

# BAHASA PEMROGRAMAN

TIM PENGAJAR PEMROGRAMAN  
Departemen Ilmu Komputer IPB

Pertemuan 1

# IDENTITAS MATA KULIAH

Nama Mata Kuliah	:	BAHASA PEMROGRAMAN
Kode Mata Kuliah	:	KOM204
Koordinator	:	Julio Adisantoso (JAS)
Semester	:	Genap 2017/2018
Prasyarat	:	KOM200 - Dasar Pemrograman
Pengajar	:	JAS,HKH,DAR,ANN,HRS,AAS,SHW
PJ Praktikum	:	HKH/DAR
Jadwal Kuliah	:	K1: Selasa (10:00-11:40) RKU 2.0 K2: Rabu (08:00-09:40) TL TPB-2.3
Jadwal Praktikum	:	Senin, 13.00-15.00, Lab 1 + Lab 2

# PENENTUAN NILAI AKHIR

- Praktikum, tugas, dan kuis : 10% (*jika sesuai standard*)
- UTS dan UAS Tertulis : 60-70%
- UTS dan UAS Praktikum : 30%
- Catatan: **Tidak ada ujian perbaikan**

# PERANGKAT PERKULIAHAN

- Peserta: Mahasiswa Mayor ILKOM
- Situs latihan dan praktikum: <http://lx.apps.cs.ipb.ac.id>
- Site material elektronik (resources) <http://lms.ipb.ac.id>
- Bahasa Pemrograman : Prolog, C++, dan Java

# REFERENSI UTAMA

- ❶ Programming Languages: Principles and Paradigms. 2<sup>nd</sup> Ed. Allen B. Tucker and Robert E. Noonan.
- ❷ Concepts of Programming Languages. 10<sup>th</sup> Ed. Robert W. Sebesta.
- ❸ C: How to Program. 3<sup>rd</sup> Ed. Deitel, H. M. and P. J. Deitel. Pearson Education Prentice Hall. Prentice-Hall, New Jersey.
- ❹ Java: How to Program. 3<sup>rd</sup> Ed. Deitel, H. M. and P. J. Deitel. Pearson Education Prentice Hall. Prentice-Hall, New Jersey
- ❺ Logical Programming, Lecture Notes by Benjamin J. Keller

# TATA TERTIB

- Kehadiran paling lambat 15 menit setelah dosen masuk kelas/lab.  
**TERLAMBAT, TIDAK DAPAT MASUK KELAS, DAN DIANGGAP TIDAK HADIR.**
- Berpakaian sesuai ketentuan TaTib IPB
- Minimum kehadiran 11 kali masing-masing untuk kuliah dan praktikum (syarat untuk UAS)
- Handphone harus dinonaktifkan (*silent*), no chatting/BBM/FB/etc saat kuliah maupun praktikum
- Tidak diperbolehkan mengoperasikan laptop saat kuliah, kecuali jika diperintahkan oleh dosen pengajar.

# KEJUJURAN AKADEMIK

- Setiap KECURANGAN akan diberikan imbalan nilai 0 (**huruf mutu E**) pada mata kuliah ini
  - Menyontek ataupun bekerja sama pada saat ujian atau kuis
  - Menyalin tugas hasil pekerjaan pihak lain (sering terjadi pada tugas praktikum)
  - Titip tanda tangan kehadiran
- Imbalan (sanksi) akan diberikan untuk si pelaku maupun yang memberikan kesempatan

# TOP 10 SKILLS

## in 2020

- Complex problem solving
- Critical Thinking
- Creativity
- People Management
- Coordinating with Others
- Emotional Intelligence
- Judgement and Decision Making
- Service Orientation
- Negotiation
- Cognitive Flexibility

## in 2015

- Complex problem solving
- Coordinating with Others
- People Management
- Critical Thinking
- Negotiation
- Quality Control
- Service Orientation
- Judgement and Decision Making
- Active Listening
- Creativity



# STUDENT STANDARDS ([www.iste.org](http://www.iste.org))

- Empowered Learner
- Digital Citizen
- Knowledge Constructor
- Innovative Designer
- Computational Thinker
- Creative Communicator
- Global Collaborator

# ILMU KOMPUTER vs TIK

- Ilmu Komputer (CS:Computer Science) atau Informatika (IF) dan TIK (Teknologi Informasi dan Komunikasi) saling melengkapi, tetapi **tidak sama**.
- CS/IF adalah sebuah disiplin ilmu yang mencari pemahaman dan mengeksplorasi dunia di sekitar kita, baik yang natural maupun artifisial. CS/IF secara khusus walau tidak eksklusif berkaitan dengan studi, perancangan, dan implementasi dari sistem komputer, dan pemahaman terhadap prinsip yang mendasari perancangan tersebut.
- TIK berkaitan dengan kegunaan dari sistem komputer untuk memecahkan persoalan dunia-nyata (real-world problem), termasuk untuk kebutuhan bisnis, spesifikasi dan instalasi dari HW dan SW, serta evaluasi dari dayagunanya. TIK adalah penggunaan teknologi secara produktif, kreatif dan penggunaan eksploratif.

# MENGAPA BELAJAR BAHASA PEMROGRAMAN?

(Sebesta, 2010)

- Increased capacity to express ideas.
- Improved background for choosing appropriate languages.
- Increased ability to learn new languages.
- Better understanding of the significance of implementation.
- Better use of languages that are already known.
- Overall advancement of computing.

# PROGRAMMING DOMAIN

Komputer telah dimanfaatkan di hampir setiap sudut kehidupan, mulai dari mengendalikan pembangkit listrik tenaga nuklir hingga menangani semua fasilitas ponsel.

- Scientific Applications
- Business Applications
- Artificial Intelligence
- Systems Programming
- Web Software

# BAHASA PEMROGRAMAN

- Bahasa pemrograman adalah suatu **sistem notasi** untuk menuliskan tugas komputasi yang harus dilaksanakan oleh mesin, dan bentuknya dapat dibaca serta dipahami oleh manusia.
- Beberapa bahasa komputer dirancang untuk memfasilitasi operasi-operasi tertentu, misalnya komputasi numerik, manipulasi teks, I/O, etc.
- Pada umumnya, bahasa pemrograman komputer biasanya dirancang dengan menggunakan **paradigma pemrograman** tertentu. Artinya mengikuti aliran atau *genre* tertentu.

# PRINSIP BAHASA PEMROGRAMAN

Prinsip perancangan bahasa pemrograman: (1) Sintaks, (2) Nama dan Tipe, (3) Semantik.

- **Sintaks** menjelaskan bagaimana struktur program yang benar. Struktur bahasa pemrograman modern didefinisikan menggunakan bahasa formal yang disebut *context-free-grammar*.
- **Nama dan Tipe** menunjukkan bagaimana aturan penamaan entitas (variabel, fungsi, class, parameter, dsb).
- **Semantik**, arti dari program. Ketika program dijalankan, efek tiap instruksi didefinisikan oleh semantik dari bahasa.

# DESAIN BAHASA PEMROGRAMAN

- **Architecture.** Bahasa pemrograman dirancang untuk komputer: **well-match atau tidak** dengan arsitektur komputer yang ada.
- **Technical Setting**, memperhatikan sistem operasi, IDE (Integrated Development Environment), network, dan referensi lingkungan lainnya.
- **Standards:** ANSI (American National Standards Institute), atau ISO (International Standards Organization). Contoh: ISO Pascal (1990), ANSI/ISO C++ (2003), dsb.

# DESAIN BAHASA PEMROGRAMAN

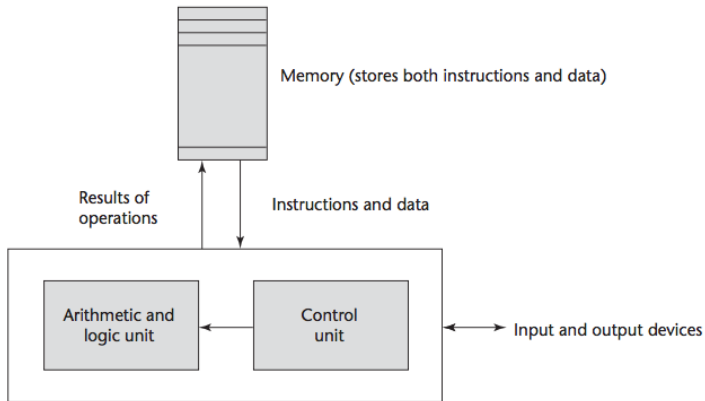
## Goals:

- **Simplicity** and Readability, program harus mudah ditulis, dan mudah dibaca oleh programmer umumnya.
- **Clarity about Binding**, memiliki batasan definisi dan waktu yang jelas, misalnya reserved words, ukuran memori suatu tipe data, run time, dsb.
- **Reliability**, program akan melakukan hal yang sama ketika memperoleh input data yang sama.
- **Support**, mudah diakses, dipelajari, dan di-install oleh siapa saja.
- **Efficient**.



# PENGARUH ARSITEKTUR KOMPUTER

Arsitektur komputer sangat berpengaruh pada perancangan bahasa pemrograman komputer.



# PARADIGMA PEMROGRAMAN

- Paradigma pemrograman adalah bentuk pemecahan masalah mengikuti aliran atau ”genre” tertentu dari program dan bahasa.
- Klasifikasi:

Imperative/ Algorithmic	Declarative		Object-Oriented
	Functional Programming	Logic Programming	
Algol Cobol PL/1 Ada C Modula-3	Lisp Haskell ML Miranda APL	Prolog	Smalltalk Simula C++ Java

# EKSEKUSI KODE PROGRAM

## von Neumann architecture

```
initialize the program counter
repeat forever
    fetch the instruction pointed to by the program counter
    increment the program counter to point at the next instruction
    decode the instruction
    execute the instruction
end repeat
```

## Implementasi bahasa pemrograman:

- Compilation
- Pure Interpretation
- Hybrid Implementation Systems

# FUNCTIONAL PROGRAMMING

- Program serba fungsi, artinya setiap persoalan diselesaikan dengan menggunakan fungsi.
- Mulai dikembangkan tahun 1960an, dimotivasi oleh peneliti bidang artificial intelligence, symbolic computation, theorem proving, rule-based system, dan NLP.
- Bahasa fungsional pertama adalah LISP (McCarthy, 1960).
- Memodelkan masalah komputasi sebagai suatu **fungsi matematika**, yang mempunyai input (**domain**) dan hasil atau output (**range**).

# LOGIC PROGRAMMING (LP)

- **Pemrograman deklaratif**, mendeklarasikan tujuan komputasi, bukan menyusun algoritme secara detail. Disebut juga **rule-based programming**.
- LP dirancang untuk mendeskripsikan properti dari suatu obyek. Hubungan antar obyek dinyatakan dengan aturan **if-then** (jika-maka).
- LP memiliki mekanisme *built-in* untuk menarik kesimpulan (*inference*) berdasarkan deskripsi properti obyek tersebut.
- Contoh Aplikasi:
  - Artificial intelligence, misalnya **MYCIN**
  - Database information retrieval, misalnya **SQL**

# REPRESENTASI

Perhatikan 4 pernyataan berikut:

- Jika hari hujan, partai A tidak dipilih.
- Juminten akan menang hanya jika partai A dan partai B dipilih.
- Juminten atau Mukidi akan menang, tetapi tidak keduanya.
- Jika hari hujan, Mukidi akan menang.

# LOGIKA PROPOSISI

Representasi simbol: R (hari hujan); A (Partai A terilih); J (Juminten menang); B (Partai B terpilih); M (Mukidi menang)

- Jika hujan, partai A tidak dipilih.

$$R \rightarrow \neg A$$

- Juminten akan menang hanya jika partai A dan partai B dipilih.

$$A \wedge B \rightarrow J$$

- Juminten atau Mukidi akan menang, tetapi tidak keduanya.

$$(J \wedge \neg M) \vee (\neg J \wedge M)$$

- Jika hujan, Mukidi akan menang.

$$R \rightarrow M$$

# FIRST ORDER LOGIC : FOL

- Dasar pemrograman logika adalah **First Order Predicate Logic**, sering disingkat **FOPL**, atau **First Order Logic** (FOL).
- Komponen FOL:
  - Objek: kom204, rku20, joko, emptylist, ...
  - Relasi atau Predikat: lelaki(joko), ruang(kom204, rku20), kelompok(apel, buah) ...
  - Fungsi: temanBaik(), awalDari(), ...
  - Hubungan:  $\vee$ ,  $\wedge$ ,  $\neg$ ,  $\Rightarrow$ ,  $\Leftrightarrow$
  - Quantifiers:
    - Universal:  $\forall$
    - Existential:  $\exists$



# QUANTIFIER

- Semua mahasiswa ILKOM adalah cerdas  
 $\forall_x [mhs(x) \wedge dept(x, ILKOM) \Rightarrow cerdas(x)]$
- Tidak semua mahasiswa ILKOM adalah cerdas  
 $\exists_x [mhs(x) \wedge dept(x, ILKOM) \Rightarrow cerdas(x)]$

# LATIHAN KELAS

Tuliskan *quantifier* dari setiap pernyataan berikut:

- Semua jamur ungu itu beracun.
- Tidak ada mahasiswa menyukai semua mata kuliah.
- Everest adalah gunung tertinggi di dunia.
- Sedikitnya ada 2 apel dalam sebuah keranjang.
- Sedikitnya ada 2 apel dalam setiap keranjang.