

PERTEMUAN 1

KONSEP PERANCANGAN PROGRAM

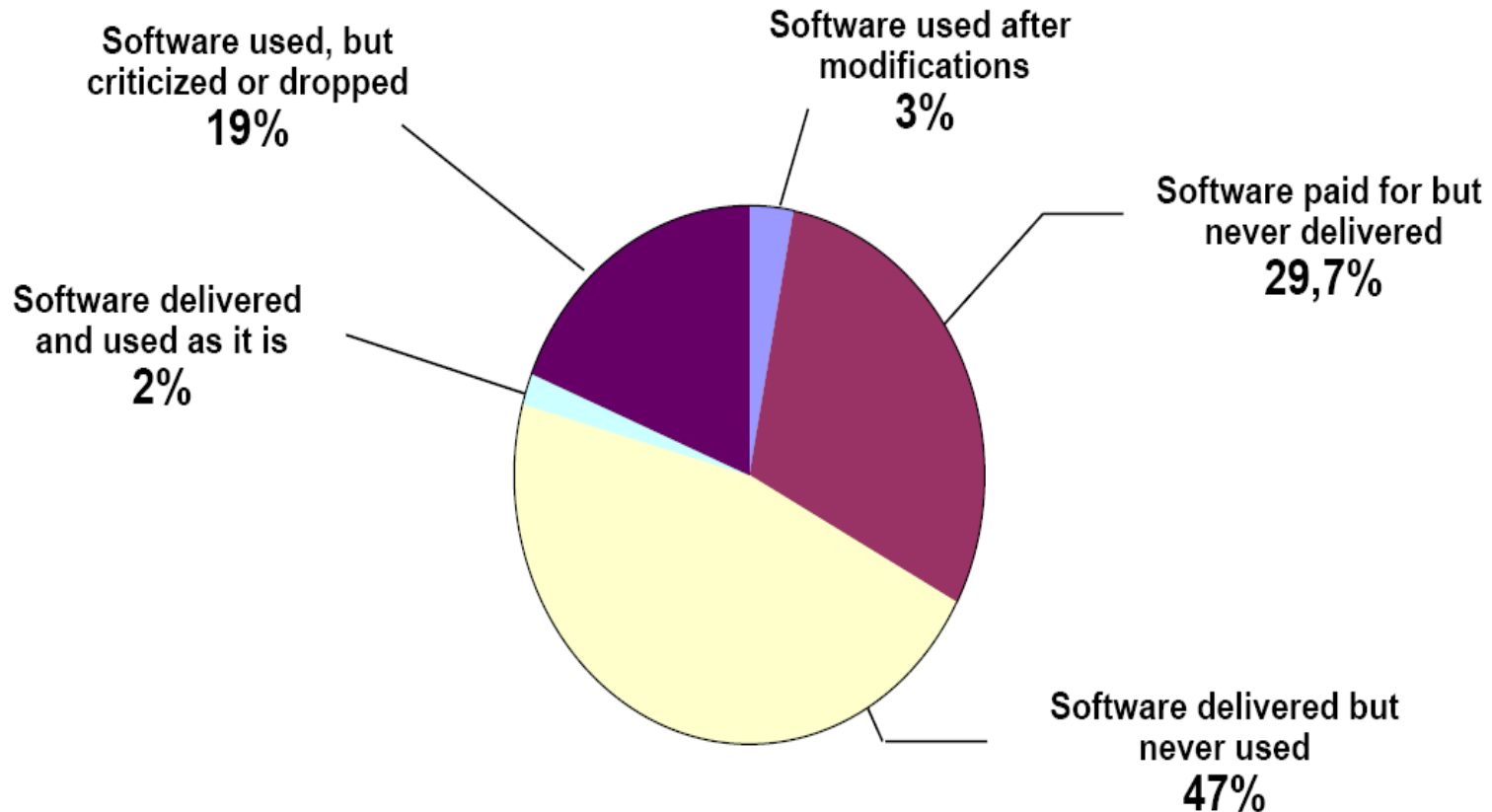
POKOK BAHASAN

1. Pendahuluan
2. Permasalahan Terkait Perangkat Lunak (Program)
3. Program dan Faktor Kualitas Program
4. Software Development Life Cycle
5. Fakta Dalam Pembangunan Program
6. Metodologi Perancangan Program
7. Teknik Pemrograman
8. Paradigma Bahasa Pemrograman
9. Kriteria Bahasa Pemrograman

PENDAHULUAN

- Kebutuhan informasi menjadi unsur utama dalam bisnis sehingga mendorong pertumbuhan industri software.
- Nilai industri software di Indonesia sampai akhir tahun 2013 antara Rp 3.5 –5 Trilyun dengan nilai pertumbuhan 18-20% per tahun (Pratama, 2013).
- Produk perangkat lunak dikembangkan (*developed*) atau direkayasa (*engineered*) sesuai dengan kebutuhan pemakai akhir (end user).

PERMASALAHAN TERKAIT PERANGKAT LUNAK (PROGRAM)



APA PROGRAM ?

Kumpulan instruksi / perintah yang dirangkai sehingga membentuk suatu proses untuk mengolah data.

FAKTOR KUALITAS PROGRAM

1. Correctness : besarnya program dapat memuaskan spesifikasi dan objektivitas dari misi pelanggan.
2. Reliability : besarnya program dapat diharapkan memenuhi fungsi-fungsi yang dikehendaki.
3. Efficiency : jumlah sumber-sumber dan kode yang dibutuhkan program untuk menjalankan fungsi-fungsinya.
4. Integrity : besarnya pengontrolan pengaksesan oleh seorang yang tidak mempunyai otorisasi terhadap program dan data.
5. Usability : usaha yang dibutuhkan untuk mempelajari, mengoperasikan, menyiapkan input & menginterpretasikan output program .

FAKTOR KUALITAS PROGRAM (lanjutan)

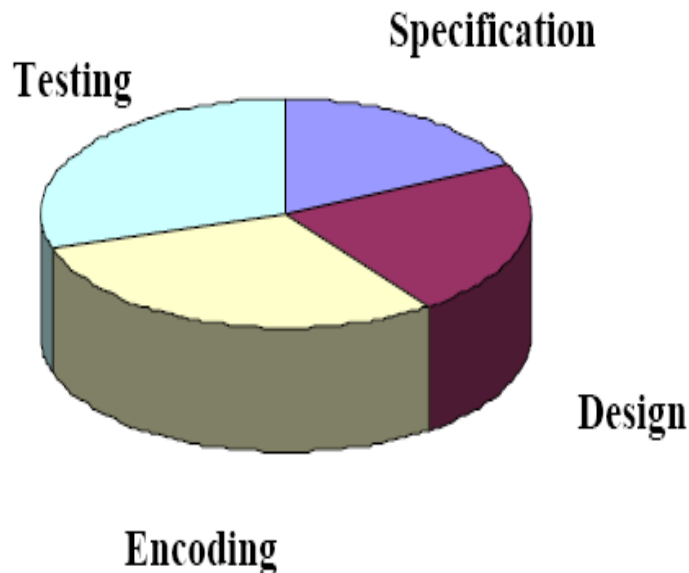
6. Maintability : usaha yang dibutuhkan untuk menempatkan & menetapkan kesalahan pada program.
7. Flexibility : usaha yang dibutuhkan untuk memodifikasi program yang dibutuhkan.
8. Testability : usaha yang dibutuhkan untuk menguji program untuk menjamin program sesuai yang diharapkan.
9. Portability : usaha yang dibutuhkan untuk mnetransfer program dari lingkungan ke lingkungan lain.
10. Reusability : besarnya program dapat digunakan oleh aplikasi lain.

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

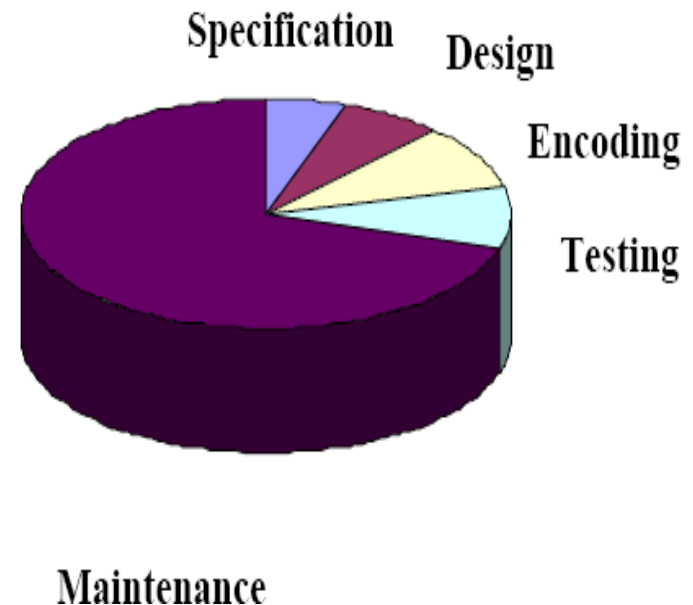
- SDLC adalah serangkaian aktivitas yang dilaksanakan oleh profesional dan pemakai sistem informasi untuk mengembangkan dan mengimplementasikan software (program).
- Aktivitas atau proses standar yang digunakan untuk membangun program, yaitu :
 - Analisa kebutuhan (spesifikasi)
 - Desain
 - Coding
 - Pengujian

FAKTA DALAM PEMBANGUNAN PROGRAM

**Distribution of effort :
what is believed**



**Distribution of effort:
what happens**



METODOLOGI PERANCANGAN PROGRAM

Prinsip dasar dalam pembangunan program berdasarkan input data, proses dan output. Beberapa pendekatan perancangan program :

1. Procedure-Driven
2. Event-Driven
3. Data-Driven

PROCEDURE-DRIVEN

- *Procedure-Driven* untuk perancangan program berdasarkan pada apa yang akan dilakukan program (proses atau fungsi).
- Strateginya adalah dengan memecahkan setiap fungsi menjadi lebih kecil dan lebih spesifik.
- Contoh :
Program untuk persegi panjang dibagi menjadi fungsi : hitung keliling dan hitung luas persegi panjang

EVENT-DRIVEN

- *Event-Driven* untuk perancangan program berdasarkan pada kejadian atau interaksi dari luar dapat menyebabkan perubahan dari satu kondisi ke kondisi lain.
- Keadaan awal dari sebuah program mengidentifikasikan seluruh pemicu yang mewakili kejadian untuk kondisi yang akan dihasilkan.
- Contoh : apabila tombol mulai pada program stopwatch di klik maka program akan menghitung waktu yang berjalan sampai tombol stop di klik.

DATA-DRIVEN

- *Data-Driven* untuk perancangan program berdasarkan struktur data.
- Dimulai dengan analisis data dan hubungan antara data, untuk menentukan struktur data dasar. Setelah struktur data telah ditetapkan, output data yang diperlukan diperiksa dalam rangka menentukan proses apa yang diperlukan untuk mengkonversi data input ke output.

CONTOH *DATA-DRIVEN*

- Untuk membuat program menghitung bonus karyawan. Besar bonus masing-masing karyawan tergantung gaji pokok dan jenis tanggung-jawabnya.
- Data Input : Data Profil Karyawan (seperti : nik, nama, jabatan, gaji pokok)
- Data Output : Bonus karyawan
- Proses :

Ambil tanggung_jawab dari profil_karyawan

If tanggung_jawab = "Sales manager"

Then bonus = gaji_pokok * 0.10

Else

If tanggung_jawab = "Senior salesman"

Then bonus = gaji_pokok * 0.08

Else ... dan seterusnya ...

BELAJAR PEMROGRAMAN ?

- Pemrograman adalah proses menulis, menguji dan memperbaiki (debug), dan memelihara kode yang membangun sebuah program komputer.
- Belajar pemrograman adalah belajar tentang metodologi pemecahan masalah kemudian menuangkannya dalam suatu notasi yang mudah dibaca dan dipahami.

TEKNIK PEMROGRAMAN

- Permasalahan bisnis yang semakin kompleks dan rumit sedangkan waktu yang diberikan untuk pengembangan program semakin cepat.
- Dibutuhkan teknik pemrograman untuk mengembangkan program yang kompleks dan rumit dalam waktu yang cepat.

TEKNIK PEMROGRAMAN (lanjutan)

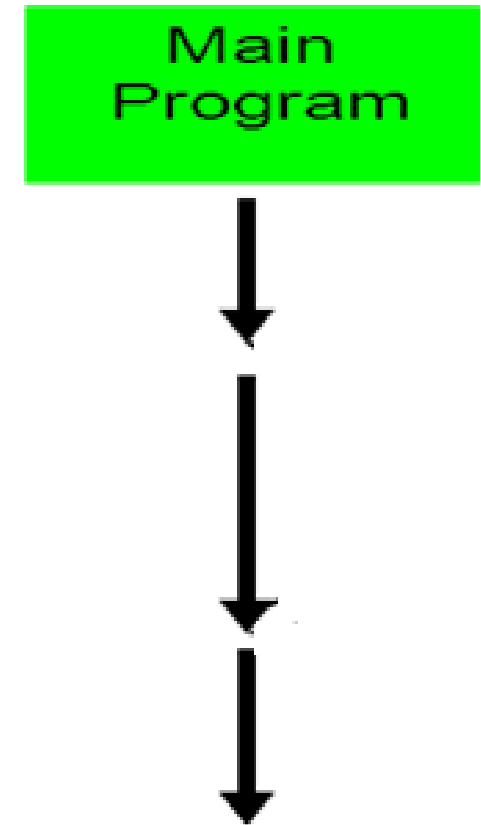
1. Pemrograman Tidak Terstruktur
2. Pemrograman Terstruktur
3. Pemrograman Prosedural
4. Pemrograman Modular
5. Pemrograman Berorientasi Objek

1. PEMROGRAMAN TIDAK TERSTRUKTUR

- Pemrograman tidak terstruktur adalah suatu teknik pemrograman dimana penulisan instruksi tidak teratur dan sistematis sehingga sulit untuk dipahami.
- Contoh : Penggunaan perintah GOTO yang banyak dalam program.

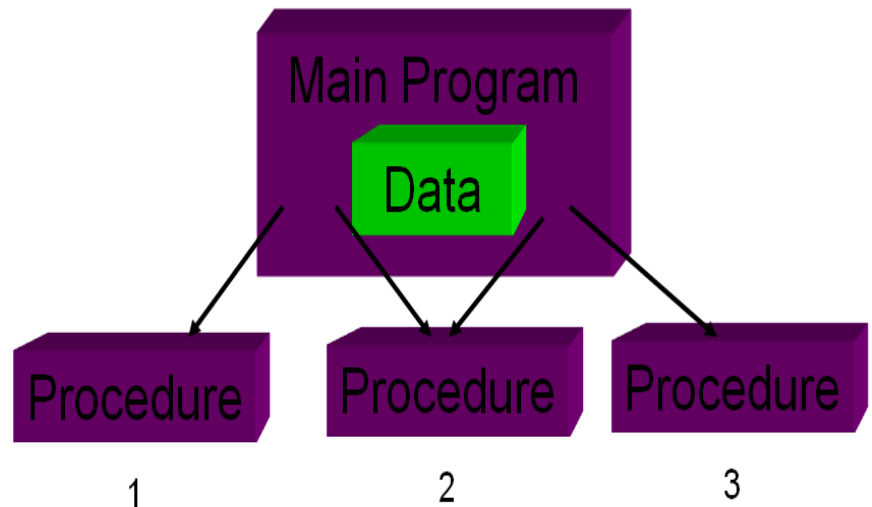
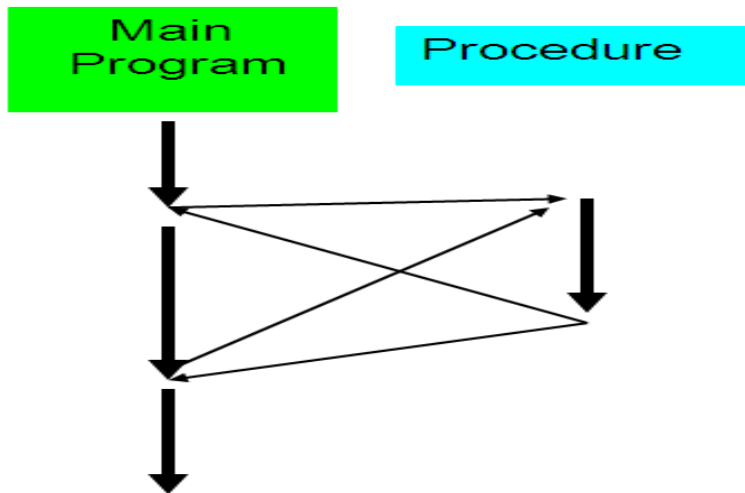
2. PEMROGRAMAN TERSTRUKTUR

- Pemrograman terstruktur adalah suatu teknik pemrograman dimana penulisan instruksi secara sistematis, logis dan tersusun berdasarkan algoritma yang sederhana dan mudah dipahami.
- Prinsip pada pemrograman terstruktur jika suatu proses sudah sampai pada titik tertentu, maka proses selanjutnya tidak bisa melakukan proses pada baris sebelumnya.



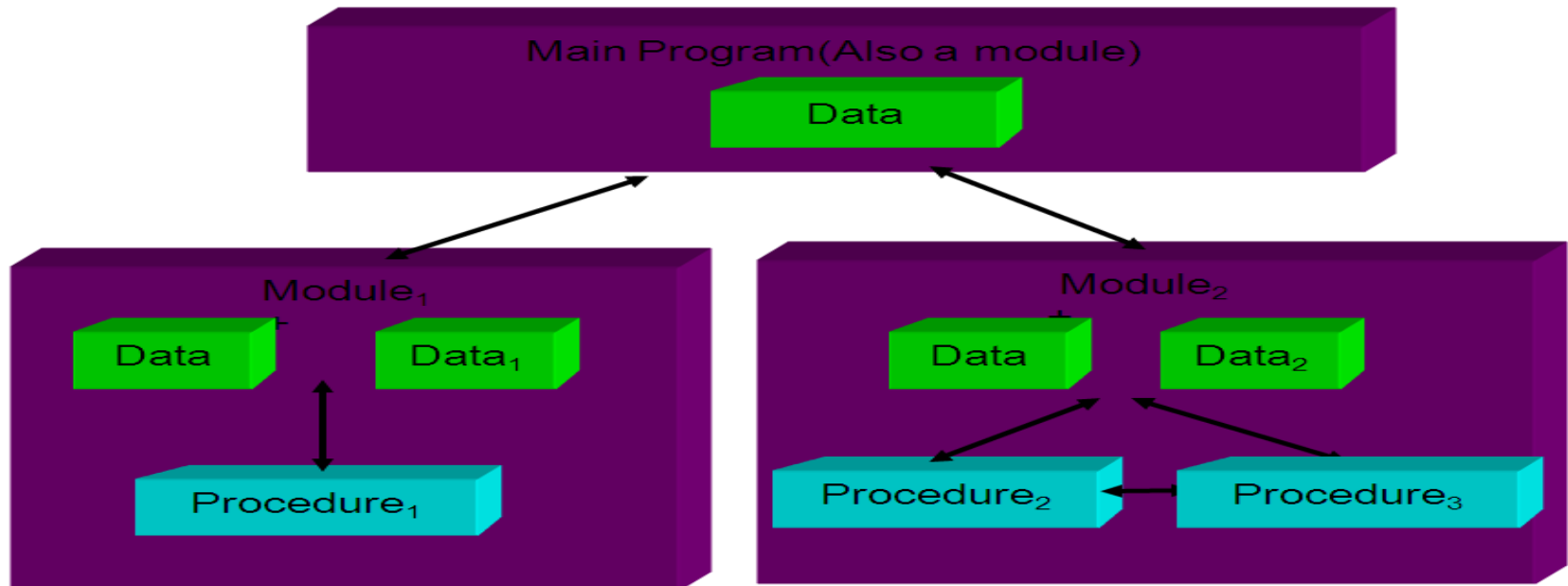
3. PEMROGRAMAN PROSEDURAL

- Pemrograman prosedural adalah suatu teknik pemrograman dimana penulisan instruksi yang sama dibuat dalam sub program tersendiri sehingga pembuatan program lebih cepat dan memudahkan perbaikan apabila terjadi kesalahan.



4. PEMROGRAMAN MODULAR

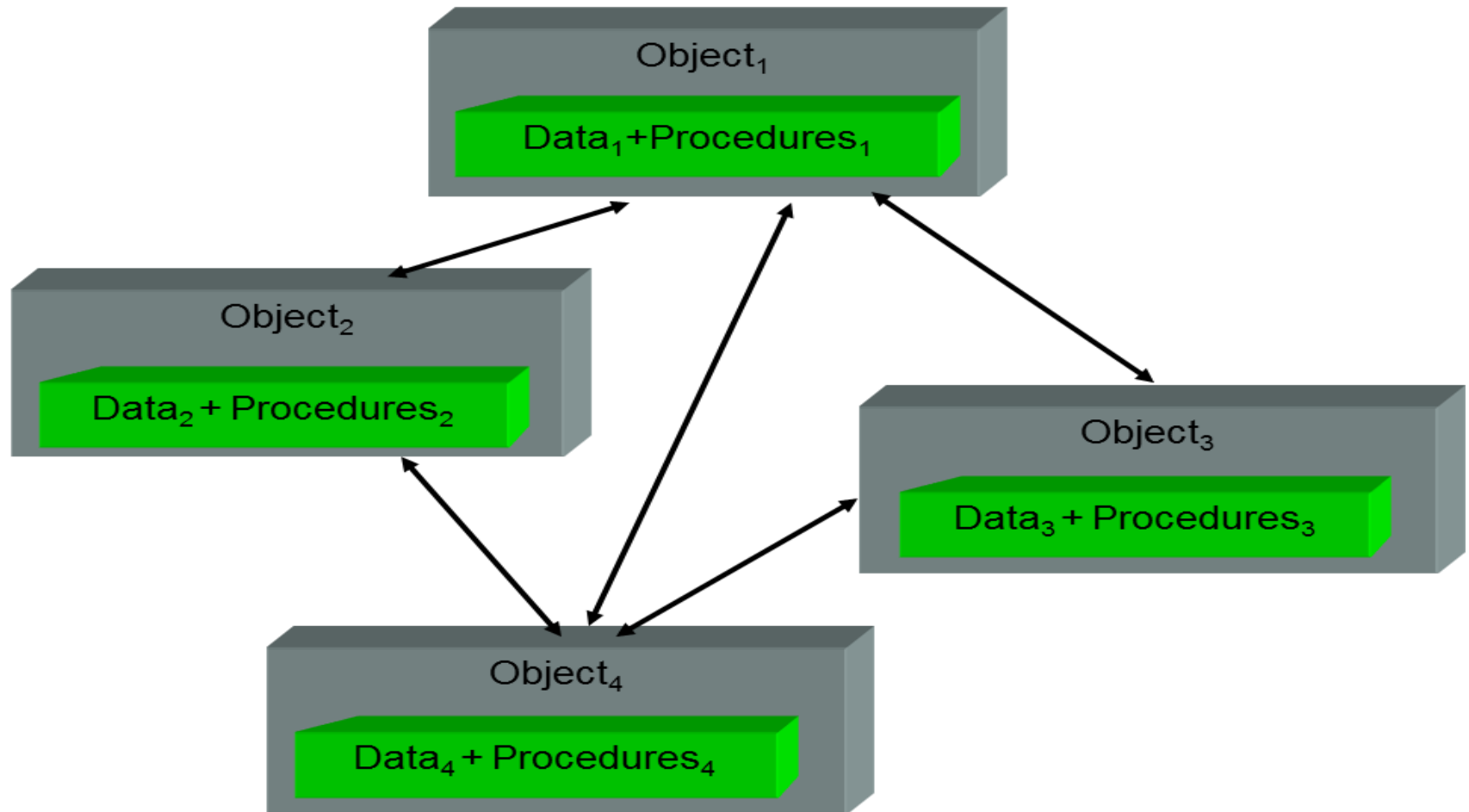
- Pemrograman modular adalah suatu teknik pemrograman dimana prosedur dari fungsi umum dikelompokkan dalam modul-modul dan setiap modul menunjukkan fungsi dan tugas tunggal.



5. PEMROGRAMAN BERORIENTASI OBJEK

- Pemrograman berbasis obyek (OOP) mendasarkan pada konsep **objek dan interaksinya**.
- Objek berasal dari tipe data abstrak.
- Objek dapat menerima pesan (*message*), mengolah data, dan mengirimkan pesan ke object lain → membentuk interaksi antar object.
- Objek bersifat independen: tiap object dapat dipandang sebagai sebuah entitas mandiri yang memiliki peran atau tanggung jawab tertentu.

5. PEMROGRAMAN BERORIENTASI OBJEK (lanjutan)



BELAJAR BAHASA PEMROGRAMAN ?

- Bahasa pemrograman merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer.
- Belajar bahasa pemrograman adalah belajar memakai suatu bahasa pemrograman sesuai dengan aturan tata bahasanya.

TIGA FAKTOR PENTING DALAM BAHASA PEMROGRAMAN

1. Sintaks
2. Semantik
3. Kebenaran Logika

SINTAKS

Sintaks adalah **aturan penulisan** suatu bahasa pemrograman (tata bahasanya).

Contoh :

```
Public class MyClass {  
    public static void main(string[] args) {  
        //pernyataan  
        system.out.println("Hello world");  
    }  
}
```

Apabila terjadi kesalahan dalam penulisan sintaks maka akan terjadi error pada saat kompilasi.

Penjelasan contoh

Setiap baris kode di Java harus berada di dalam Class, dalam contoh nama class nya **MyClass**. Nama class di Java harus selalu dimulai dengan huruf besar.

Catatan: java bersifat case-sensitive: “MyClass” dan “myclass” memiliki arti yang berbeda.

SEMANTIK

Semantik adalah **arti atau maksud** yang terkandung didalam **statement** tersebut.

Contoh :

- *\n; arti semantiknya baris baru.*
- *Printf(); arti semantiknya mencetak string ke layar.*
- *Getch(); arti semantiknya membaca sebuah karakter.*

KEBENARAN LOGIKA

Kebenaran logika adalah berhubungan dengan **benar tidaknya urutan statement**.

Contoh :

```
Public void infiniteloop()
{
    while (true) {
        system.out.println("Hello world");
    }
}
```

Output dari program di atas terjadi perulangan terus menerus yang tidak berakhir.

RESUME

SINTAKS	SEMANTIK	KEBENARAN LOGIKA
aturan penulisan suatu bahasa pemograman (tata bahasanya)	arti atau maksud yang terkandung didalam statement tersebut	berhubungan dengan benar tidaknya urutan statement

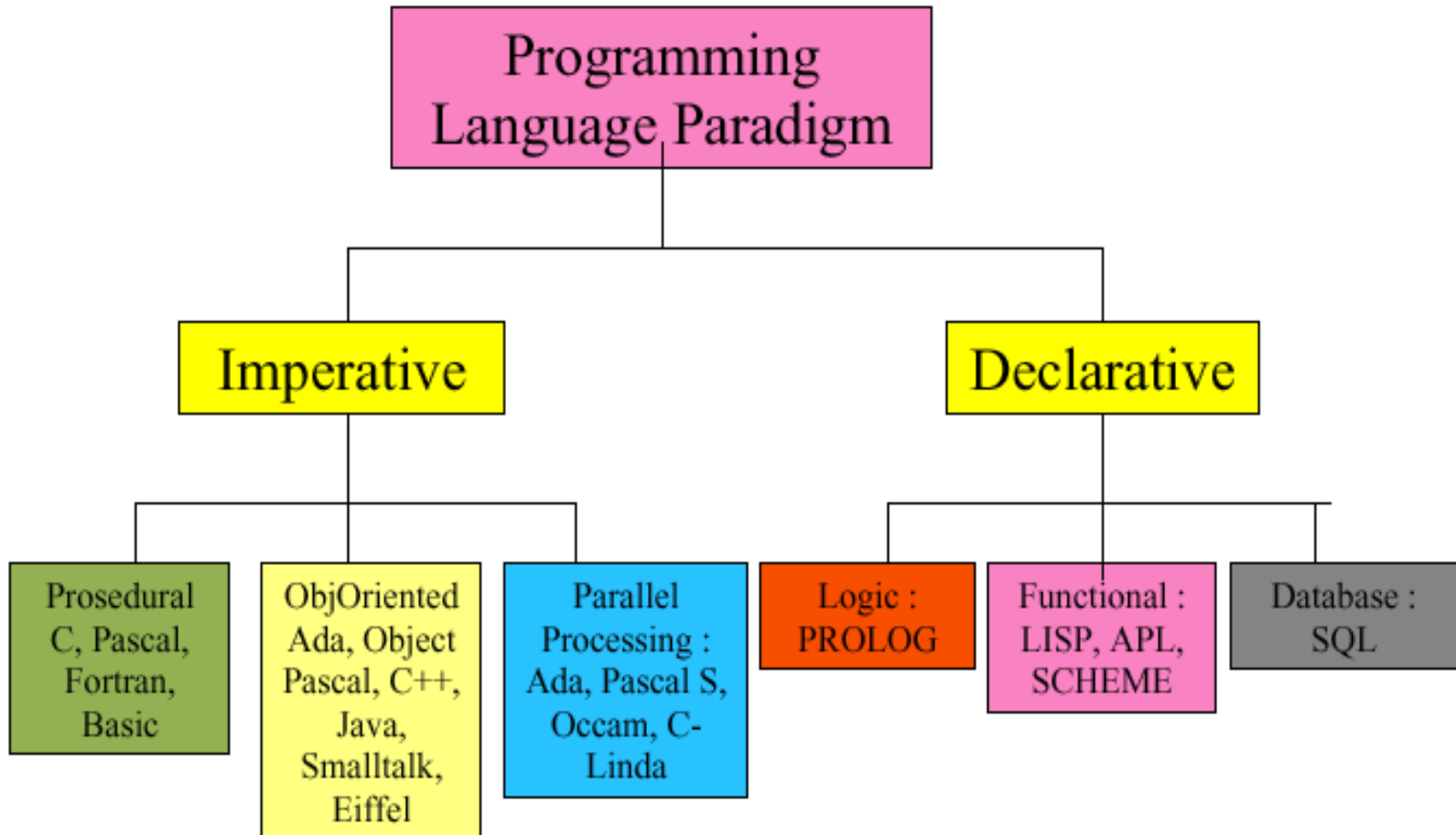
JENIS-JENIS BAHASA PEMROGRAMAN

1. Bahasa Tingkat Rendah
 - Dirancang agar setiap instruksi langsung dikerjakan oleh komputer tanpa translator.
 - Contoh : Assembler
2. Bahasa Tingkat Tinggi
 - Bahasa Pemrograman yang dalam penulisan pernyataannya mudah dipahami secara langsung.
 - Perlu diterjemahkan oleh translator bahasa
 - Contoh : Algol, Fortran, Pascal, Visual Basic, Oracle, dll.

CONTOH

BAHASA TINGKAT RENDAH	BAHASA TINGKAT TINGGI
//Menampilkan kalimat "Hello world" //menggunakan Assembler	//Menampilkan kalimat "Hello world" //menggunakan Pascal
<pre>org 0x100 ; mov dx, msg ; mov ah, 9 ; int 0x21 ; mov ah, 0x4c ; int 0x21 ; msg db 'Hello, World!', 0x0d, 0x0a, '\$' ;</pre>	<pre>PROGRAM hello_world; USES CRT; BEGIN WRITE('Hello World'); READLN; END.</pre>

PARADIGMA BAHASA PEMROGRAMAN



KRITERIA BAHASA PEMROGRAMAN

1. **Clarity, simplicity dan unity**

Kemudahan, kesederhanaan dan kesatuan merupakan suatu kombinasi yang membantu programmer mengembangkan suatu algoritma.

2. **Orthogonality**

Merupakan suatu atribut yang dapat dikombinasikan dengan beragam fitur bahasa pemrograman sehingga setiap kombinasinya mempunyai arti dan dapat digunakan.

KRITERIA BAHASA

PEMROGRAMAN (lanjutan)

3. Kewajaran untuk aplikasi

Bahasa pemrograman harus mempunyai struktur data, operasi-operasi, struktur kontrol dan sintaks yang tepat untuk memecahkan suatu masalah.

4. Mendukung Abstraksi

Abstraksi merupakan suatu hal yang substansial bagi programmer untuk membuat suatu solusi dari masalah yang dihadapi.

KRITERIA BAHASA PEMROGRAMAN (lanjutan)

5. Kemudahan untuk verifikasi program

Dengan verifikasi data yang mudah, maka suatu program akan dengan mudah dibangun dan dikembangkan.

6. Lingkungan pemrograman

Lingkungan pemrograman dapat berarti editor yang digunakan, dokumentasi yang baik, fasilitas debugging, user interface yang baik ataupun tools lain yang dapat digunakan untuk memudahkan pekerjaan programmer.

KRITERIA BAHASA PEMROGRAMAN (lanjutan)

7. Portabilitas program

Kemudahan program untuk dipakai di berbagai jenis komputer.

8. Biaya penggunaan

- Biaya eksekusi program
- Biaya kompilasi program
- Biaya penciptaan, testing dan penggunaan program
- Biaya pemeliharaan program

.

DISKUSI

1. Apa penyebab terjadinya permasalahan dalam perangkat lunak (program) yang dihasilkan.
2. Dalam mempelajari bahasa pemrograman apa yang pertama kali harus dipelajari sehingga dapat melakukan pemrograman ?

SOAL LATIHAN

1. usaha yang dibutuhkan untuk mentransfer program dari lingkungan ke lingkungan lain.

- a. Maintability
- b. Flexibility
- c. Testability
- d. Portability
- e. Reusability

2. Kumpulan instruksi / perintah yang dirangkaian sehingga membentuk suatu proses untuk mengolah data adalah

- a. Program
- b. Instruksi
- c. Sistem
- d. Pemrograman
- e. Database

SOAL LATIHAN (lanjutan)

3. Perancangan program berdasarkan pada apa yang akan dilakukan program (proses atau fungsi), dikenal dengan pendekatan perancangan program:

- a. Event-Driven
- b. Structure-Driven
- c. Procedure-Driven
- d. Logical-Driven
- e. Data-Driven

4. Bahasa Tingkat Rendah (Low Level Language) dirancang agar setiap instruksi langsung dikerjakan oleh komputer tanpa translator, yang termasuk dalamnya adalah:

- a. Repeater
- b. Assembler
- c. Basic
- d. Pascal
- e. Fortra

SOAL LATIHAN (lanjutan)

5. Kemudahan, kesederhanaan dan kesatuan merupakan suatu kombinasi yang membantu programmer mengembangkan suatu algoritma, termasuk dalam kriteria bahasa pemrograman:
 - a. Clarity
 - b. Orthogonality
 - c. Abstracty
 - d. Verificaty
 - e. Diversificaty