pertandingan II : ... <diinput>

Nilai Pertandingan III : ... <diinput>

Layar Keluaran

Siswa yang bernama ... <tampil data>

Memperoleh nilai rata-rata <hasil proses> dari hasil perlombaan yang diikutinya.

Hadiah yang didapat adalah ... <hasil proses>

2. Buatlah program untuk menghitung total pembayaran dari sebuah penjualan agen susu di kota besar ini.. Dengan ketentuan sebagai berikut :

a. Jenis susu diinput diinput berdasarkan kode yang sudah ditentukan, yaitu :

- Jika kode A adalah Dancow
- Jika kode B adalah Bendera
- Jika kode A adalah SGM

b. Ukuran kaleng susu diinput berdasarkan kode yang sudah ditentukan.

- Jika kode 1 adalah Kecil
- Jika kode 2 adalah Sedang
- Jika kode 3 adalah Besar

c. Harga susu sesuai dengan jenis susu dan ukuran kaleng susu

JENIS SUSU	HARGA BERDASARKAN UKURAN KALENG SUSU		
	KECIL	SEDANG	BESAR
DANCOW	25000	20000	15000
BENDERA	20000	17500	13500
SGM	22000	18500	15000

d. Proses yang dilakukan untuk mendapatkan Total Pembayaran

Total Bayar = Harga Susu per ukuran dan Jenis dikali dengan banyak beli

- e. Tampilan Layar Masukkan dan Keluaran yang diinginkan sebagai berikut :

TOKO KELONTONG KERONCONGAN

A. Susu Dancow

1. Ukuran Kecil
2. Ukuran Sedang
3. Ukuran Besar

B. Susu Bendera

1. Ukuran Kecil
2. Ukuran Sedang
3. Ukuran Besar

C. Susu SGM

1. Ukuran Kecil
2. Ukuran Sedang
3. Ukuran Besar

Masukan Merk Susu [Dancow | Bendera | SGM] : < diinput >

Masukan Ukuran Kaleng [Kecil|Sedang|Besar] : < diinput >

Harga Satuan Barang Rp.< tampil harga satuan >

Jumlah Yang dibeli : ... < diinput >

Harga Yang Harus dibayar Sebesar Rp. <hasil proses>

C. Perulangan

1) Pernyataan perulangan for

Bentuk umum pernyataan perulangan for :

```
for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )  
{  
    pernyataan / perintah;  
    pernyataan / perintah;  
    pernyataan / perintah;  
}
```

Kegunaan dari masing-masing argumen for diatas adalah :

- ✓ Inisialisasi : merupakan bagian untuk memberikan nilai awal untuk variable-variabel tertentu.
- ✓ Syarat Pengulangan : memegang kontrol terhadap pengulangan, karena bagian ini yang akan menentukan suatu perulangan diteruskan atau dihentikan.
- ✓ Pengubah Nilai Pencacah : mengatur kenaikan atau penurunan nilai pencacah.

Contoh perulangan for:

Sebagai contoh program untuk mencetak bilangan dari 1 hingga 10 secara menaik, secara selengkapnya seperti dibawah ini:

```
for.java
class contoh_for
{
    public static void main (String[] args)
    {
        int a;
        for(a=1; a<=10; ++a)
            System.out.println(a);
    }
}
```

Hasilnya :

```
General Output
1
2
3
4
5
6
7
8
9
10
Process completed.
```

2) Pernyataan Nested for

Pernyataan Nested for adalah suatu perulangan for didalam perulangan for yang lainnya. Bentuk umum pernyataan Nested for sebagai berikut :

```
for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )
{
    for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )
    {
        pernyataan / perintah;
    }
}
```

Didalam penggunaan nested-for, perulangan yang didalam terlebih dahulu dihitung hingga selesai, kemudian perulangan yang diluar diselesaikan.

Contoh perulangan nested for:

```
nested_for.java *
class nested_for
{
    public static void main(String[] args)
    {
        int a, b;
        for(a=1; a<=5; a++)
        {
            System.out.println();
            for(b=a; b<=5; b++)
            {
                System.out.print(a);
            }
            System.out.println("\n");
        }
    }
}
```

Hasilnya :

```
General Output
-----
11111
2222
333
44
5

Process completed.
```

3) Pernyataan perulangan While dan Do While

Pernyataan While : Pernyataan perulangan while merupakan instruksi perulangan yang mirip dengan perulangan for.

Bentuk perulangan while dikendalikan oleh syarat tertentu, yaitu perulangan akan terus dilaksanakan selama syarat tersebut terpenuhi.

Bentuk umum perulangan while, sebagai berikut :

```
while ( syarat )
{
    Pernyataan / perintah ;
    Pernyataan / perintah ;
}
```

Contoh perulangan While :

```
perulangan_while.java
class perulangan_while
{
    public static void main(String[] args)
    {
        int bil=1;
        while (bil<=10)
        {
            System.out.println(bil);
            ++bil;
        }
    }
}
```

Hasilnya :

```
General Output
1
2
3
4
5
6
7
8
9
10
Process completed.
```

Pernyataan perulangan do-while

Pernyataan perulangan do - while merupakan bentuk perulangan yang melaksanakan perulangan terlebih dahulu dan pengujian perulangan dilakukan dibelakang.

Bentuk umum perulangan do - while, sebagai berikut :


```
do
{
    Pernyataan / perintah ;
    Pernyataan / perintah ;
}
while ( syarat );
```

Contoh do-while

```
perulangan_dowhile2.java
class perulangan_dowhile
{
    public static void main(String[] args)
    {
        int bil=1;
        do
        {
            System.out.println(bil);
            bil+=2;
        }
        while (bil<=10);
    }
}
```

Hasilnya :

```
General Output
1
3
5
7
9
Process completed.
```

Latihan perulangan

1. Buat program dengan perulangan for yaitu mencetak bilangan penurunan dari 10-1, dan untuk mencetak bilangan loncat dua angka secara menaik.
2. Buat program dengan prulangan do-while deret bilangan genap.
3. Buat program untuk menampilkan bilangan prima, sbb :

2, 3, 5, 7, 11, 13, 17, 19

Perintah Konversi Data

Bentuk data yang diinputkan melalui keyboard, secara umum berupa nilai string, maka dalam hal proses perhitungan matematika tidak bisa diproses, maka supaya bisa digunakan, harus dikonversi ke bentuk tipe data yang diinginkan.

1) Konversi String to Integer

Untuk melakukan konversi String ke Integer, dengan menggunakan bentuk penulisan seperti dibawah ini :

```
var_penampung = Integer.parseInt(nilai_string);
```

Untuk lebih jelasnya anda bisa lihat contoh dibawah ini :

```
StringToInteger.java *
import java.io.*;
class StringToInteger
{
    public static void main(String args[]) throws Exception
    {
        //membuat objek baru
        InputStreamReader keyreader = new InputStreamReader(System.in);
        BufferedReader input = new BufferedReader(keyreader);
        //deklarasi variabel
        String s1, s2;
        int n1, n2, n3;
        System.out.print("Masukan Nilai Pertama : ");
        s1 = input.readLine();
        System.out.print("Masukan Nilai Kedua : ");
        s2 = input.readLine();
        //konversi
        n1 = Integer.parseInt(s1);
        n2 = Integer.parseInt(s2);
        n3 = n1 + n2;
        System.out.println("\nHasil Hitung = " + n3);
    }
}
```

Hasilnya :

```
General Output
-----Configuration:
Masukan Nilai Pertama : 90
Masukan Nilai Kedua : 80

Hasil Hitung = 170

Process completed.
```

2) Konversi String to Float

Untuk melakukan konversi String ke Float, dengan menggunakan bentuk penulisan seperti dibawah ini :

```
var_penampung = Float.parseFloat(nilai_string);
```

Untuk lebih jelasnya anda bisa lihat contoh dibawah ini :

```
StringToFloat.java
import java.io.*;
class StringToFloat
{
    public static void main(String args[]) throws Exception
    {
        //membuat objek baru
        InputStreamReader keyreader = new InputStreamReader(System.in);
        BufferedReader input = new BufferedReader(keyreader);
        //deklarasi variabel
        String s1, s2;
        float n1, n2, n3;
        System.out.print("Masukan Nilai Pertama : ");
        s1 = input.readLine();
        System.out.print("Masukan Nilai Kedua : ");
        s2 = input.readLine();
        //konversi
        n1 = Float.parseFloat(s1);
        n2 = Float.parseFloat(s2);
        n3 = n1 + n2;
        System.out.println("\nHasil Hitung = " + n3);
    }
}
```

Hasilnya :

```
General Output
-----Configuration:
Masukan Nilai Pertama : 90
Masukan Nilai Kedua : 5

Hasil Hitung = 95.0

Process completed.
```

3) Konversi String to Double

Untuk melakukan konversi String ke Double, dengan menggunakan bentuk penulisan seperti dibawah ini :

```
var_penampung = Double.parseDouble(nilai_string);
```

Untuk lebih jelasnya anda bisa lihat contoh dibawah ini :

```
StringToDouble.java

import java.io.*;

class StringToDouble
{
    public static void main(String args[]) throws Exception
    {
        //membuat objek baru
        InputStreamReader keyreader = new InputStreamReader(System.in);
        BufferedReader input = new BufferedReader(keyreader);
        //deklarasi variabel
        String s1, s2;
        double n1, n2, n3;
        System.out.print("Masukan Nilai Pertama : ");
        s1 = input.readLine();
        System.out.print("Masukan Nilai Kedua : ");
        s2 = input.readLine();
        //konversi
        n1 = Double.parseDouble(s1);
        n2 = Double.parseDouble(s2);
        n3 = n1 + n2;
        System.out.println("\nHasil Hitung = " + n3);
    }
}
```

Hasilnya :

```
General Output

-----Configuration:
Masukan Nilai Pertama : 90
Masukan Nilai Kedua : 10

Hasil Hitung = 100.0

Process completed.
```

Latihan

Buatlah program menghitung Nilai Akhir Siswa Informatika. Dengan Ketentuan seperti dibawah ini :

- a. Nilai UTS, UAS dan Tugas Mandiri diinput melalui keyboard
- b. Menghitung Nilai Murni
 - ✓ Nilai Murni UTS = Nilai UTS dikali dengan 35%
 - ✓ Nilai Murni UAS = Nilai UAS dikali dengan 45%
 - ✓ Nilai Murni Tugas Mandiri = Nilai Tugas Mandiri dikali dengan 20%
- c. Nilai Akhir adalah perhitungan Nilai Murni - Nilai Murni
- d. Layar masukan dan keluaran seperti dibawah ini :

Masukan Nama Siswa : _____

Nilai UTS : _____

Nilai UAS : _____

Nilai Tugas Mandiri : _____

Nilai Murni yang diperoleh :

Nilai Murni UTS : _____

Nilai Murni UAS : _____

Nilai Murni Tugas : _____

Nilai Akhir yang diperoleh yaitu : _____