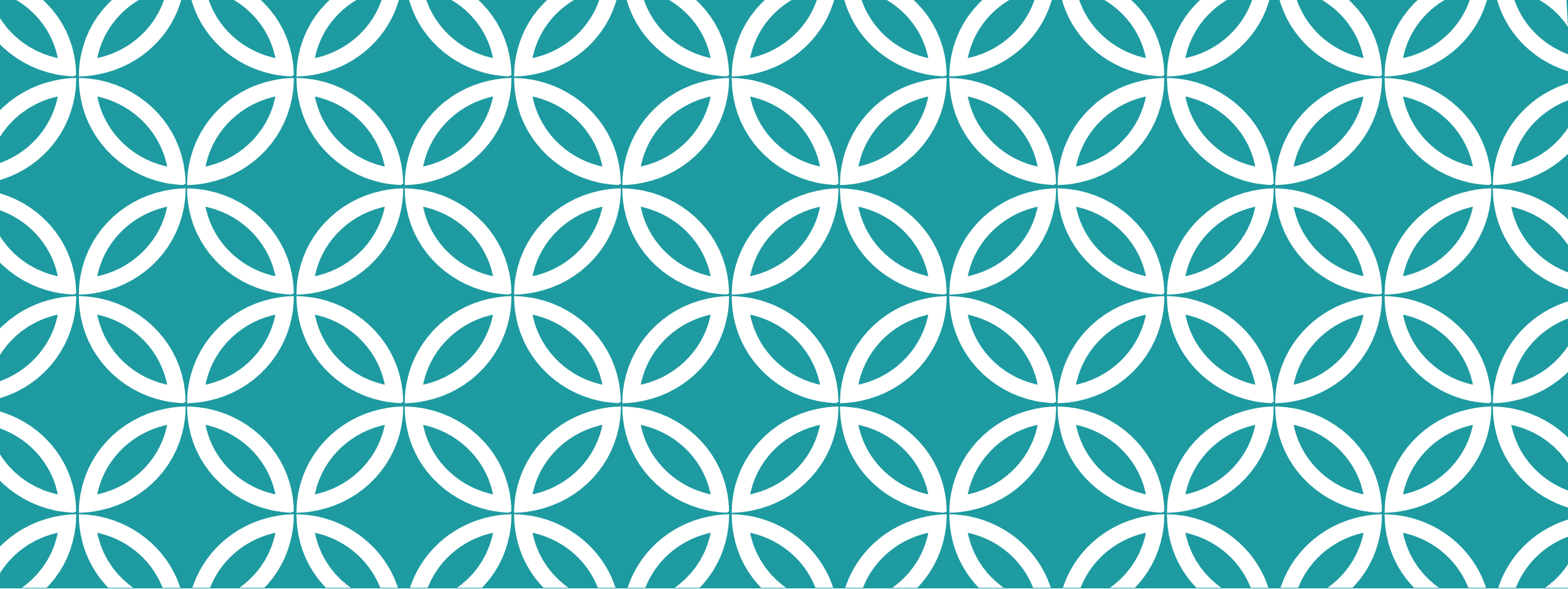




# INPUT & OUTPUT

**Algorithms and Programming**  
Renny P. Kusumawardani  
Department of Information Systems  
ITS



**INPUT/OUTPUT** |

# OUTPUT TO STANDARD OUTPUT

Kita telah mempelajari cara mencetak keluaran ke standard output, yaitu dengan menggunakan fungsi-fungsi pada kelas System.out, diantaranya:

```
System.out.print(argumen);
```

```
System.out.println(argumen);
```

Kelas System juga menyediakan fungsi-fungsi yang terkait dengan standard input

(For further information, refer to:

<http://docs.oracle.com/javase/8/docs/api/java/lang/System.html>)

# GETTING ARGUMENTS VIA THE MAIN FUNCTION

Pengguna program anda dapat memberikan 'masukan' melalui argumen dari fungsi main

Ingat ini?

```
public static void main(String[] args) { ... }
```

- Anda dapat mengakses nilai variabel String array args
- Nilai variabel args diberikan saat eksekusi java dilakukan:

```
java NamaKelas args[0] args[1] args[2] ...
```

- Satu nilai dengan nilai yang lain dipisahkan dengan spasi
- Program boleh tidak diberi argumen, jika memang dalam program tidak dibutuhkan

# EXAMPLE

Compile kelas HelloStudent

```
javac HelloStudent.java
```

Jalankan dengan memberikan argumen

```
java HelloStudent Vera
```

```
class HelloStudent {  
    public static void main(String[] masukan) {  
        System.out.println("Hello, " + masukan[0] + "!");  
    }  
}
```

# CONVERSION FROM STRING TO OTHER DATA TYPES

- Argumen dari program secara default akan dibaca sebagai String
- Anda perlu melakukan parsing (pembacaan) argumen

Contoh:

```
int angkaMasukan = Integer.parseInt(args[0]);
```

# IO USING THE SCANNER CLASS

<https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Scanner.html>

```
Scanner sc = new Scanner(System.in);  
System.out.println("Please input the value of the initial velocity: ");  
v = sc.nextDouble();
```

```
String s = sc.next();  
System.out.println(s);  
int i = sc.nextInt();
```

Warning: untuk di kelas/tugas, JANGAN menggunakan Scanner apabila tidak diperlukan!

- Contoh: misalnya yang ditanyakan adalah kode untuk rekursi, sorting, dsb. yang sebetulnya tidak membutuhkan user input, tidak perlu menggunakan Scanner, supaya inti dari kode tetap terlihat jelas

# IMPORT

```
import java.util.Scanner
```

Constructors:

- File
- Input Stream
- etc.



# NEXT AND HAS NEXT

`next`: returns various datatypes

`hasNext`: for checking whether the input indeed can be interpreted as the datatype checked by the `hasNext`

# EXCEPTIONS

Let's try some things in which unexpected inputs are received....

1. Compilation **error** → code won't even compile
2. Runtime **exception** → because of external/user inputs, etc.

# TRY THIS:

- ❖ Tuliskan kode untuk formula yang sudah Anda buat untuk tugas dalam bentuk method
- ❖ Tuliskanlah kode untuk meminta nilai-nilai parameter untuk kode tersebut dari pengguna

# GETTING INPUT FROM STANDARD INPUT

Anda dapat membaca input dari standard input dengan kelas `System.in`

- `System.in` ini dapat dijadikan argumen untuk pembuatan `InputStreamReader`
- Selanjutnya, objek `InputStreamReader` yang dihasilkan dapat dijadikan masukan untuk pembuatan `BufferedReader`

Sementara ini, silakan menganggap urutan-urutan ini sebagai mantra-mantra

- Silakan diingat, sambil dicoba dimengerti

# MAGIC SPELLS ZONE

## Rules

- Hafalkan
- You don't need to know yet why it is the way it is. Pada saatnya, anda akan mengerti
- Ingat-ingatlah mantra-mantra ini setiap hari sebelum anda tidur 😊

```
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.IOException;

public class TestInputRead {
    public static void main(String[] args) {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        try {
            br.readLine();
        } catch (IOException iox) {

        }
    }
}
```

But this will not 'produce' any outputs.... What happens?



<ketikkan import statements seperti slide lalu...>

```
public class TestInputRead {  
    public static void main(String[] args) {  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
        System.out.println("Silakan ketikkan nama anda, diikuti Enter");  
        try {  
            String inputString = br.readLine();  
            System.out.println("Nama anda adalah: " + inputString);  
        } catch (IOException iox) {  
            iox.printStackTrace();  
        }  
    }  
}
```

## CONTOH DENGAN AKSESORIS TAMBAHAN

<ketikkan import statements seperti slide lalu...>

```
public class TestInputRead {  
    public static void main(String[] args) {  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
        boolean readOn = true;  
        String inputString = null;  
        while (readOn) {  
            try {  
                System.out.println("Silakan ketikkan nama anda, diikuti Enter");  
                inputString = br.readLine();  
                System.out.println("Nama anda adalah: " + inputString);  
                readOn = false;  
            } catch (IOException iox) {  
                System.out.println("Terjadi kesalahan pembacaan!");  
                System.out.println("Silakan coba masukkan nama anda kembali.");  
            }  
        }  
    }  
}
```



# CHALLENGE #3

1. Silakan anda mencoba mengetikkan kode-kode yang ada di slide kuliah ini (tentang I/O) dan mengamati hasilnya.
  - Apa perbedaan antara kode dengan System.in yang sudah dibahas di kelas dengan kode “Contoh dengan aksesoris tambahan”?
2. Silakan membuat variasi sederhana terhadap program yang anda hasilkan

Ingat: PRACTICE MAKES PERFECT!