

# Pemrograman Berorientasi Obyek

**Pengantar & Kontrak**

Antonius Rachmat C

# Deskripsi

- Matakuliah ini berisi tentang paradigma pemrograman yang berorientasi obyek, bagaimana cara menyusun langkah-langkah pemecahan masalah menggunakan konsep class dan obyek, aturan-aturan berorientasi obyek, dan menggunakan tool bahasa pemrograman Java & class diagram beserta berbagai studi kasusnya dalam praktikum

# Kompetensi

- Mampu membandingkan paradigma OOP dengan Struktural
- Mampu menjelaskan konsep utama OOP
- Mampu melakukan langkah-langkah pemecahan masalah menggunakan OOP
- Mampu membuat program menggunakan konsep OOP dengan bahasa Java
- Mampu menggunakan berbagai fitur Java lainnya secara umum
- Mampu menggunakan class diagram pada studi kasus
- Mampu menggunakan fitur JDBC sederhana

# Detail

- SKS : 3
- Matakuliah : Wajib
- Kelas : D (Kamis 10.30) dan E (Kamis 13.30)
- Ruang: **D25**
- Berpraktikum 1x seminggu
- Dosen: Antonius Rachmat C, S.Kom, M.Cs
- Email: [anton@ti.ukdw.ac.id](mailto:anton@ti.ukdw.ac.id)
- YM: antonie\_oo
- Web: <http://lecturer.ukdw.ac.id/anton>
- E-class: <http://www.ukdw.ac.id/e-class>

# Silabus

- Kontrak Perkuliahan dan Overview PBO – 19/1
- Class dan Object I – 26/1
- Class dan Object II – 2/2
- Atribut dan Method – 9/2
- Inheritance – 16/2
- Interface & Abstract - 23/2
- **Tes Kecil I - 1/3**
- Masa TTS – **Remidi Tes Kecil 1**

# Silabus (2)

- Polymorphism dan Data Binding – 22/3
- Package & Relasi Antar Class – 29/3
- Array dan Collections – 12/4
- Class Diagram – 19/4
- **Tes Kecil II – 26/4**
- Exception Handling – 3/5
- JAR File dan JDBC – 10/5 (**ada remidi Tes Kecil II**)
- **Tes Akhir Semester** (bahan: semua materi - open books)

# Komponen Penilaian

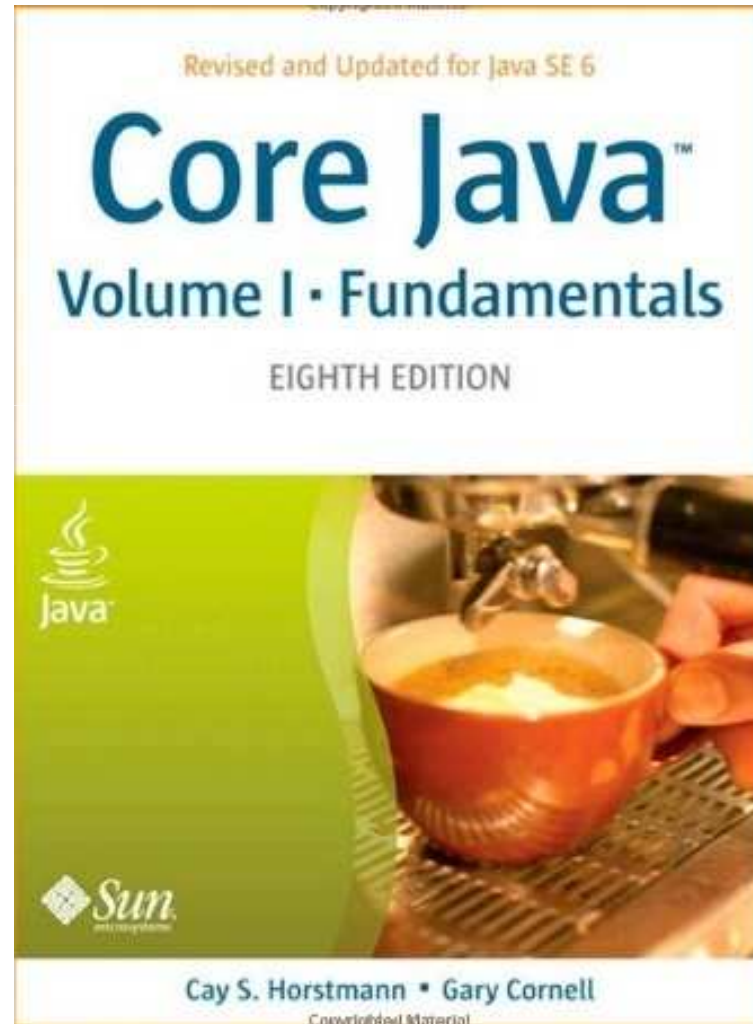
- Tes Kecil 1 – **15** point
- Tes Kecil 2 – **20** point
- Praktikum – **35** point
- Keaktifan kelas – maks **5** point
- TAS – **25** point

# Penilaian

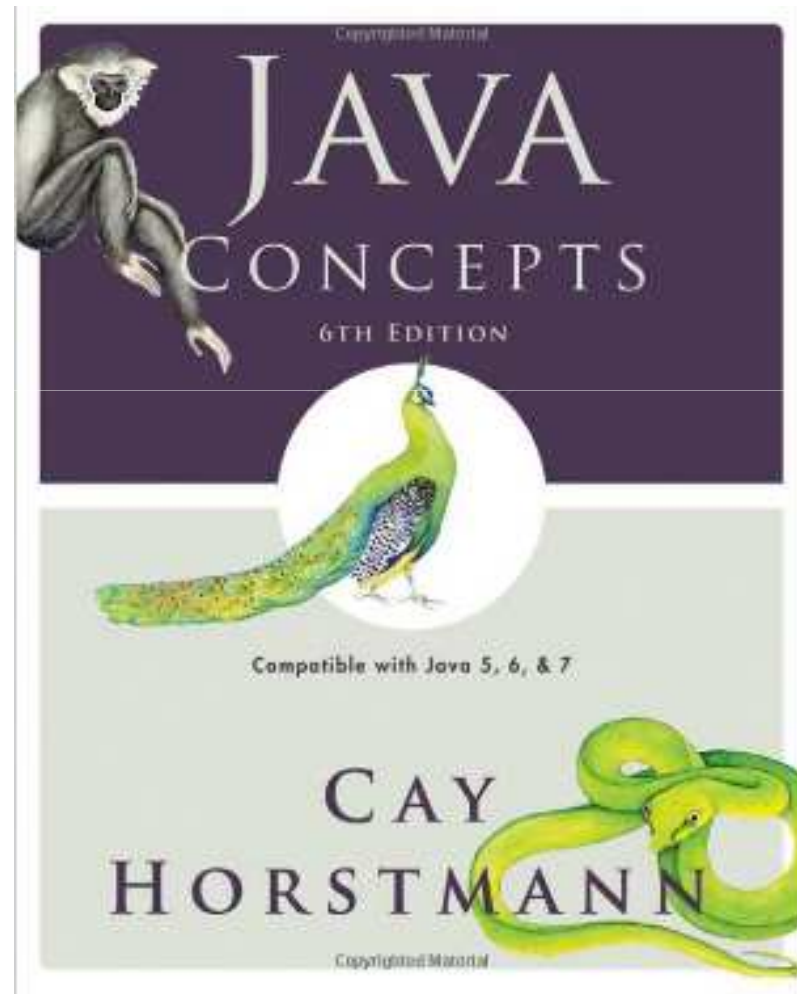
85.0 - 100	A	4.0
80.0 - 84.9	A-	3.7
75.0 - 79.9	B+	3.3
70.0 - 74.9	B	3.0
65.0 - 69.9	B-	2.7
60.0 - 64.9	C+	2.3
55.0 - 59.9	C	2.0
0 - 54.9	E	0.0



# Core Java Volume 1 – 8 Ed

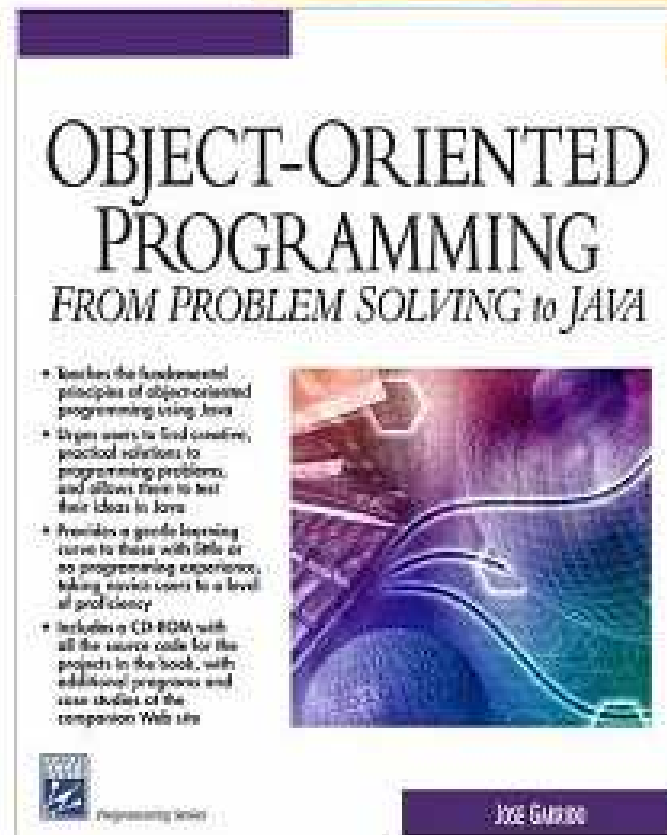


# Java Concepts 6 Ed

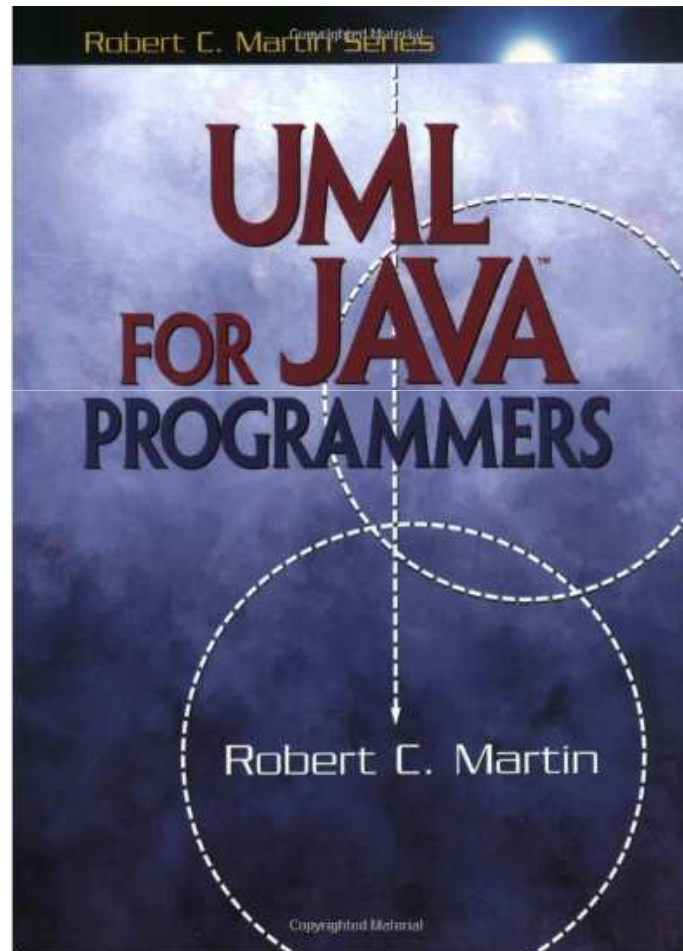


# Object-Oriented Programming

Click to **LOOK INSIDE!**



# UML for Java Programmers



# Peraturan

- Harap tenang selama kuliah berlangsung
- Saling menghargai, saling menghormati
- Berbuat curang = **E**
- Peraturan absensi mengikuti aturan Universitas (**75%**)

# Peraturan Tes

- Tidak ada test susulan untuk Test Kecil 1 dan 2 maupun TAS
- Tidak ada tugas tambahan untuk menambah nilai

# TAS

- Materi dari awal hingga akhir
- Open books
- Bentuk soal: pilihan ganda dan essay!
- **Tidak ada remidi TAS**
- Bobot **25** point

# Praktikum

- Ada tiga kegiatan:
  - Kegiatan Sebelum Praktikum (Pra-Praktikum)
  - Kegiatan Saat Praktikum (Praktikum)
  - Kegiatan Setelah Praktikum (Pasca-Praktikum)



# Remidi

- Untuk TK1 jika point  $\leq 9$
- Nilai maks remidi TK1 = 9
- Untuk TK2 jika point  $\leq 12$
- Nilai maks remidi TK2 = 12

# Tools

- JCreator
  - <http://www.jcreator.com>
- Java Software Development Kit (JDK)
  - <http://java.sun.com/javase/downloads/widget/jdk6.jsp>
- Netbeans 7.x
  - <http://netbeans.org/downloads/index.html>
- Eclipse
  - [www.eclipse.org/downloads](http://www.eclipse.org/downloads)
- UML Editor
  - <http://staruml.sourceforge.net>
  - <http://argouml-downloads.tigris.org>

Q & A

# Pengantar OOP dan Java

# Generasi Bahasa Pemrograman

- Generasi 1: Machine / Binary
- Generasi 2: Assembly
- Generasi 3: (High Level)
  - Java, C++, Pascal, C
- Generasi 4 (Special Purpose Language)
  - Report Generator: Crystal Reports, RAVE
  - Forms Generator: Delphi, VB, C#, FoxPro
  - CASE: Rational Rose, Poseidon
  - DBMS: FoxPro, Oracle, SQL

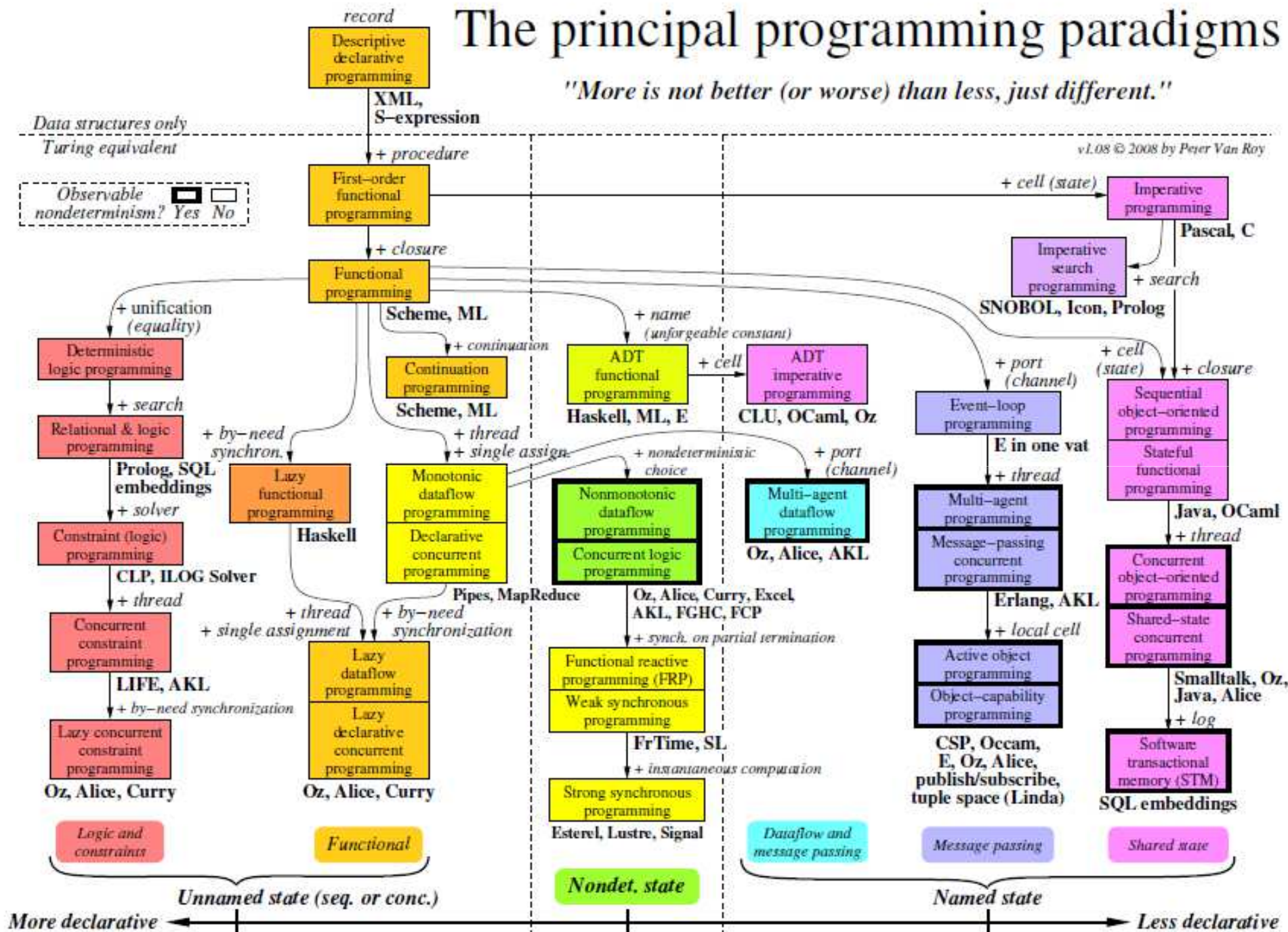
# Paradigma Pemrograman

- Suatu cara berpikir dalam membuat program komputer yang direpresentasikan dalam sejumlah konsep dan teknik pemrograman
- Terdapat banyak paradigma pemrograman
- Suatu bahasa pemrograman bisa mendukung lebih dari satu paradigma pemrograman

# The principal programming paradigms

"More is not better (or worse) than less, just different."

v1.08 © 2008 by Peter Van Roy



# Contoh Bahasa Pemrograman

- Procedural
  - Pascal, C, COBOL, Fortran, ALGOL, Basic, PHP, dll.
- Object Oriented
  - Java, C#, C++, Objective C, PHP, Visual Basic.Net, Object Pascal, dll.
- Pada C++, bisa procedural, bisa juga Object Oriented !



# PBO

- Paradigma pemrograman yg menggunakan pendekatan berorientasi pada obyek
- Jadi permasalahan yang ada dipandang sebagai obyek
- **Obyek** => suatu bentuk nyata yang dapat dibayangkan, memiliki segala sesuatu yang memang melekat padanya, dan dapat melakukan tindakan tertentu
- Paradigma ini juga menghasilkan paradigma pemrograman lain seperti **Component Oriented Programming**

# Procedural / Struktural vs OOP

- **Procedural**

- Menyusun langkah-langkah untuk menyelesaikan suatu masalah
- Misal: menghitung luas bangun segi empat
- Langkahnya:
  - Input panjang dan lebar
  - Cari luas dengan cara kalikan panjang dan lebar
  - Tampilkan luas

# Procedural / Struktural vs OOP

- **Object Oriented**

- Menyusun / merancang **obyek** yang akan dioperasikan
- Segi empat **memiliki** panjang dan lebar
- Segi empat **bisa dihitung luasnya** dengan *panjang x lebar*
- Langkahnya:
  - Buat **obyek** segi empat, isikan data panjang dan lebar
  - **Meminta obyek segi empat** menghitung luasnya

# Contoh obyek “SegiEmpat”

- **Atribut: sifat yg melekat**
  - Panjang
  - Lebar
- **Behaviour: tingkah laku / kegiatan**
  - Hitung Luas
  - Hitung Keliling

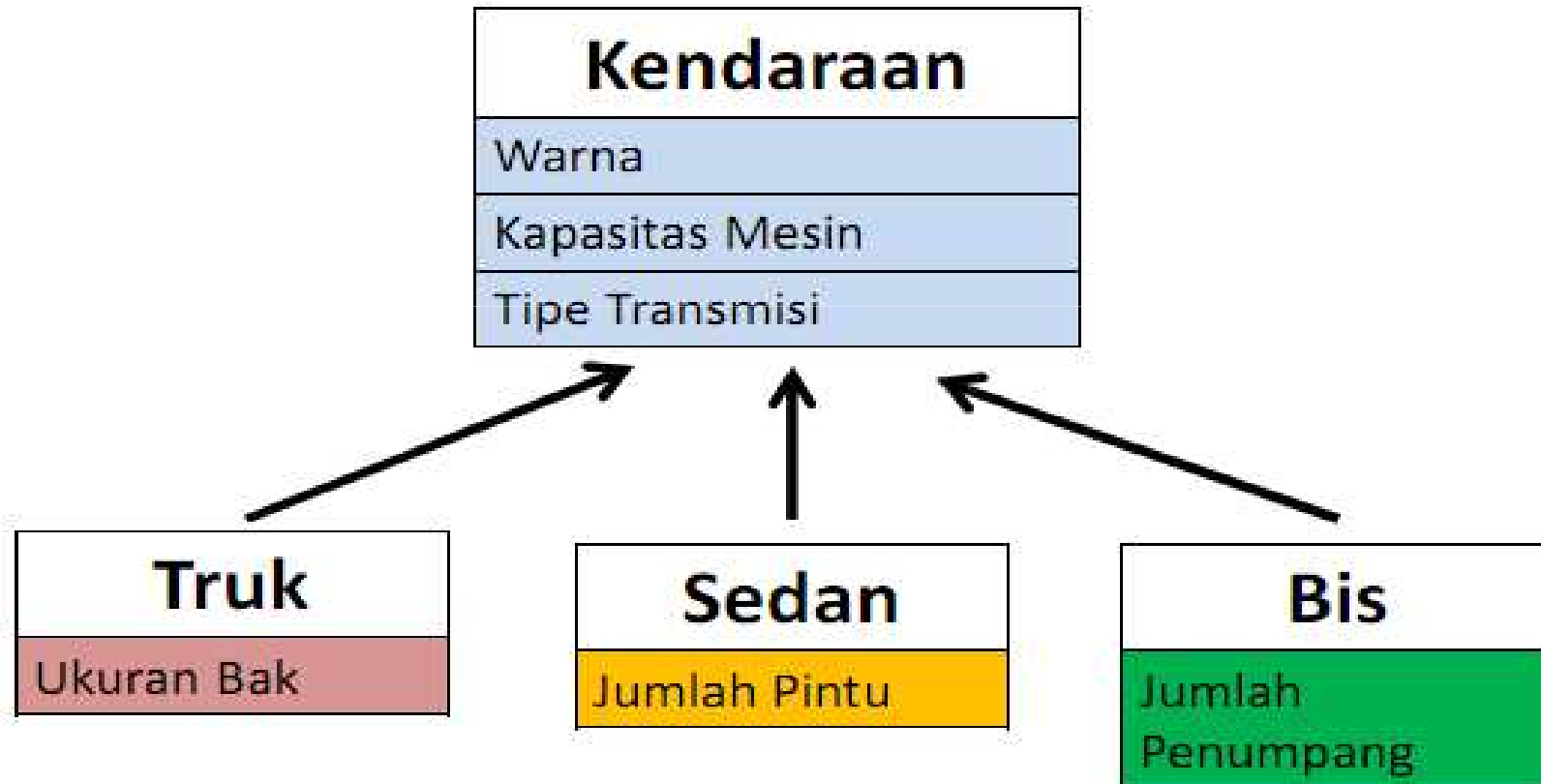
# Obyek dalam PBO

- SegiEmpat merupakan salah satu dari sekian banyak bangun 2 dimensi lainnya
  - Segitiga
  - Lingkaran
  - Belah Ketupat
  - Segilima
  - Segienam
- Kebanyakan **atribut** dan **behaviour** untuk benda 2 dimensi adalah **mirip**
- Namun bisa juga **spesifik** untuk masing-masing bendanya
  - Sangat tergantung bendanya

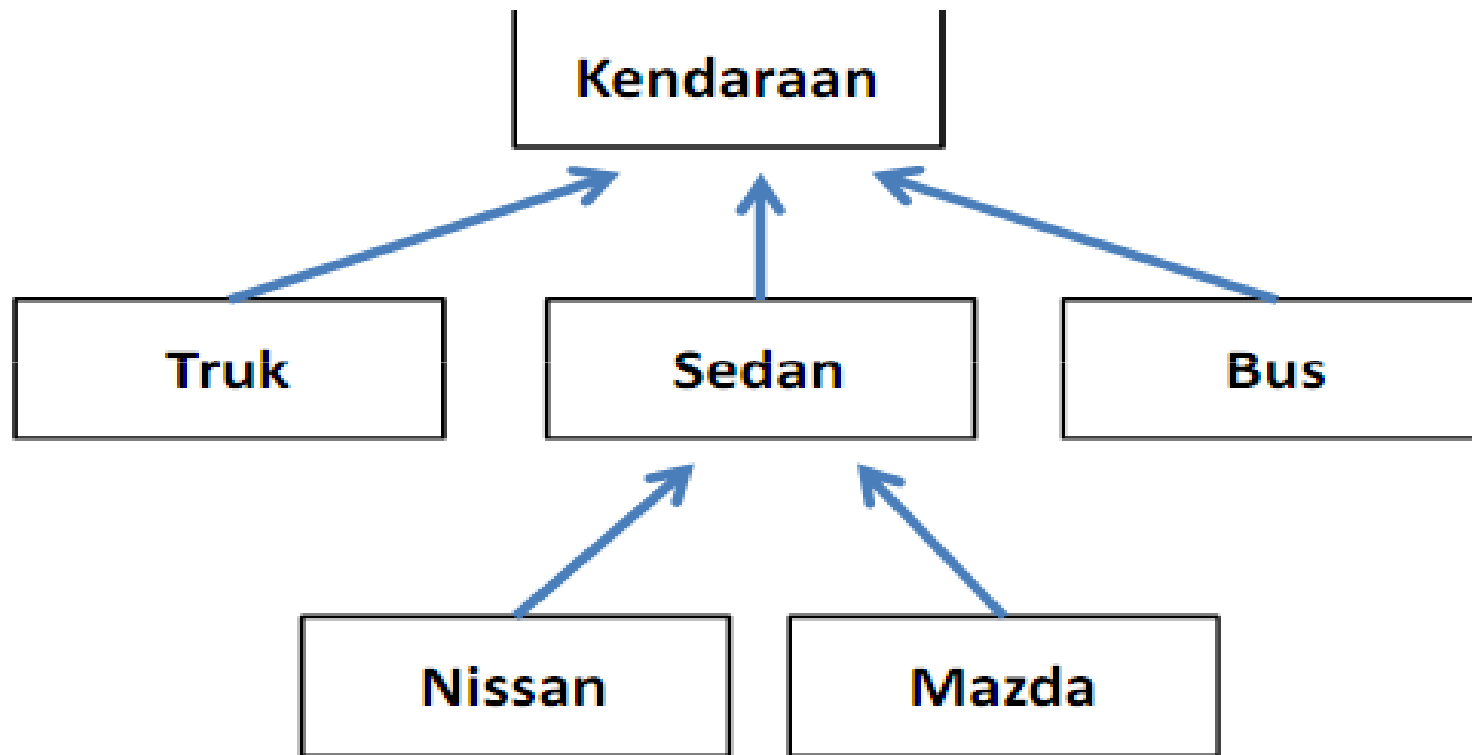
# Contoh Lain OOP : Kendaraan

<b>Truk</b>	<b>Sedan</b>	<b>Bis</b>
Warna	Warna	Warna
Kapasitas Mesin	Kapasitas Mesin	Kapasitas Mesin
Tipe Transmisi	Tipe Transmisi	Tipe Transmisi
Ukuran Bak	Jumlah Pintu	Jumlah Penumpang

# Pemisahan dalam OOP



# Ekstensibilitas dalam OOP





# Procedural / Struktural vs OOP

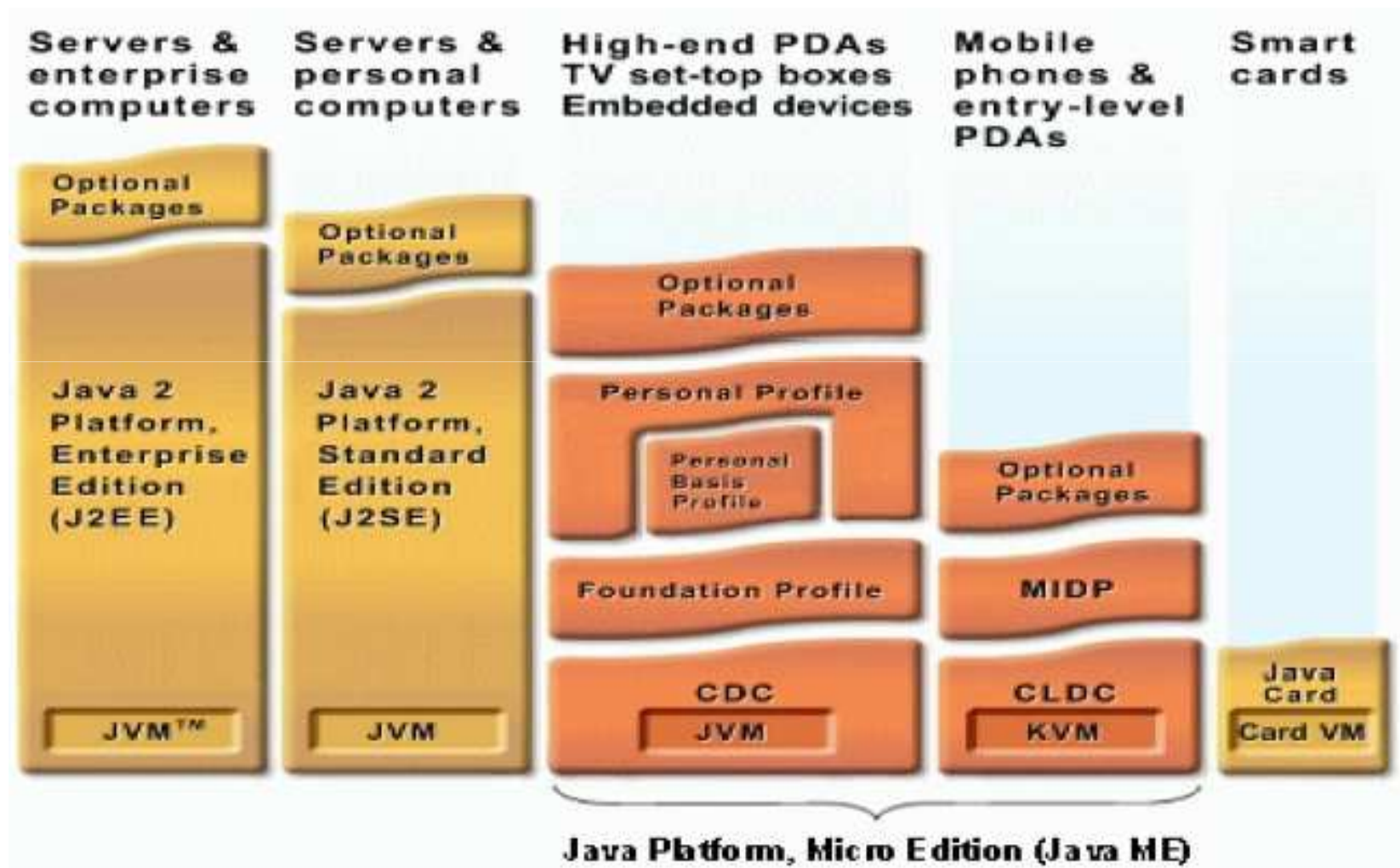
- **Procedural**
  - Lebih cepat untuk memecahkan masalah-masalah berskala kecil
  - Mudah membuatnya
- **Object Oriented**
  - Scalable, cocok untuk masalah-masalah berskala besar
  - Pengembangannya mudah

# Pengenalan Bahasa Java

# Bahasa Java

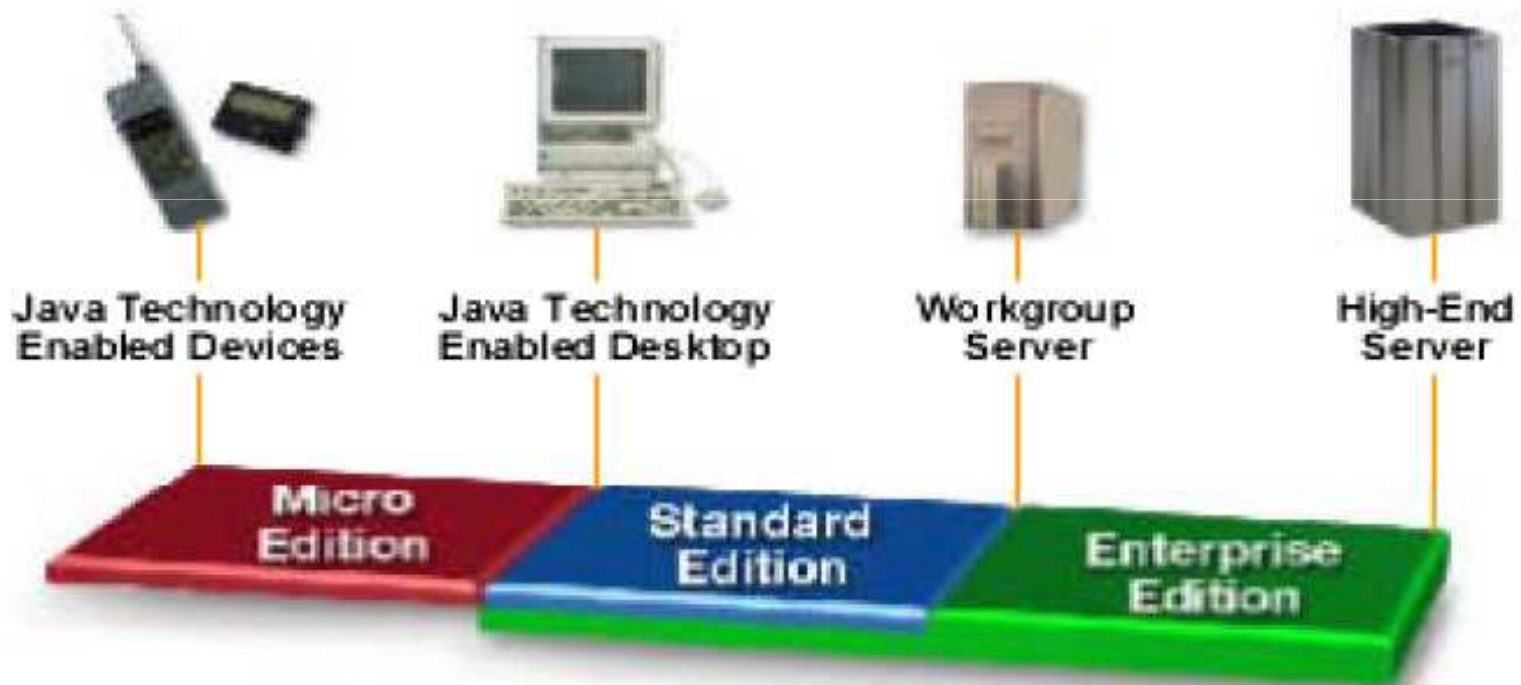
- Dibuat oleh Sun Microsystem
- <http://java.sun.com> menjadi <http://www.oracle.com/technetwork/java/index.html>
- Proyek awal: **Green Project**
  - Bahasa **OAK** oleh James Gosling
  - Kemudian oleh Sun disebut **Java**
- Konsep Java menggunakan OOP
  - Sifatnya: **Write Once Run Everywhere?**
- Mendukung **multiplatform** language

# Jenis Java

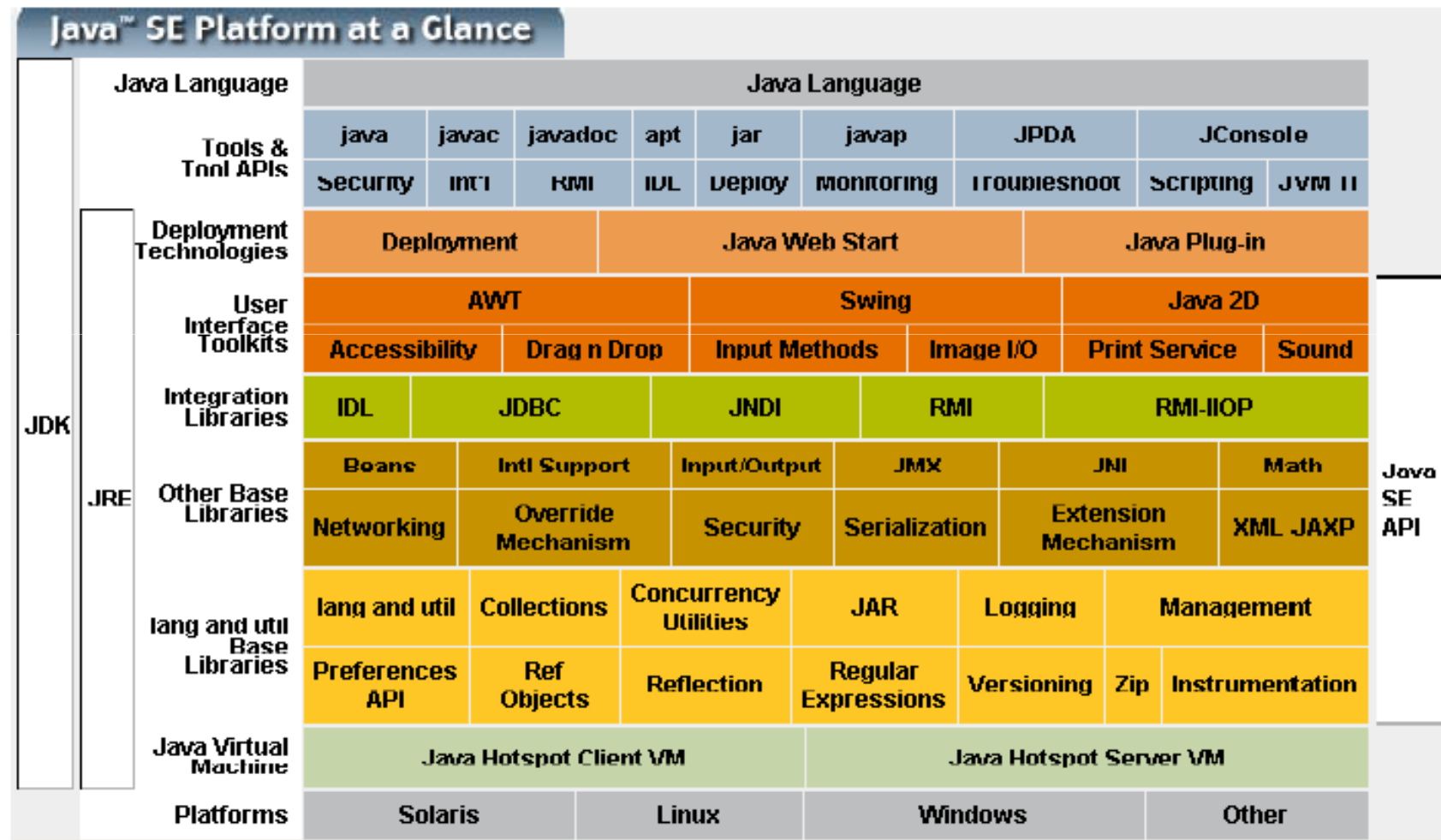


# Java Platform

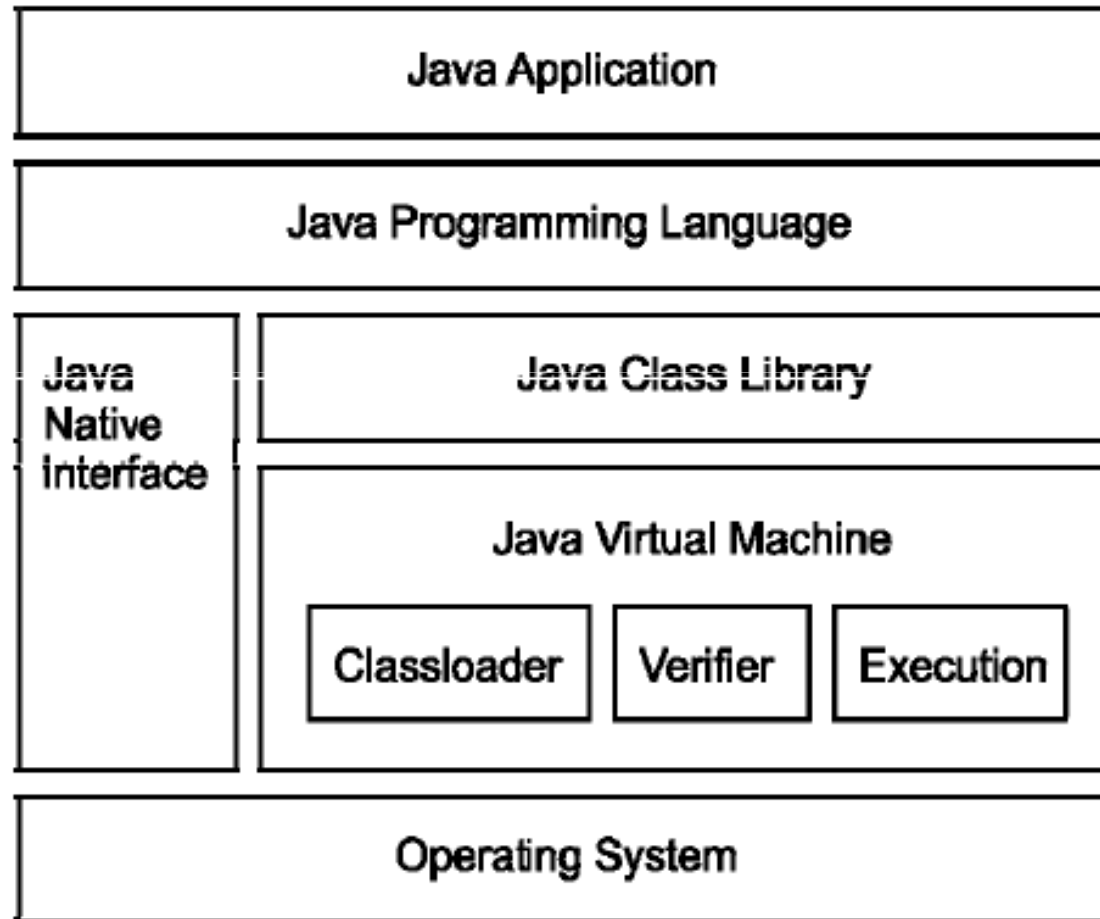
## The Java™ 2 Platform



# Java 2 SE at a glance



# Java system



# Yang dibutuhkan untuk Java

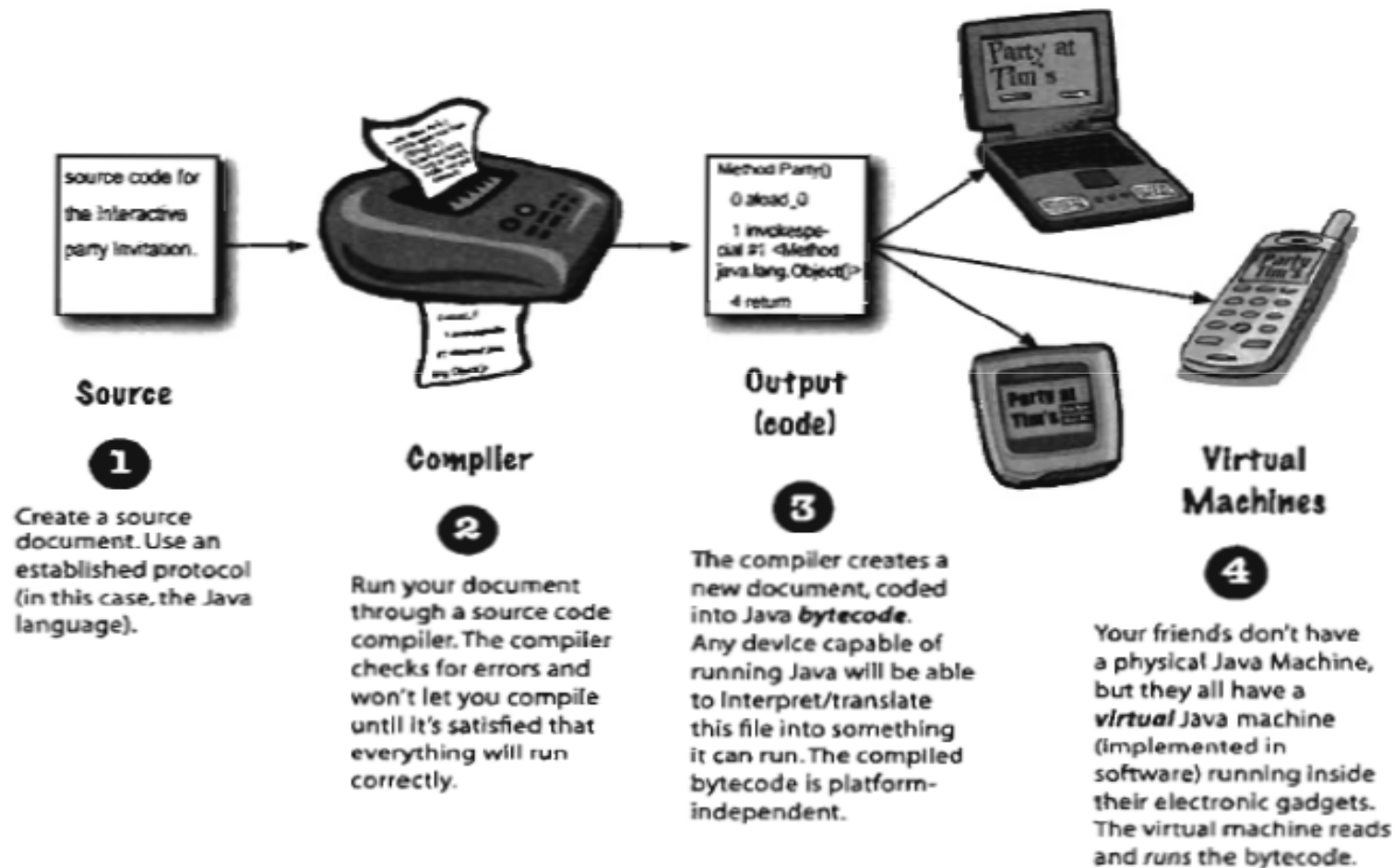
- **J2RE** untuk menjalankan Java.
  - J2RE akan berisi Java Virtual Machine.
- **J2SDK** untuk mendvelop Java.
  - J2SDK akan berisi JVM dan SDK
- **J2ME** untuk mendvelop aplikasi Mobile
- **J2EE** untuk mendvelop aplikasi Enterprise
  - J2EE berisi seluruh komponen Java 2 Platform



# Program penting pada Java

- Javac -> Compiler
- Java -> Interpreter
- Jdb -> Debugger
- Javap -> Disassembler
- Appletviewer -> Penampil applet
- Javadoc -> Pengenerate documentation
- Javah -> Pengenerate header bahasa C

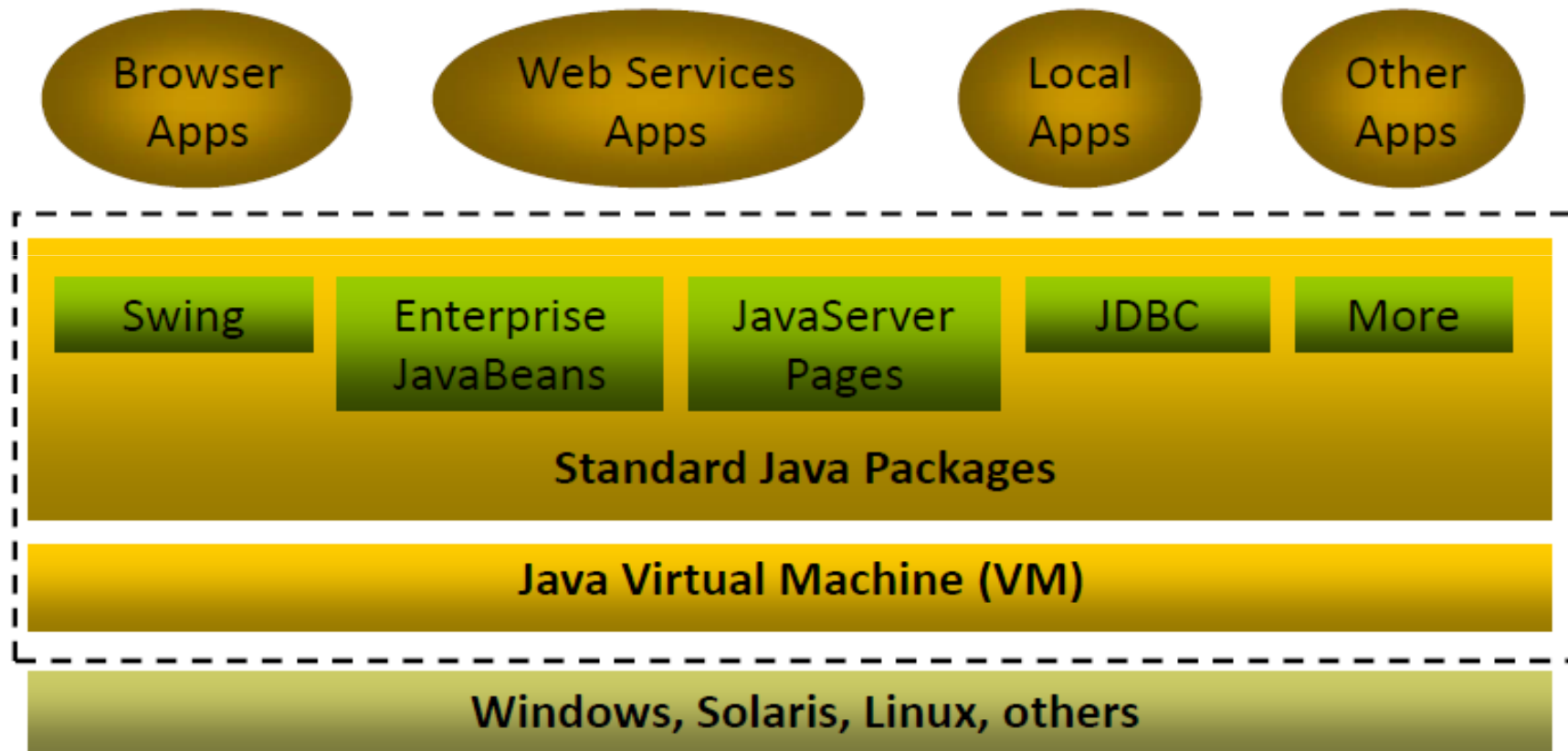
# Bagaimana Java Bekerja



# 4 jenis aplikasi Java

- **Applications:** program standalone di komputer, dari aplikasi console sampai dengan **GUI desktop** yang kompleks yang menggunakan **javax.swing**
- **Applet:** program Java yang dijalankan di **web browser** dengan menggunakan **HTML**.
- **Servlet:** program yang melakukan generating isi webpage namun berjalan di **java-enabled web server** yang kemudian akan dikirimkan hasilnya ke client.
- **JSP:** aplikasi web yang berjalan di sisi **server**
  - Merupakan pengembangan Servlet

# Lingkungan Pengembangan Java



# Next

- Class dan Object I