

# Pemrograman Dasar



PEMROGRAMAN TERSTRUKTUR  
INPUT / OUTPUT

# Pemrograman Terstruktur

2

- **Pemrograman Terstruktur** merupakan pola penyusunan program komputer hanya dengan menggunakan tiga struktur kontrol.
  - Sequence
  - Selection
  - Repetition

# Sequence

3

- **Sequence** merupakan urutan pengerjaan dari perintah/statement pertama sampai dengan perintah/statement terakhir.
- Umumnya bahasa pemrograman mempunyai *sequence* (urutan pengerjaan dari perintah / statement ) mulai dari atas ke bawah dan dari kiri ke kanan.
- Top-down

# Contoh Sequence

4

- Algoritma :

Cetak "Jumlah Mahasiswa"

Set Jumlah to 49

Cetak "Tambahan mahasiswa baru"

Baca mhs\_baru

$\text{Jumlah} = \text{Jumlah} + \text{mhs\_baru}$

Cetak "Jumlah Mahasiswa"

Cetak jumlah

- Penjelasan :

- Urutan pengerjaan dimulai dari urutan pertama sampai dengan urutan terakhir.
- Jika mhs\_baru diisi dengan 2, maka jumlah yang tercetak adalah 51.

# Selection

5

- Struktur Kontrol **Selection** adalah penggambaran sebuah kondisi dan pilihan diantara dua aksi.
- Statement Pertama akan dikerjakan jika kondisi bernilai benar, jika tidak maka akan mengerjakan perintah setelah *keyword* “else” (jika ada)

# Contoh Selection

6

- Algoritma :  
    IF Hari=1 THEN  
        Cetak “Senin”  
    ELSE  
        Cetak “Bukan hari Senin”  
    ENDIF
- Penjelasan :
  - Tulisan “Senin” akan ditampilkan jika Hari bernilai 1, jika tidak maka tulisan “Bukan hari Senin” yang akan ditampilkan

# Repetition

7

- Beberapa *statement* / perintah dapat diulang dengan menggunakan struktur kontrol ***repetition***.
- *Statement* / perintah akan tetap diulang selama kondisi perulangan memenuhi (jika menggunakan *statement* WHILE)

# Contoh Repetition

8

- Algoritma :  
    Bintang = 0  
    WHILE bintang < 5  
        Cetak bintang  
        bintang = bintang + 1  
    END
- Penjelasan :
  - Pertama kali bintang akan diisi dengan 0, setelah itu isi dari bintang akan dicetak sebanyak lima kali, sehingga tampilannya akan sebagai berikut:  
0 1 2 3 4



# I/O

9

- Operasi I/O sangat kompleks, melibatkan ***software level*** dan ***electronic level***.
- class **java.util.Scanner** digunakan untuk menjalankan fungsi input
- **Scanner** BUKAN salah satu bagian dasar bahasa pemrograman Java, namun bagian dari package, **java.util**, yang dapat di-include-kan ke dalam program kita
- **Package** adalah collection dari classes yang digunakan dalam program kita

# I/O Streams

10



- Sumber dari input data disebut sebagai **input stream** dan output data disebut sebagai **output stream**
- Data input disebut **reading** data, dan data output disebut **writing** data

# Standard Streams

11

- Terdapat 3 standard I/O streams:
  - `System.in` —stream input .
  - `System.out` —stream output untuk hasil normal.
  - `System.err` — stream output untuk pesan error.
- **System.in** terhubung dengan keyboard dan kumpulan data characters.
- **System.out** dan **System.err** terhubung ke monitor, dan juga mengandung data character.

- Apa yang keyboard kirimkan ke program ketika menuliskan angka di bawah ini:

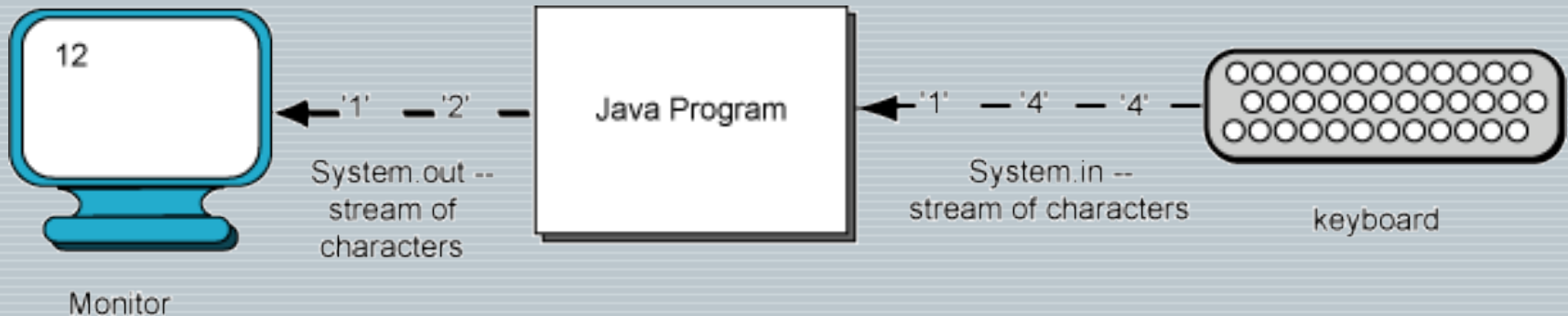
1234

- Jawab:  
characters '1' , '2' , '3' , dan '4' .

# Characters In, Characters Out

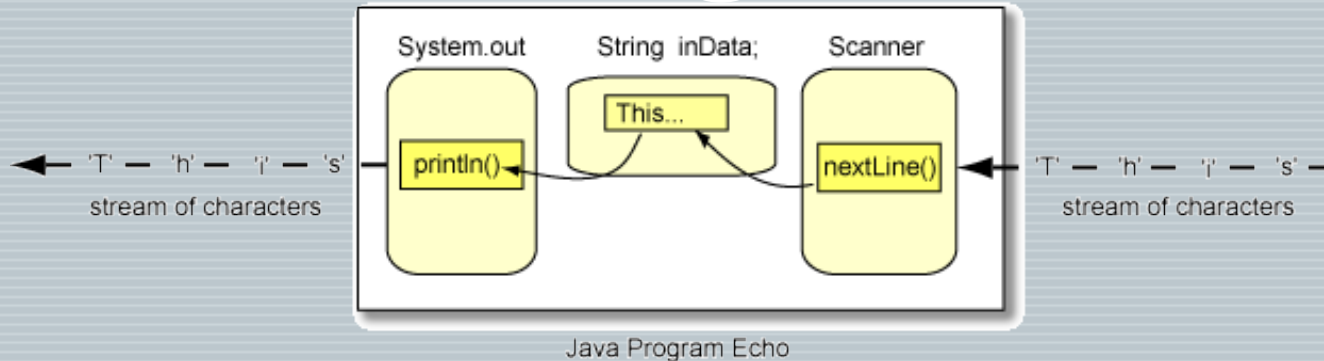
13

- keyboard mengirimkan data **character** ke komputer, bahkan dalam bentuk angka.
- Program mengirimkan characters ke monitor, bahkan dengan dalam bentuk hasil perhitungan numerik.



# Example I/O Program

14



- Method **nextLine()** dari **Scanner** membaca data baris character dari keyboard.
- characters dimasukkan ke dalam object **String**
- assignment statement menempatkan data ke dalam variabel **inData**.
- Menampilkan output ke monitor, program menggunakan method **println()** dari **System.out**

# Echo.java

15

```
import java.util.Scanner;

class Echo
{
    public static void main (String[] args)
    {
        String inData;
        Scanner scan = new Scanner( System.in );

        System.out.println("Enter the data:");
        inData = scan.nextLine();

        System.out.println("You entered:" + inData );
    }
}
```

```
Enter the data:
This is what the user enters.
You entered:This is what the user enters.
```

# Digits as Input

16

- Mengkoleksi characters dari keyboard dilakukan oleh operating system.
- Selama proses koleksi karakter, program Java berhenti.
- user dapat mengubah input, dan menekan "enter" yang mengisyaratkan input selesai.
- Contoh Output Program:

Enter the data: Columbus sailed in 1492.

You entered:Columbus sailed in 1492.

Enter the data: 1492

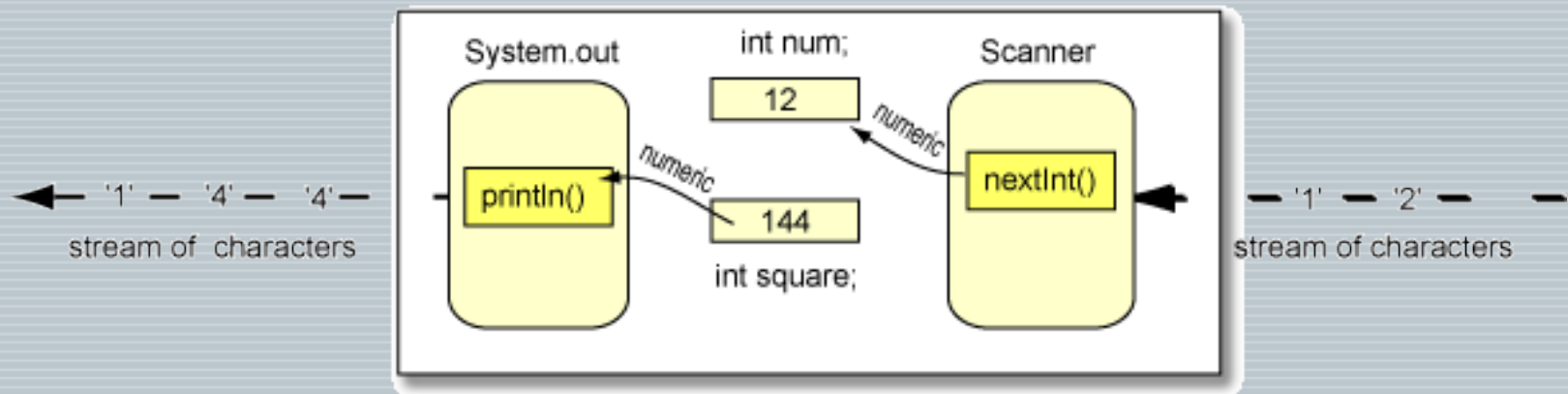
You entered:1492



# nextInt()

17

- Method **nextInt()** dari object Scanner membaca string dari digits (characters) dan mengkonversinya ke tipe data **int**



EchoSquare Java Program

- Dapatkah operasi aritmatik dilakukan dengan strings dari characters?

# EchoSquare.java

18

```
import java.util.Scanner;

class EchoSquare
{
    public static void main (String[] args)
    {
        Scanner scan = new Scanner( System.in );
        int num, square;      // declare two int variables

        System.out.println("Enter an integer:");
        num = scan.nextInt();
        square = num * num ;  // compute the square

        System.out.println("The square of " + num + " is " + square);
    }
}
```

```
Enter an integer:
12
The square of 12 is 144
```

- Apakah input di bawah ini dapat berjalan?
  - twelve hundred

# Converting to Integers

19

```
num = scan.nextInt();
```

- Assignment statements bekerja dalam 2 langkah:
  - Mengevaluasi expression pada bagian kanan "=",
  - Meletakkan nilai pada variabel di sebelah kiri "=".
- Manakah input di bawah ini yang BENAR?
  - Enter an integer: 1492
  - Enter an integer: Fourteen ninety two
  - Enter an integer: 14.92
  - Enter an integer: -1492
  - Enter an integer: 1 4 9 2

# Another Example

20

```
import java.util.Scanner;

class AddTwo
{
    public static void main (String[] args)
    {
        Scanner scan = new Scanner( System.in );
        int first, second, sum ;      // declaration of int variables

        System.out.println("Enter first integer:");
        first = scan.nextInt();      // read chars and convert to int

        System.out.println("Enter second integer:");
        second = scan.nextInt();     // read chars and convert to int

        sum = first + second;        // add the two ints, put result in sum

        System.out.println("The sum of " + first + " plus " + second + " is " + sum );
    }
}
```

# Integer Division Tester

21

```
import java.util.Scanner;
class IntDivideTest
{
    public static void main (String[] args)
    {
        Scanner scan = new Scanner( System.in );

        int dividend, divisor ;           // int versions of input
        int quotient, remainder ;         // results of "/" and "%"

        System.out.println("Enter the dividend:"); // read the dividend
        dividend = scan.nextInt();

        System.out.println("Enter the divisor:"); // read the divisor
        divisor = scan.nextInt();

        quotient = dividend / divisor ;      // perform int math
        remainder= dividend % divisor ;

        System.out.println( dividend + " / " + divisor + " is " + quotient );
        System.out.println( dividend + " % " + divisor + " is " + remainder );
        System.out.println( quotient + " * " + divisor +
            " + " + remainder + " is " + (quotient*divisor+remainder) );
    }
}
```



**Thank You !**

**afif.supianto@ub.ac.id**  
**081 331 834 734 / 088 160 127 40**