

# Package

---



## Object Oriented Programming



# Packages

---

- Package adalah koleksi dari beberapa class dan interface yang berhubungan, dan menyediakan proteksi akses dan pengelolaan namespace
- 1 package adalah 1 subfolder di file system
- Package sangat berguna untuk mengorganisir file dalam suatu project atau library
- Keyword: *package name;*



# Budi.java

---

```
package kelasku;
```

```
public class Budi{  
    public void info(){  
        System.out.println("Kelas Budi");  
    }  
}
```



# Joko.java

---

```
package kelasku;
```

```
public class Joko{  
    public void info(){  
        System.out.println("Kelas Joko");  
    }  
}
```

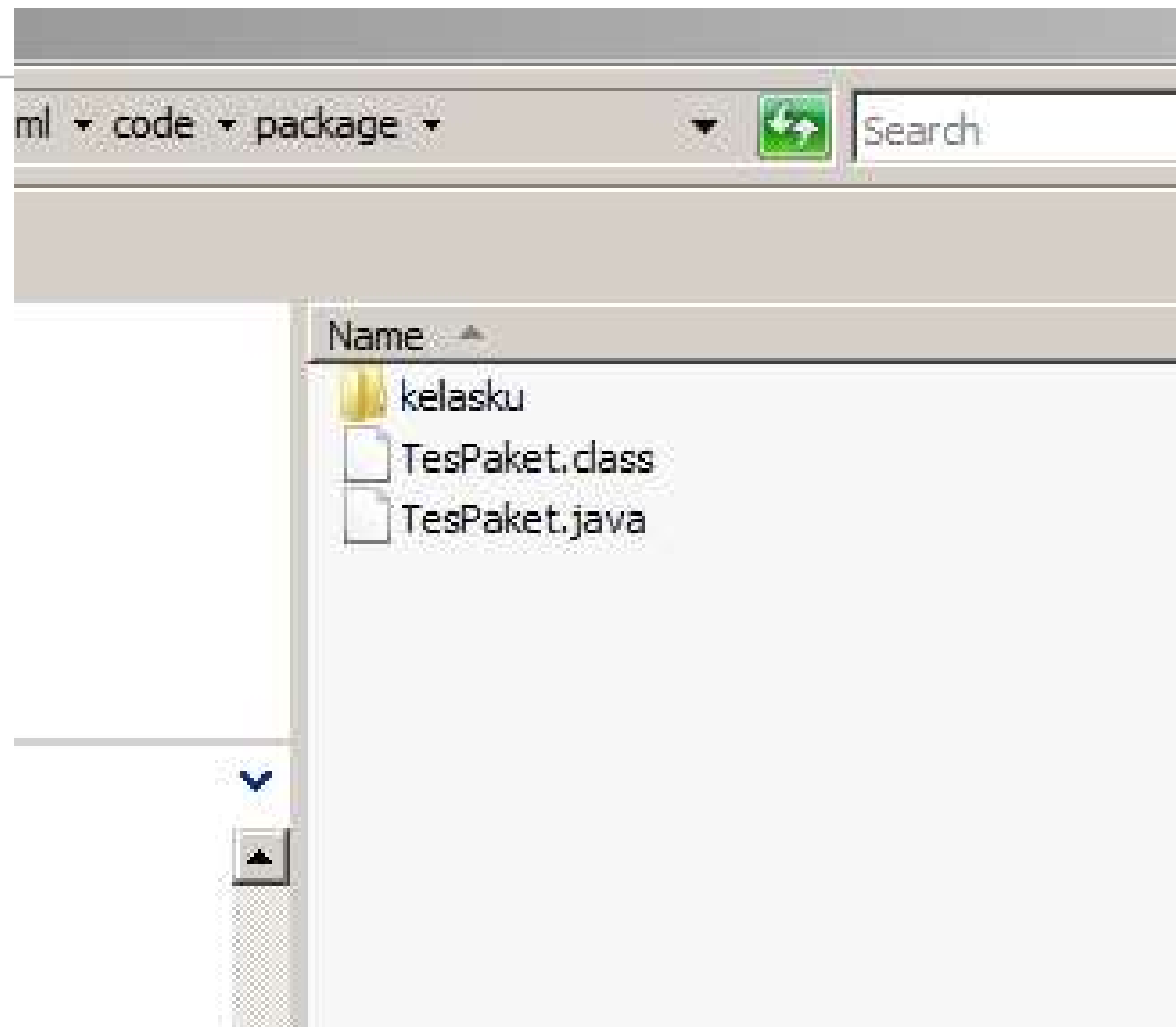


# TesPaket.java

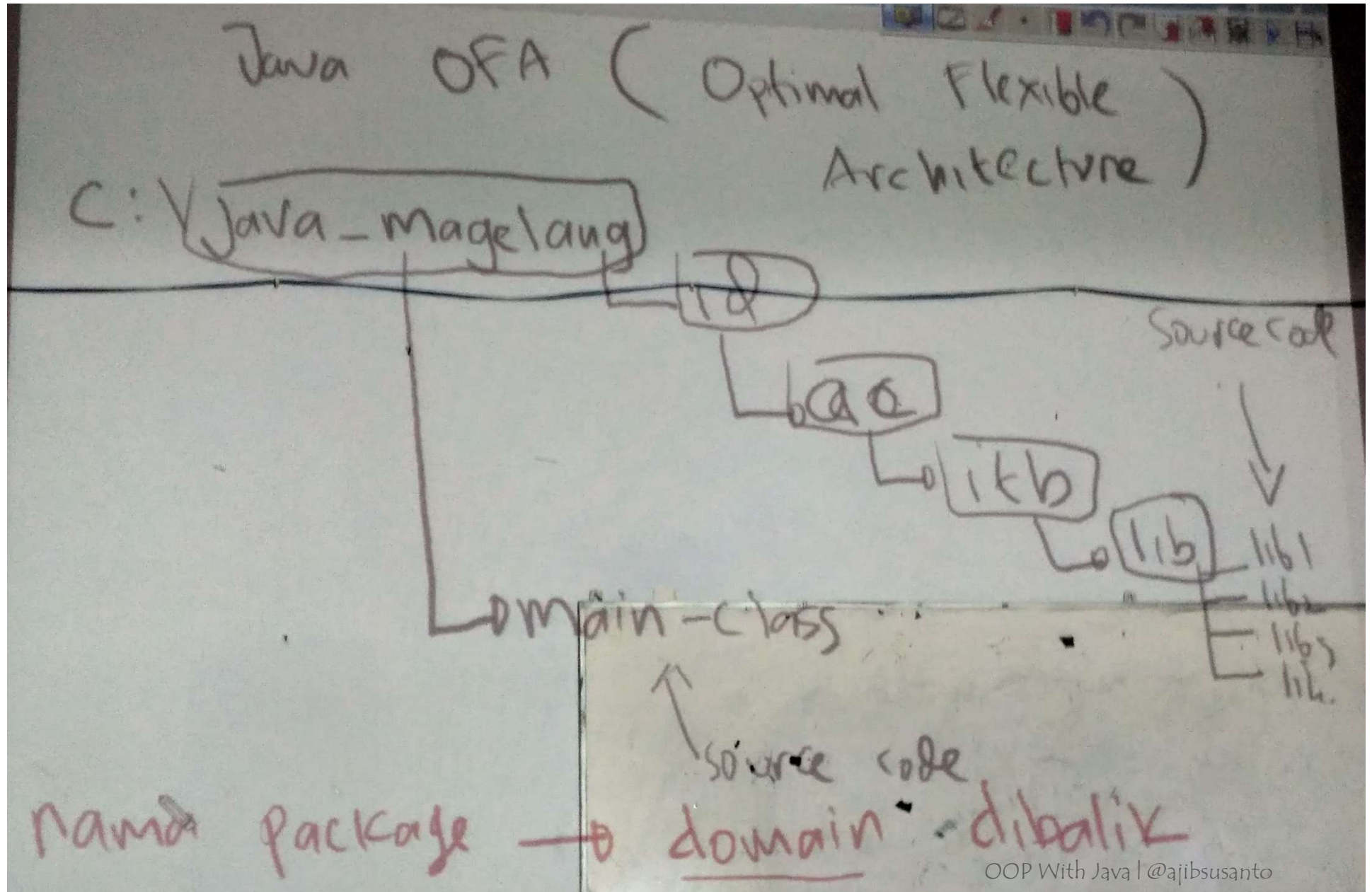
```
import kelasku.Budi;  
import kelasku.Joko;
```

```
public class TesPaket{  
    public static void main(String[] args){  
        Budi objectBudi = new Budi();  
        objectBudi.info();  
        Joko objectJoko = new Joko();  
        objectJoko.info();  
    }  
}
```

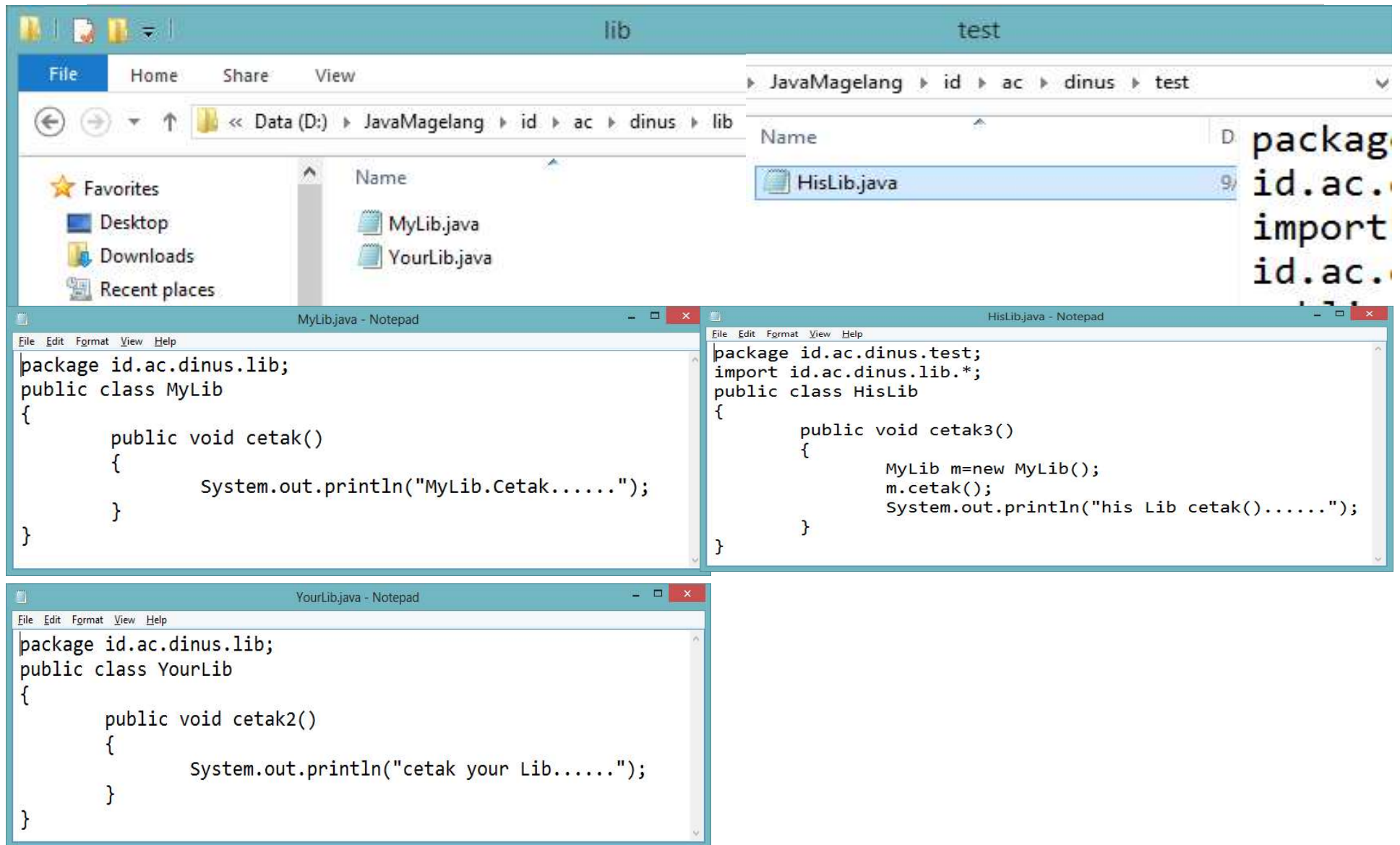




# Packages



# Packages



The screenshot displays a Windows file explorer window showing the directory structure: Data (D:) > JavaMagelang > id > ac > dinus > lib. The file explorer shows two files: MyLib.java and YourLib.java. To the right, a list of files in the 'test' directory is shown, including HisLib.java. Below the file explorer, three Notepad windows are open, each containing Java code for a different class in a specific package.

**MyLib.java - Notepad**

```
package id.ac.dinus.lib;
public class MyLib
{
    public void cetak()
    {
        System.out.println("MyLib.Cetak.....");
    }
}
```

**HisLib.java - Notepad**

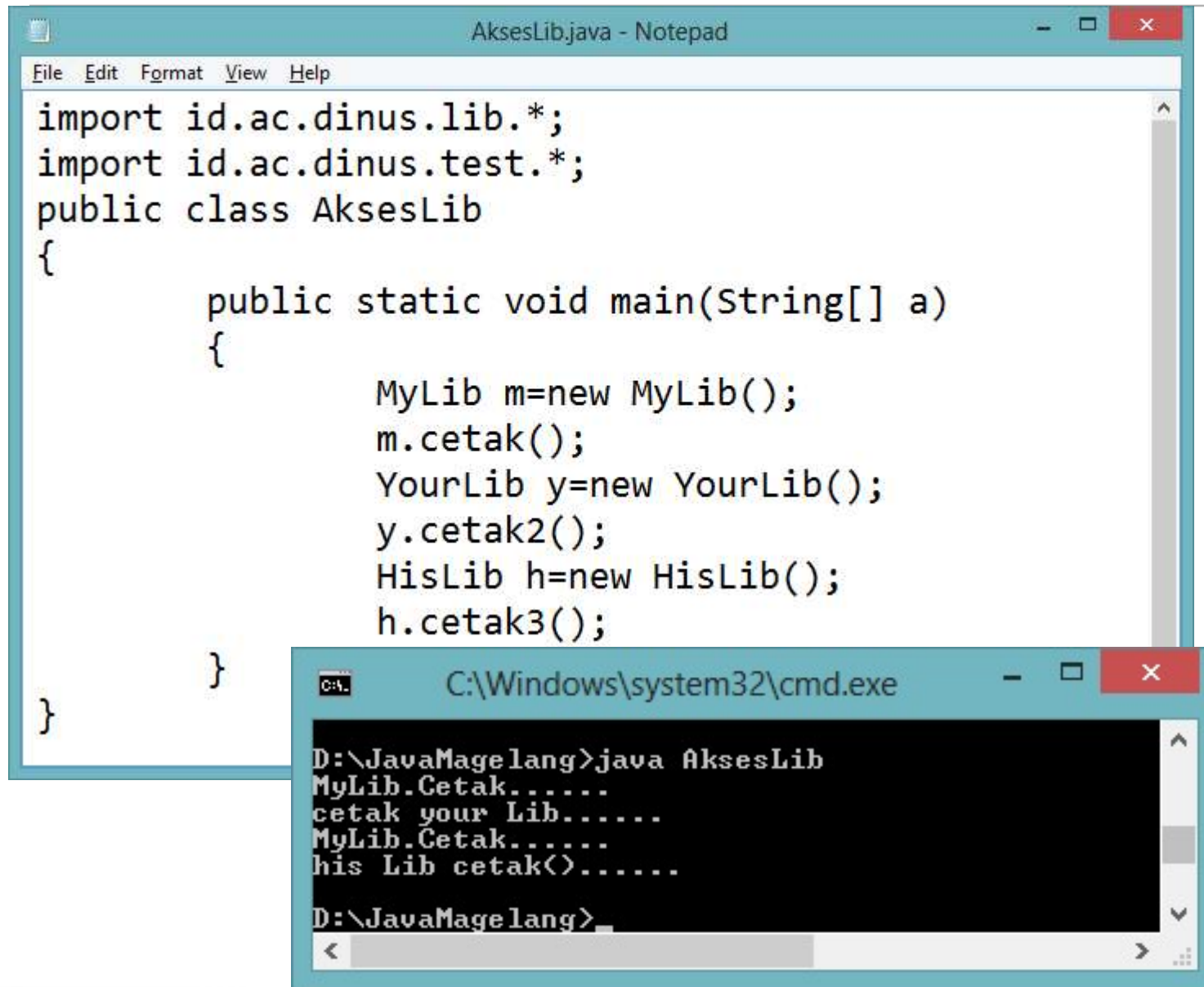
```
package id.ac.dinus.test;
import id.ac.dinus.lib.*;
public class HisLib
{
    public void cetak3()
    {
        MyLib m=new MyLib();
        m.cetak();
        System.out.println("his Lib cetak().....");
    }
}
```

**YourLib.java - Notepad**

```
package id.ac.dinus.lib;
public class YourLib
{
    public void cetak2()
    {
        System.out.println("cetak your Lib.....");
    }
}
```



# Packages



The image shows a Notepad window titled 'AksesLib.java - Notepad' containing the following Java code:

```
import id.ac.dinus.lib.*;
import id.ac.dinus.test.*;
public class AksesLib
{
    public static void main(String[] a)
    {
        MyLib m=new MyLib();
        m.cetak();
        YourLib y=new YourLib();
        y.cetak2();
        HisLib h=new HisLib();
        h.cetak3();
    }
}
```

Below the Notepad window is a Command Prompt window titled 'C:\Windows\system32\cmd.exe'. It shows the execution of the Java code:

```
D:\JavaMagelang>java AksesLib
MyLib.Cetak.....
cetak your Lib.....
MyLib.Cetak.....
his Lib cetak().....
D:\JavaMagelang>
```



# Packages

```
AksesLib2.java - Notepad
File Edit Format View Help
import id.ac.dinus.lib.MyLib; //lsg ke class
import id.ac.dinus.lib.YourLib;
import id.ac.dinus.test.*;
public class AksesLib2
{
    public static void main(String[] a)
    {
        MyLib m=new MyLib();
        m.cetak();
        YourLib y=new YourLib();
        y.cetak2();
        HisLib h=new HisLib();
        h.cetak3();
    }
}
```

```
C:\Windows\system32\cmd.exe
D:\JavaMagelang>java AksesLib2
MyLib.Cetak.....
cetak your Lib.....
MyLib.Cetak.....
his Lib cetak().....
D:\JavaMagelang>
```



# Tugas

---

- Paketkan berbagai class yang kita buat (*Mobil dan Bicycle*) dalam package *Transportasi*
- Panggil dari class lain (*TransportasiDemo.java*) dengan *import*



# Rehat Sejenak

---

- Shower Gel



# Java Class - Library

---



Object Oriented Programming



# Class Libraries

---

- sebuah kumpulan dari program yang disertakan dalam Java.
- Pemrograman menjadi lebih mudah.
- Kelengkapan library semakin beragam jika ditambah dengan karya komunitas Java



# Class Libraries

<i>Package</i>	<i>Nama Package</i>	<i>Keterangan</i>
<i>Language</i>	<code>java.lang</code>	<i>Class-class</i> utama yang merupakan inti dari bahasa Java
<i>Utilities</i>	<code>java.util</code>	<i>Class-class</i> yang mendukung utilitas struktur Java
<i>IO</i>	<code>java.io</code>	<i>Class</i> yang mendukung berbagai macam tipe input dan output
<i>Text</i>	<code>java.text</code>	<i>Class</i> yang mendukung lokalisasi penanganan teks, tanggal, bilangan, dan <i>message</i>
<i>Math</i>	<code>java.math</code>	<i>Class</i> untuk melakukan perhitungan aritmatik <i>arbitrary-procesion</i> , baik integer atau <i>floating point</i> .
<i>AWT</i>	<code>java.awt</code>	<i>Class</i> untuk perancangan <i>user-interface</i> dan <i>event-handling</i>
<i>Swing</i>	<code>javax.swing</code>	<i>Class</i> untuk membuat berbagai komponen dalam Java yang bertingkah laku sama dengan berbagai <i>platform</i> .
<i>Javax</i>	<code>javax</code>	Perluasan dari bahasa Java
<i>Applet</i>	<code>java.applet</code>	<i>Class</i> untuk membuat applet
<i>Beans</i>	<code>java.beans</code>	<i>Class</i> untuk membuat Java Beans
<i>Reflection</i>	<code>java.lang.reflect</code>	<i>Class</i> untuk memperoleh informasi <i>run-time</i>
<i>SQL</i>	<code>java.sql</code>	<i>Class</i> untuk mendukung akses dan pengolahan data dalam database.
<i>RMI</i>	<code>java.rmi</code>	<i>Class</i> untuk mendukung <i>distributed programming</i> .
<i>Network</i>	<code>java.net</code>	<i>Class</i> untuk mendukung dalam membangun aplikasi jaringan.
<i>Security</i>	<code>java.security</code>	<i>Class</i> untuk mendukung keamanan kriptografi



# Library Math

---

- `Math.abs(x)`, menghitung nilai mutlak (absolut) dari `x`. Nilai mutlak bilangan negatif adalah bilangan positif, dan bilangan positif tetap bilangan positif.
- Fungsi trigonometri `Math.sin(x)`, `Math.cos(x)`, and `Math.tan(x)` (Untuk semua fungsi trigonometri, sudut memiliki satuan radian, bukan derajat)





# Library Math

---

- Fungsi trigonometri inverse, yang mencari sudut dari suatu nilai trigonometric, kebalikan dari fungsi trigonometri, seperti arcus sin, arcus cos, dan arcus tangen. `Math.asin(x)`, `Math.acos(x)`, and `Math.atan(x)`.
- `Math.exp(x)`, menghitung pangkat dari bilangan natural  $e$ , atau  $e^x$ . Dan logaritma natural  $\log_e x$  atau  $\ln x$  bisa dihitung dengan menggunakan fungsi `Math.log(x)`.



# Library Math

---

- `Math.pow(x,y)` menghitung  $x^y$  atau  $x$  pangkat  $y$
- `Math.floor(x)` menghitung pembulatan ke bawah dari suatu bilangan riil, misalnya 3.84 akan dibulatkan ke bawah menjadi 3.0
- `Math.random()` memilih bilangan acak di antara 0.0 dan 1.0. Komputer memiliki algoritma perhitungan tertentu yang hasilnya bilangan acak (meskipun bukan bilangan yang betul-betul acak, tetapi cukup untuk kebanyakan fungsi)



# Library Math

<b>E</b>	double	<b>log10</b> (double a)	double
<b>PI</b>	double	<b>log1p</b> (double x)	double
<b>IEEEremainder</b> (double f1, double f2)	double	<b>max</b> (double a, double b)	double
<b>abs</b> (double a)	double	<b>max</b> (float a, float b)	float
<b>abs</b> (float a)	float	<b>max</b> (int a, int b)	int
<b>abs</b> (int a)	int	<b>max</b> (long a, long b)	long
<b>abs</b> (long a)	long	<b>min</b> (double a, double b)	double
<b>acos</b> (double a)	double	<b>min</b> (float a, float b)	float
<b>asin</b> (double a)	double	<b>min</b> (int a, int b)	int
<b>atan</b> (double a)	double	<b>min</b> (long a, long b)	long
<b>atan2</b> (double y, double x)	double	<b>nextAfter</b> (double start, double direction)	double
<b>cbrt</b> (double a)	double	<b>nextAfter</b> (float start, double direction)	float
<b>ceil</b> (double a)	double	<b>scalb</b> (float f, int scaleFactor)	float
<b>copySign</b> (double magnitude, double sign)	double	<b>signum</b> (double d)	double
<b>copySign</b> (float magnitude, float sign)	float	<b>signum</b> (float f)	float
<b>cos</b> (double a)	double	<b>sin</b> (double a)	double
<b>cosh</b> (double x)	double	<b>sinh</b> (double x)	double
<b>exp</b> (double a)	double	<b>sqrt</b> (double a)	double
<b>expm1</b> (double x)	double	<b>tan</b> (double a)	double
<b>floor</b> (double a)	double	<b>tanh</b> (double x)	double
<b>getExponent</b> (double d)	int	<b>toDegrees</b> (double angrad)	double
<b>getExponent</b> (float f)	int	<b>toRadians</b> (double angdeg)	double
<b>hypot</b> (double x, double y)	double	<b>ulp</b> (double d)	double
<b>log</b> (double a)	double	<b>ulp</b> (float f)	float



# Library Date

- sebuah class yang dapat digunakan untuk mendapatkan tanggal.
- Terletak di package java.util.Date.
- Sebelum menggunakan :
  - import java.util.Date;
- Penerapan cukup sederhana, cukup dengan membuat obyek dari class Date.

```
import java.util.Date;
public class Tanggal {
    public static void main(String[] args) {
        Date tanggal=new Date();
        System.out.println(tanggal);
    }
}
```

Mon Apr 14 21:59:53 ICT 2014



# Format Tanggal

---

```
import java.util.Date;
import java.text.SimpleDateFormat;
public class Tanggal {
    public static void main(String[] args) {
        Date tanggal=new Date();
        SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");
        SimpleDateFormat sdf1=new SimpleDateFormat("dd MMMM yyyy");
        System.out.println(tanggal);
        System.out.println(sdf.format(tanggal));
        System.out.println(sdf1.format(tanggal));
    }
}
```

Mon Apr 14 22:04:31 ICT 2014

14-04-2014

14 April 2014





# Format Tanggal

```
import java.util.Date;
import java.text.SimpleDateFormat;
import java.util.Calendar;

public class Tanggal {
    public static void main(String[] args) {
        Date tanggal=new Date();
        SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");
        SimpleDateFormat sdf1=new SimpleDateFormat("dd MMMM yyyy");
        System.out.println(tanggal);
        System.out.println(sdf.format(tanggal));
        System.out.println(sdf1.format(tanggal));
        Calendar cal=Calendar.getInstance();
        cal.add(Calendar.DATE, 3);
        Date tigaHariLagi=cal.getTime();
        System.out.println("3 Hari lagi : "+sdf.format(tigaHariLagi));
    }
}
```

Mon Apr 14 22:04:31 ICT 2014

14-04-2014

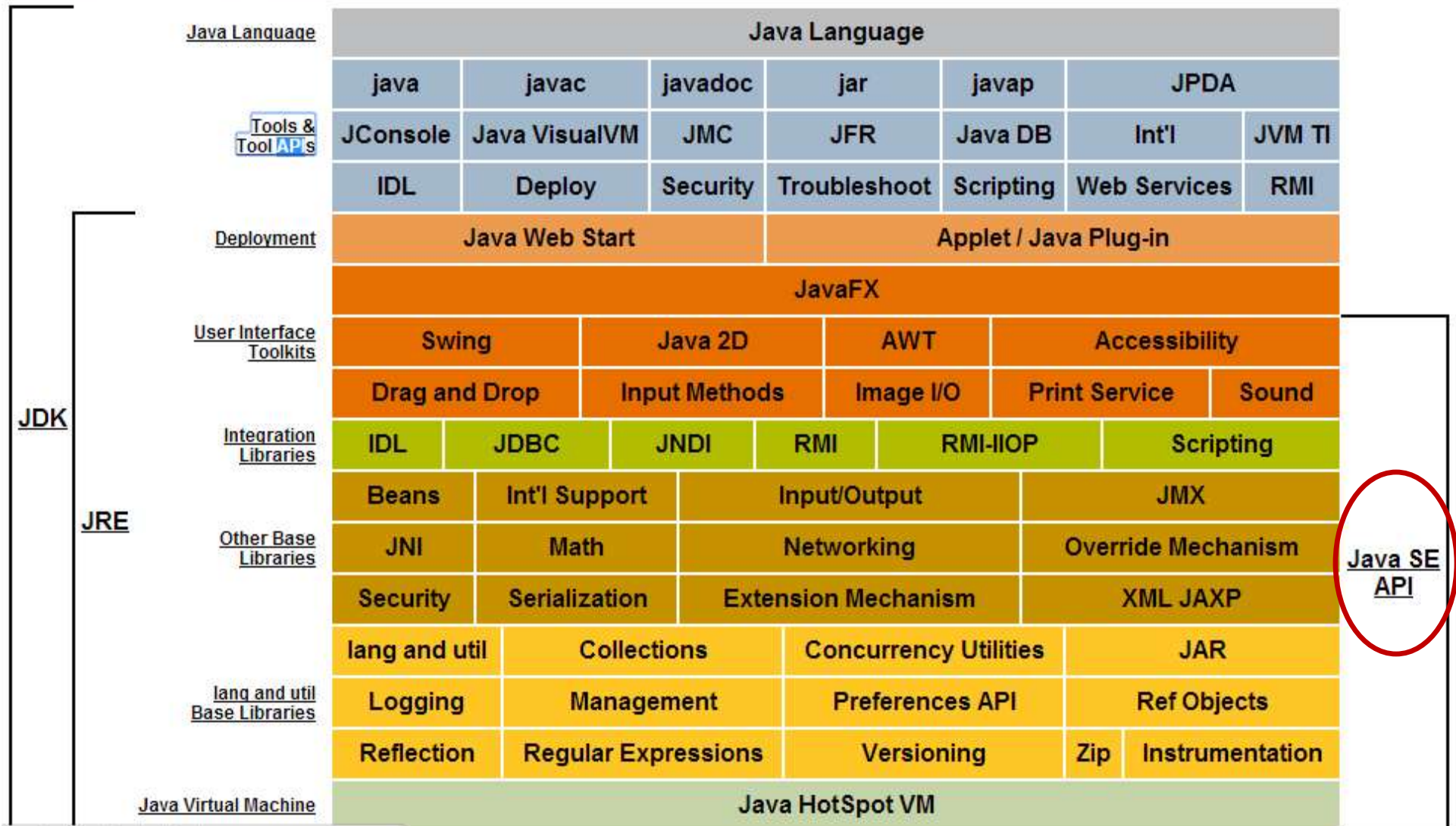
14 April 2014

3 Hari lagi : 17-04-2014



# Java SE API - <http://docs.oracle.com/javase/7/docs/>

## Description of Java Conceptual Diagram



# Java SE API

## <http://docs.oracle.com/javase/7/docs/api/>

Java™ Platform  
Standard Ed. 7

All Classes

Packages

java.applet  
java.awt  
java.awt.color

All Classes

AbstractAction  
AbstractAnnotationValueVisitor6  
AbstractAnnotationValueVisitor7  
AbstractBorder  
AbstractButton  
AbstractCellEditor  
AbstractCollection  
AbstractColorChooserPanel  
AbstractDocument  
AbstractDocument.AttributeContext  
AbstractDocument.Content  
AbstractDocument.ElementEdit  
AbstractElementVisitor6  
AbstractElementVisitor7  
AbstractExecutorService  
AbstractInterruptibleChannel  
AbstractLayoutCache  
AbstractLayoutCache.NodeDimensions  
AbstractList  
AbstractListModel  
AbstractMap  
AbstractMap.SimpleEntry  
AbstractMap.SimpleImmutableEntry  
AbstractMarshallerImpl

OverviewPackageClassUseTreeDeprecatedIndexHelp

PrevNextFramesNo Frames

## Java™ Platform, Standard Edition 7 API Specification

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

See: Description

Packages

Package	Description
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.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
java.awt.print	Provides classes and interfaces for a general printing API.

24

OOP With Java | @ajibsusanto



# Jar (Java Archive)

---

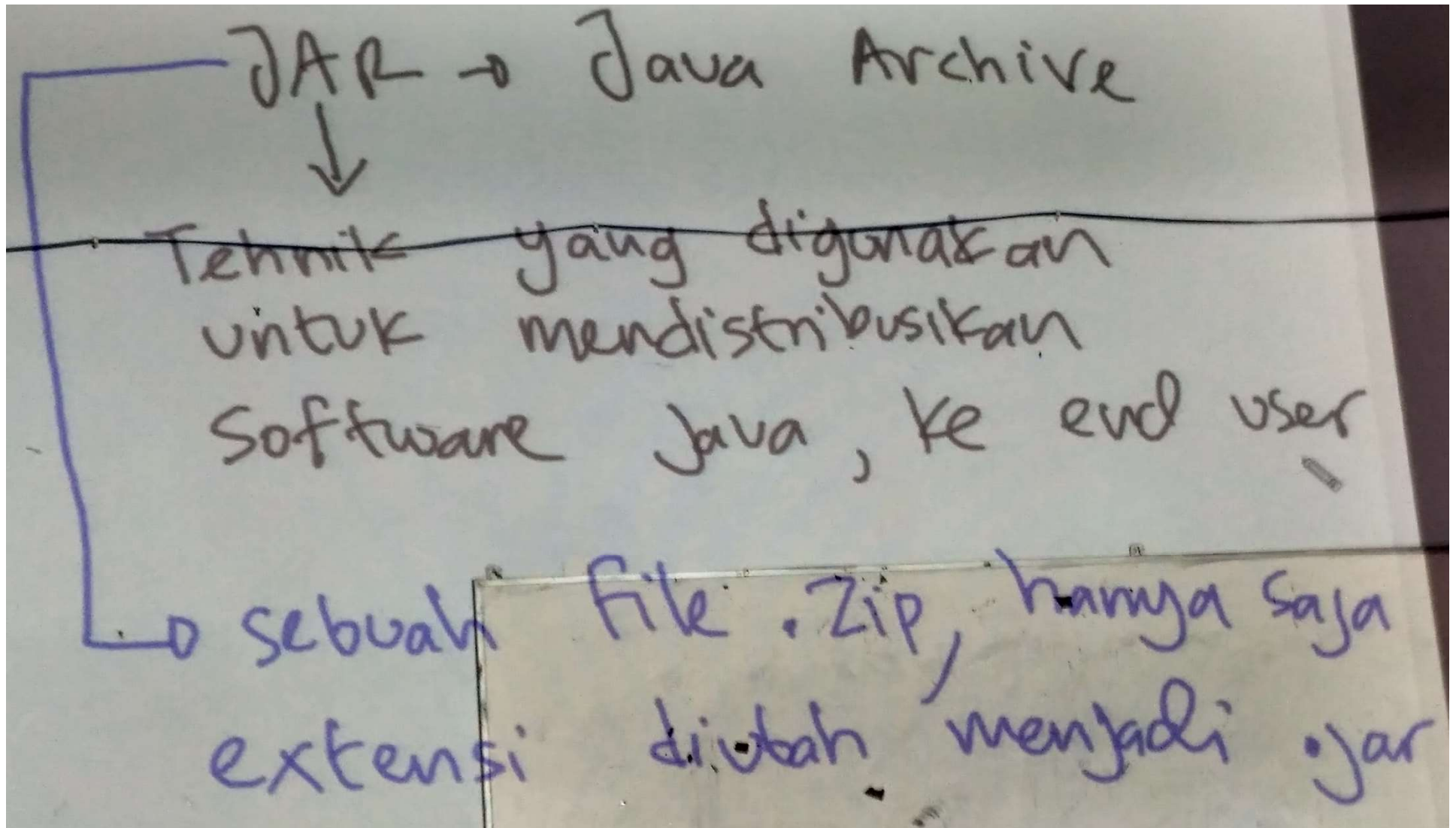


Object Oriented Programming

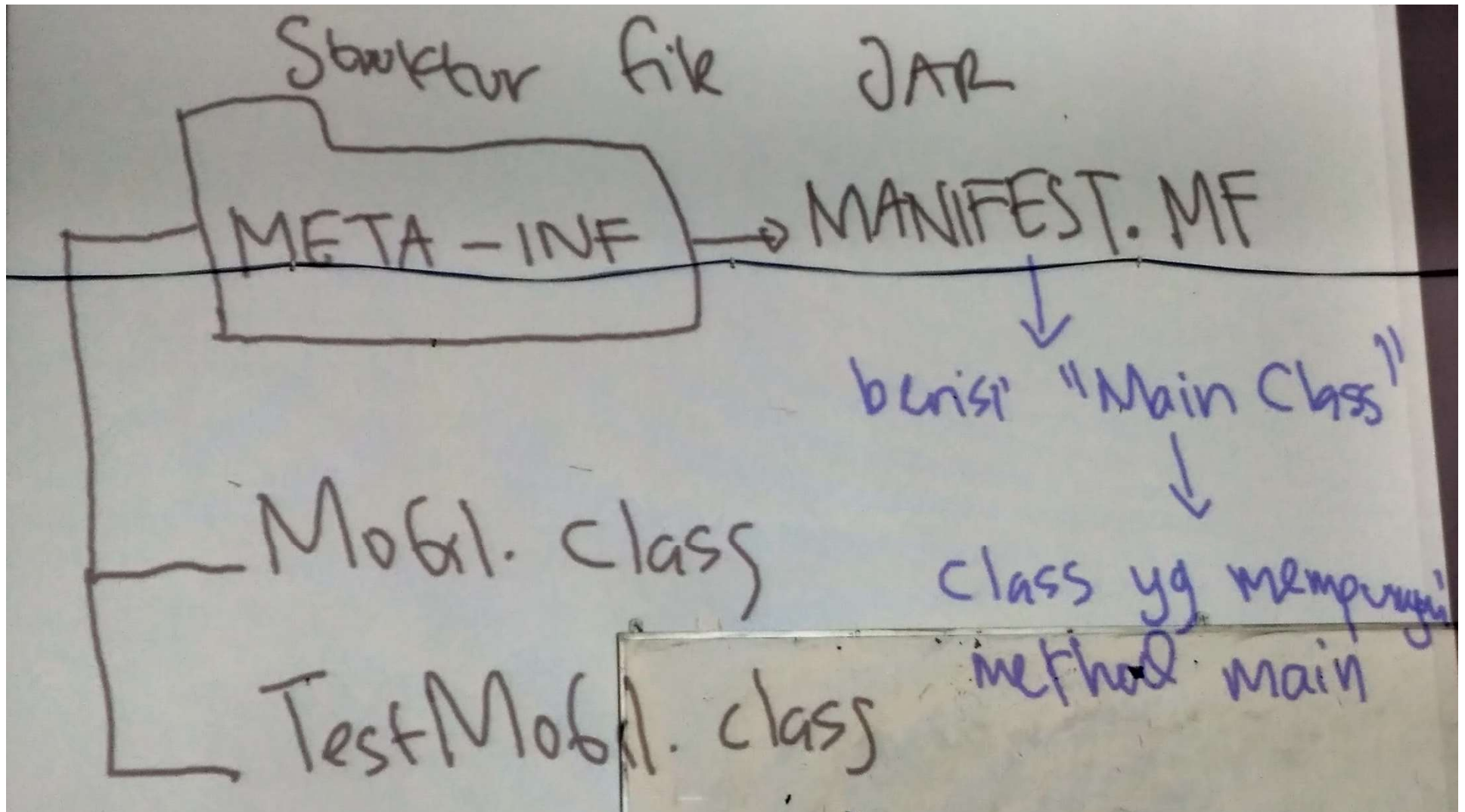


# JAR

---

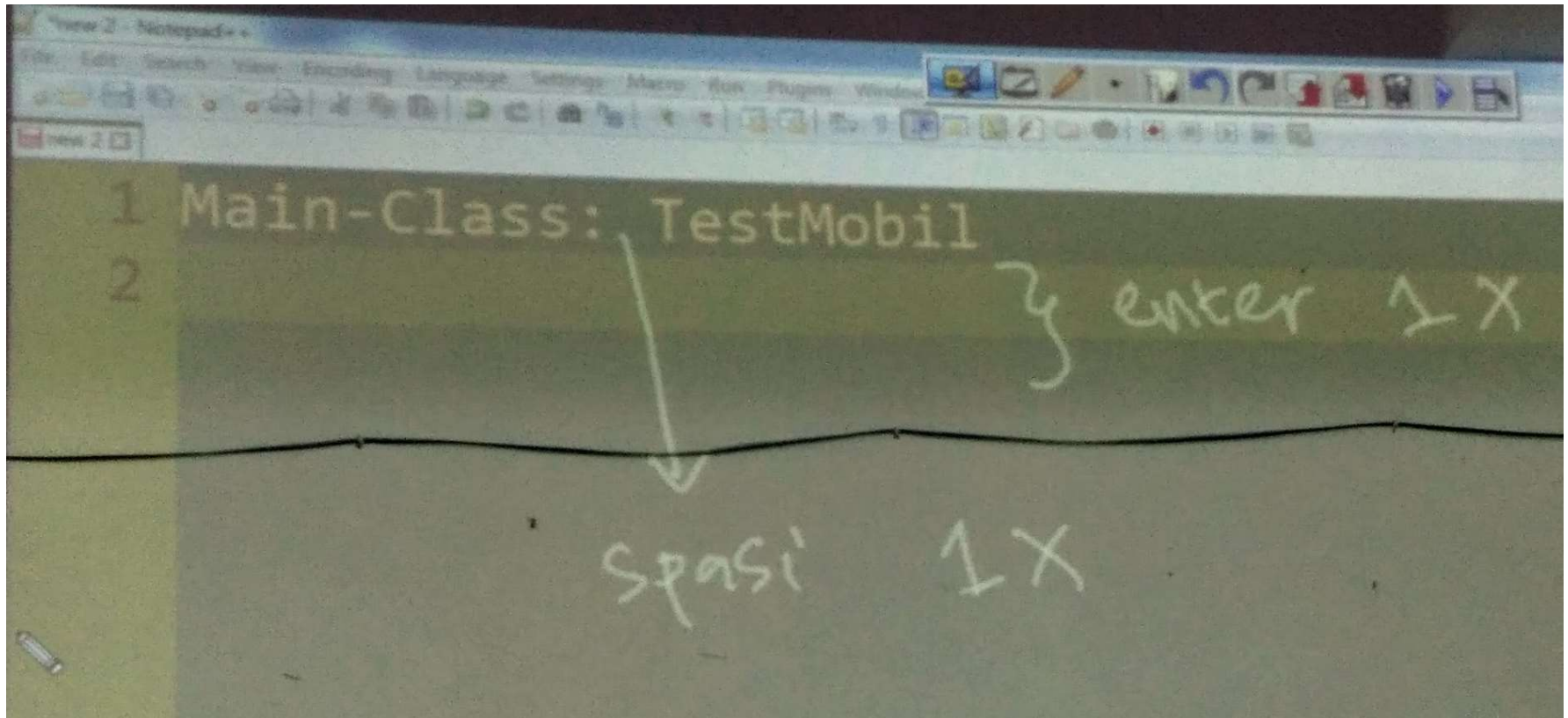


# JAR



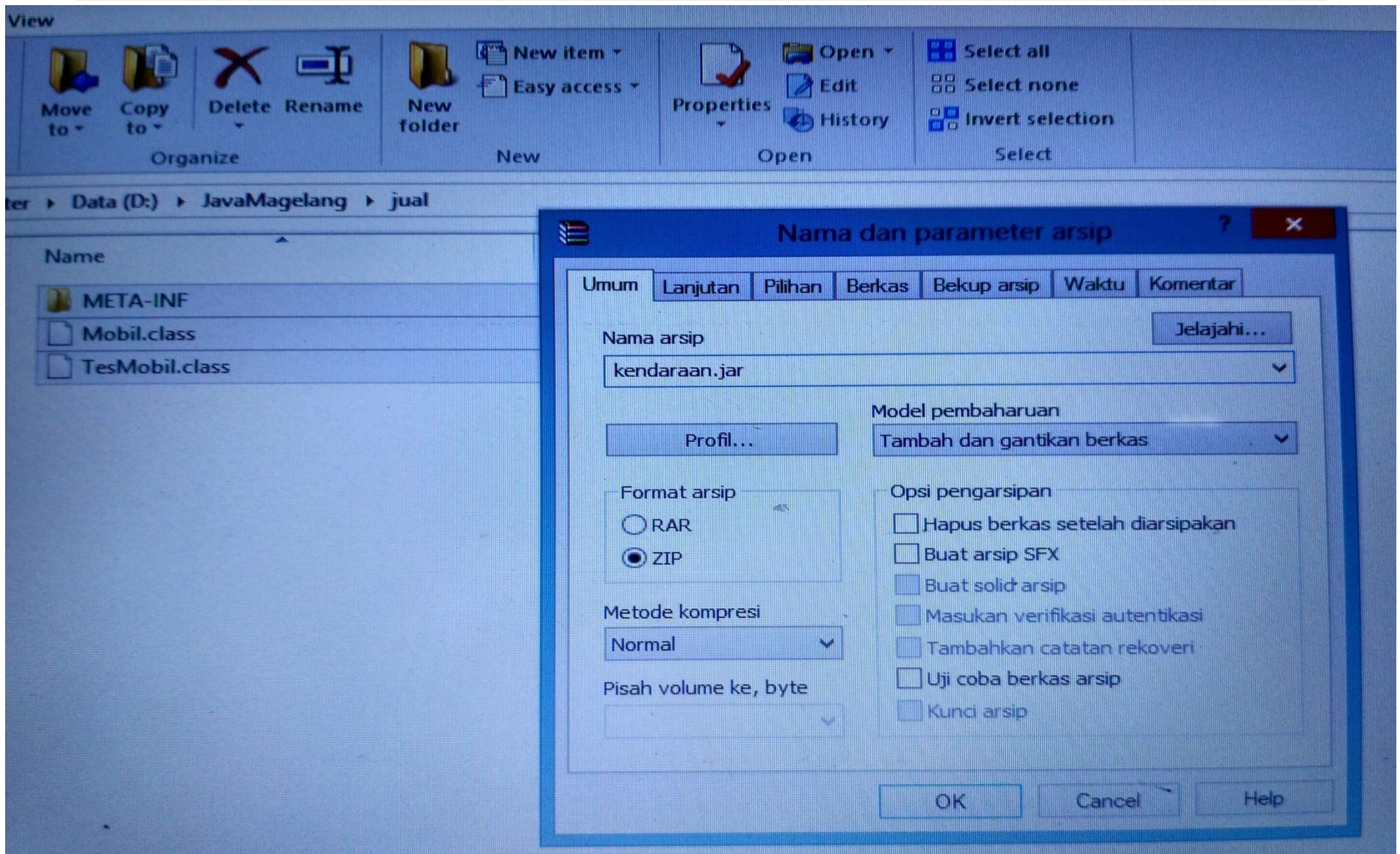
# JAR

---



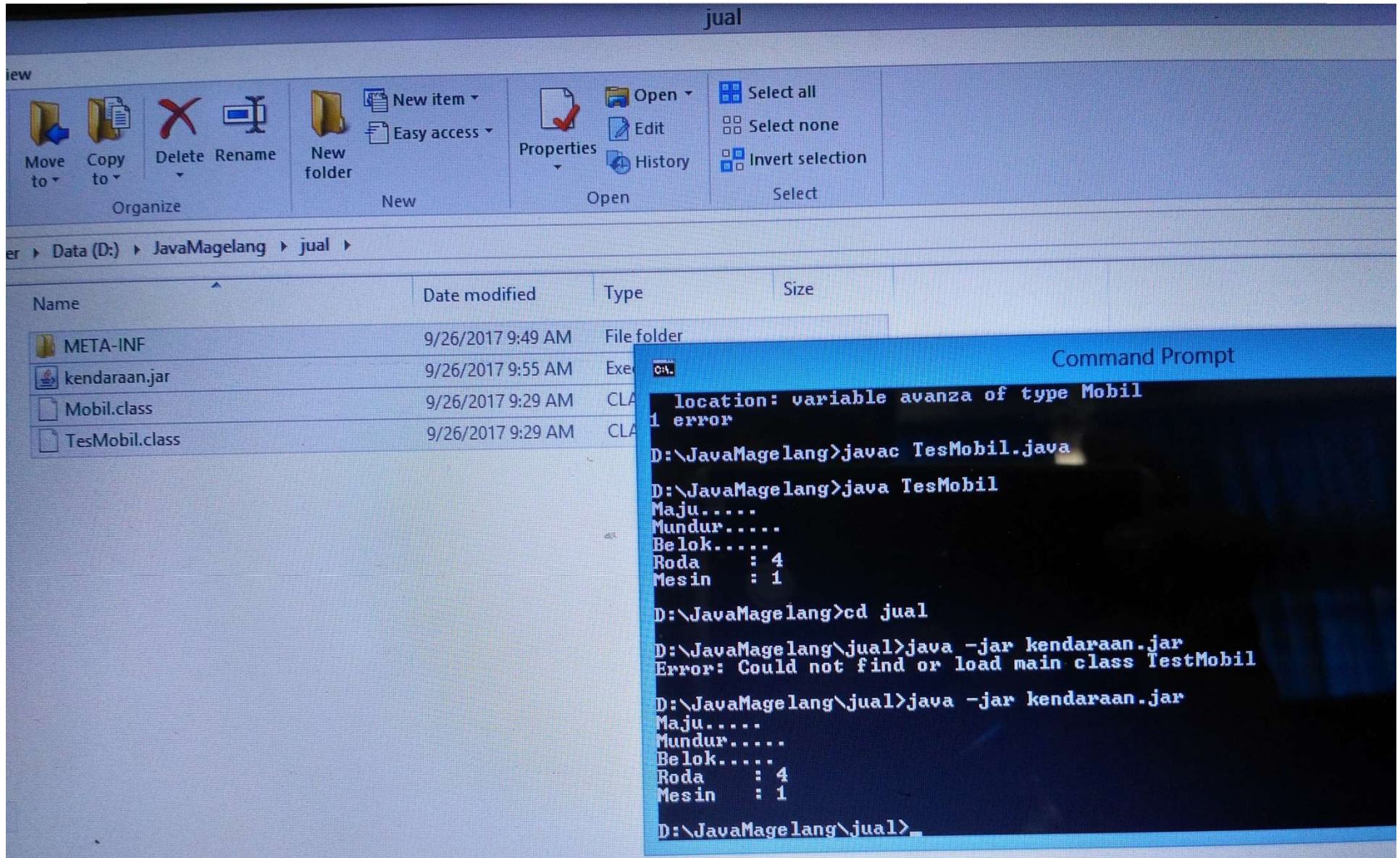


# JAR





# JAR



The screenshot shows a Windows File Explorer window titled 'jual' with the address bar showing 'Data (D:) > JavaMagelang > jual'. The file list contains:

Name	Date modified	Type	Size
META-INF	9/26/2017 9:49 AM	File folder	
kendaraan.jar	9/26/2017 9:55 AM	Executable file	
Mobil.class	9/26/2017 9:29 AM	Class file	
TesMobil.class	9/26/2017 9:29 AM	Class file	

A Command Prompt window is overlaid on the right, showing the following commands and output:

```
location: variable avanza of type Mobil
1 error
D:\JavaMagelang>javac TesMobil.java
D:\JavaMagelang>java TesMobil
Maju.....
Mundur.....
Belok.....
Roda      : 4
Mesin     : 1
D:\JavaMagelang>cd jual
D:\JavaMagelang\jual>java -jar kendaraan.jar
Error: Could not find or load main class TestMobil
D:\JavaMagelang\jual>java -jar kendaraan.jar
Maju.....
Mundur.....
Belok.....
Roda      : 4
Mesin     : 1
D:\JavaMagelang\jual>
```







# Referensi

---

- Object First With Java, Fifth edition, David J. Barnes & Michael Kölling, Prentice Hall / Pearson Education, 2012.
- The Java™ Tutorial,  
<http://docs.oracle.com/javase/tutorial/java/nutsandbolts/>,  
Oracle, 1995–2014.
- Java SE Tutorial,  
<http://www.oracle.com/technetwork/java/javase/downloads/jav-se-7-tutorial-2012-02-28-1536013.html>, Oracle, 2014.
- Java Platform, SE Documentation,  
<https://docs.oracle.com/en/java/javase/index.html>
- SCJP Sun Certified Programmer for Java™ 6 Study Guide Exam (310-065), Kathy Sierra & Bert Bates, Mc Graw Hill, 2008.
- Object Oriented Programming with Java, Romi Satria Wahono, 2008.

