

Interface

- Interface digunakan apabila kita ingin menentukan apa yang harus dilakukan oleh suatu class tapi **tidak menentukan bagaimana cara untuk melakukannya**
- Interface sebenarnya sama dengan class, tapi hanya memiliki **deklarasi method tanpa implementasi**
- Keyword : *interface name*

TesInterface.java

```
interface IntLampu{  
    public static final int KeadaanHidup=1;  
    public static final int KeadaanMati=0;  
  
    public abstract void hidupkan();  
    public abstract void matikan();  
}
```

```
class Lampu implements IntLampu{
    private int statusLampu=0;

    public void hidupkan(){
        if (this.statusLampu == KeadaanMati){
            this.statusLampu = KeadaanHidup;
            System.out.println("Hidupkan Lampu! --> Lampu Hidup");
        }else{
            System.out.println("Hidupkan Lampu! --> Lampu Sudah Hidup
Kok");
        }
    }

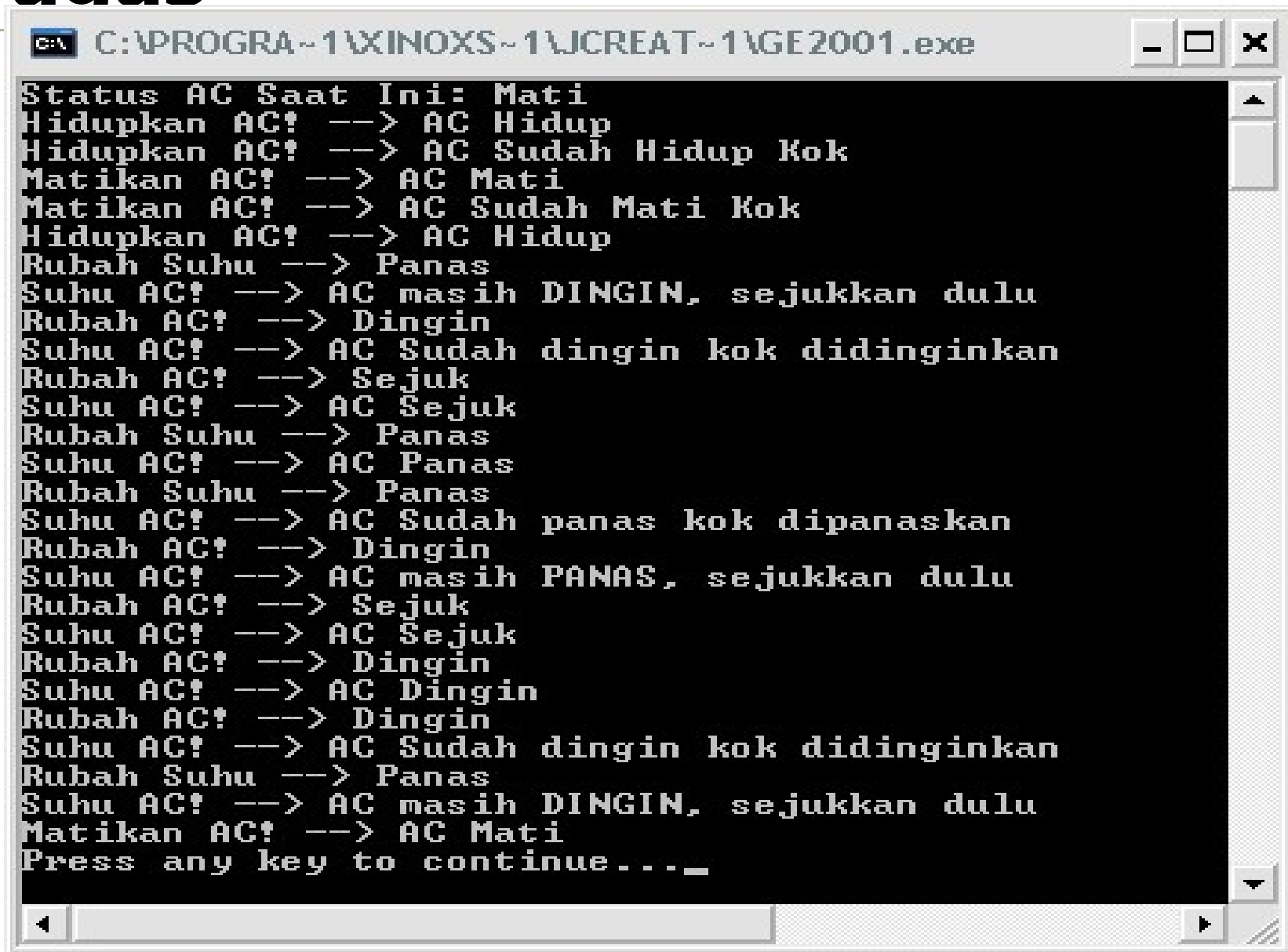
    public void matikan(){
        if (this.statusLampu == KeadaanHidup){
            this.statusLampu = KeadaanMati;
            System.out.println("Matikan Lampu! --> Lampu Mati");
        }else{
            System.out.println("Matikan Lampu! --> Lampu Sudah Mati
Kok");
        }
    }
}
```



```
public class TesInterface {  
    public static void main(String[] args){  
        Lampu lampuKamar = new Lampu();  
        System.out.println("Status Lampu Saat Ini: Mati");  
        lampuKamar.hidupkan(); //Hidupkan Lampu  
        lampuKamar.matikan(); //Matikan Lampu  
        lampuKamar.matikan(); //Matikan Lampu  
    }  
}
```



Tugas



```
C:\PROGRA~1\XINOXS~1\JCREAT~1\GE2001.exe

Status AC Saat Ini: Mati
Hidupkan AC? --> AC Hidup
Hidupkan AC? --> AC Sudah Hidup Kok
Matikan AC? --> AC Mati
Matikan AC? --> AC Sudah Mati Kok
Hidupkan AC? --> AC Hidup
Rubah Suhu --> Panas
Suhu AC? --> AC masih DINGIN, sejukkan dulu
Rubah AC? --> Dingin
Suhu AC? --> AC Sudah dingin kok didinginkan
Rubah AC? --> Sejuk
Suhu AC? --> AC Sejuk
Rubah Suhu --> Panas
Suhu AC? --> AC Panas
Rubah Suhu --> Panas
Suhu AC? --> AC Sudah panas kok dipanaskan
Rubah AC? --> Dingin
Suhu AC? --> AC masih PANAS, sejukkan dulu
Rubah AC? --> Sejuk
Suhu AC? --> AC Sejuk
Rubah AC? --> Dingin
Suhu AC? --> AC Dingin
Rubah AC? --> Dingin
Suhu AC? --> AC Sudah dingin kok didinginkan
Rubah Suhu --> Panas
Suhu AC? --> AC masih DINGIN, sejukkan dulu
Matikan AC? --> AC Mati
Press any key to continue...
```

Java API untuk Referensi Pemrograman



JDK JRE	Toolkits	Accessibility		Drag n Drop		Input Methods		Image I/O		Print Service		Sound		Java SE API
	Integration Libraries	IDL		JDBC™		JNDI™		RMI		RMI-IIOP		Scripting		
	Other Base Libraries	Beans		Intl Support		I/O		JMX		JNI		Math		
		Networking		Override Mechanism		Security		Serialization		Extension Mechanism		XML JAXP		
	lang and util Base Libraries	lang and util		Collections		Concurrency Utilities		JAR		Logging		Management		
		Preferences API		Ref Objects		Reflection		Regular Expressions		Versioning		Zip	Instrument	
	Java Virtual Machine	Java Hotspot™ Client VM						Java Hotspot™ Server VM						
	Platforms	Solaris™			Linux			Windows			Other			

Release Notes

Topics include New Features, Known Issues, Compatibility with Prior Releases, Supported System Configurations, Installation, and More)

API, Language, and Virtual Machine Documentation

[Java Platform API Specification](#) [\(NO FRAMES\)](#)
(included in the JDK documentation bundle and on java.sun.com)

[The Java Language Specification](#) [\(DOWNLOAD\)](#)

[Note About sun.* Packages](#)

[The Java Virtual Machine Specification](#) [\(DOWNLOAD\)](#)

Java™ Platform, Standard Edition 6 API Specification

This document is the API specification for version 6 of the Java™ Platform, Standard Edition.

See:

[Description](#)

Packages

java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.

Rehat Sejenak

- [Evian](#)