KOM204 : BAHASA PEMROGRAMAN

Bahasa Pemrograman

Pertemuan 1

Learning Outcomes

- Mahasiswa dapat mengetahui sejarah dan prinsip bahasa pemrograman
- Mahasiswa dapat mengelompokkan berbagai bahasa pemrograman yang ada

Outline Materi

Pustaka acuan:

- Tucker & Noonan, Chapter 1

Outline Materi:

- Prinsip dan paradigma bahasa pemrograman
- Kelompok bahasa pemrograman
- Desain bahasa pemrograman
- Compiler versus Interpreter

Prinsip Bahasa Pemrograman

- Bahasa pemrograman vs bahasa alami
  - ✓ Memfasilitasi komunikasi antar manusia
  - ✓ Bahasa pemrograman juga memfasilitasi komunikasi manusia dengan mesin
  - ✓ Bahasa pemrograman hanya pada domain komputasional
- Perancang bahasa memiliki vocabulary dasar tentang:
  - ✓ Struktur bahasa
  - ✓ Arti

Prinsip Bahasa Pemrograman

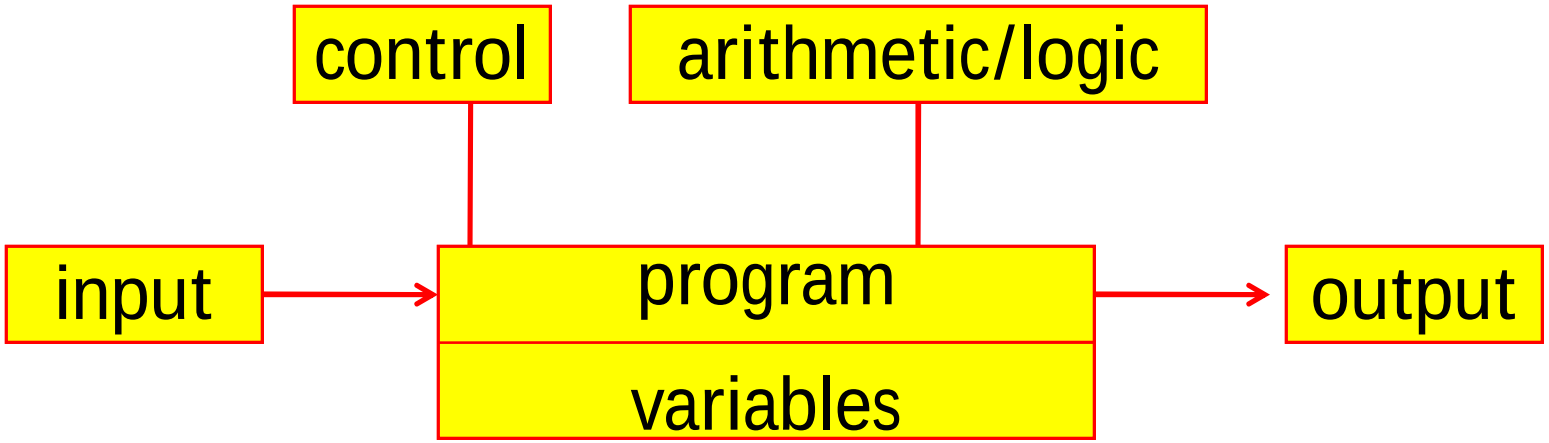
- Prinsip perancangan bahasa : (1) Sintaks, (2) Nama dan Tipe, (3) Semantik.
- Sintaks menjelaskan bagaimana struktur program yang benar. Struktur bahasa pemrograman modern didefinisikan menggunakan bahasa formal yang disebut context-free-grammar.
- Nama dan Tipe menunjukkan bagaimana aturan penamaan entitas (variabel, fungsi, class, parameter, dsb).
- Semantik, arti dari program. Ketika program dijalankan, efek tiap instruksi didefinisikan oleh semantik dari bahasa.

Paradigma

- Paradigma pemrograman adalah bentuk pemecahan masalah mengikuti aliran atau “genre” tertentu dari program dan bahasa.
- Empat paradigma pemrograman pada tiga dekade terakhir:
  - ✓ Imperative programming
  - ✓ Object-oriented programming
  - ✓ Functional programming
  - ✓ Logic programming
- Beberapa bahasa dirancang mendukung lebih dari satu paradigma. Contoh: C++ (imperative dan OOP), Leda (imperative, OOP, functional, logic).

Imperative Programming

- Paradigma paling tua, didasari oleh model komputasi klasik "von Neumann-Eckert".
- Program dan variabel disimpan bersama.
- Program terdiri dari instruksi yang membentuk perhitungan, asignment, input, output, dan kontrol.
- Contoh: Cobol, Fortran, C, Ada, Perl



Object-oriented Programming

- Program adalah kumpulan dari obyek yang saling berinteraksi satu sama lain.
- Program membungkus (encapsulate) data dan fungsi atau prosedur menjadi suatu obyek (class).
- Meliputi mekanisme obyek, pewarisan, dan passing parameter.
- Contoh: Smalltalk, C++, Java, C#

Functional Programming

- Memodelkan masalah komputasi sebagai suatu fungsi matematika, yang mempunyai input (domain) dan hasil atau output (range).
- Tidak menggunakan mekanisme assignment, karena tidak dapat diterima secara matematika, misalnya:

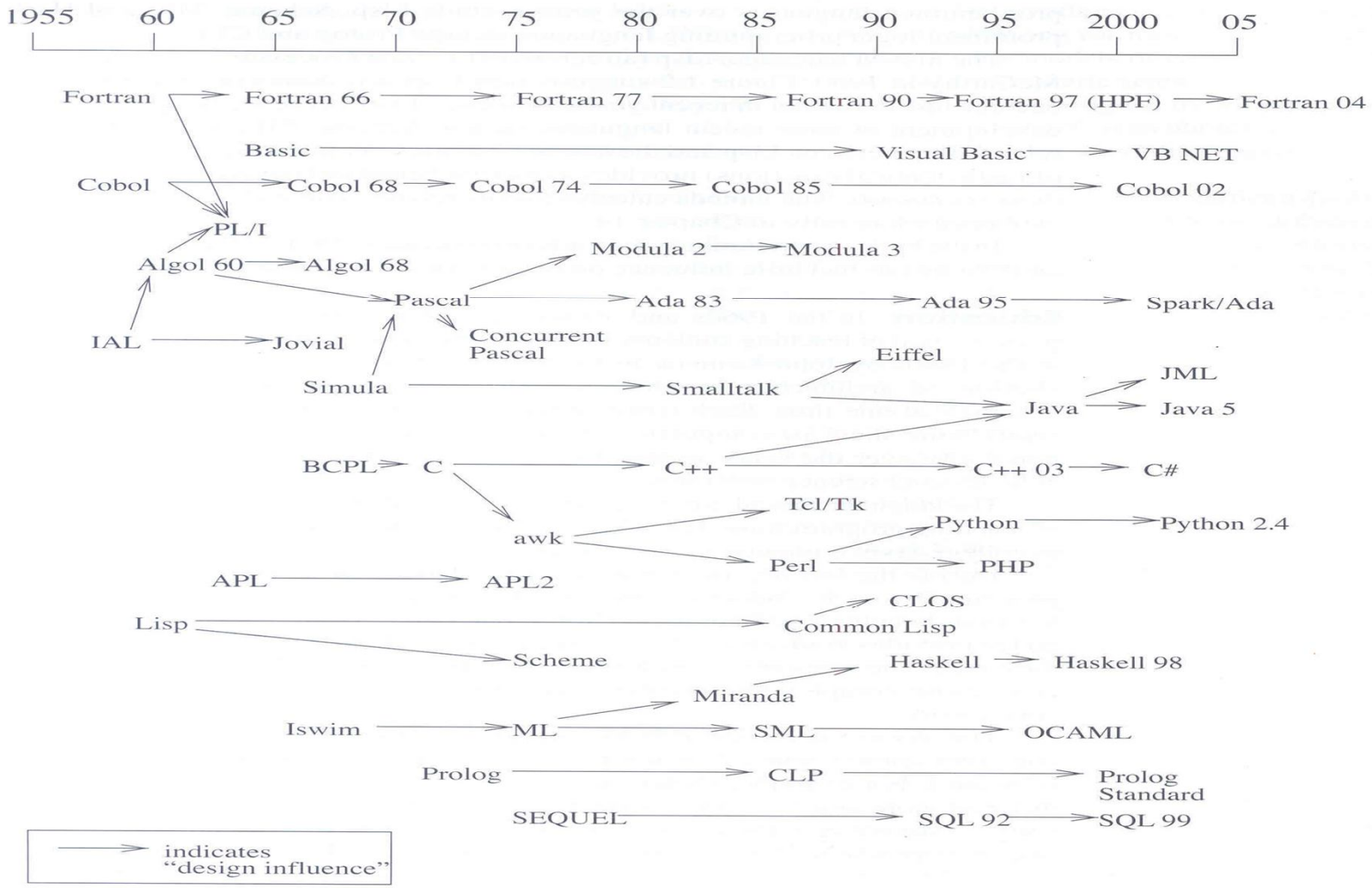
```
x = x + 1
```

- Fungsi mengkombinasikan kondisi dan rekursif.
- Contoh: Lisp (List Programming), Scheme, Haskell.

Logic Programming

- Disebut juga *Declarative Programming*
- Memodelkan masalah menggunakan bahasa deklaratif, yang terdiri dari fakta dan aturan.
- Kadang disebut juga sebagai *rule-based languages*.
- Contoh: Prolog (Programming in Logic).

PL History


1955
60
65
70
75
80
85
90
95
2000
05
Fortran
Fortran 66
Fortran 77
Fortran 90
Fortran 97 (HPF)
Fortran 04
Basic
Visual Basic
VB NET
Cobol
Cobol 68
Cobol 74
Cobol 85
Cobol 02
PL/I
Algol 60
Algol 68
Modula 2
Modula 3
Pascal
Ada 83
Ada 95
Spark/Ada
IAL
Jovial
Concurrent Pascal
Simula
Smalltalk
Eiffel
JML
Java
Java 5
BCPL
C
C++
C++ 03
C#
Tcl/Tk
awk
Python
Python 2.4
Perl
PHP
APL
APL2
CLOS
Lisp
Common Lisp
Scheme
Haskell
Haskell 98
Miranda
Iswim
ML
SML
OCAML
Prolog
CLP
Prolog Standard
SEQUEL
SQL 92
SQL 99
indicates "design influence"

Disain Bahasa Pemrograman

- Menciptakan bahasa sehingga pemrogram dapat memecahkan persoalan yang kompleks.
- Kendala yang harus diperhatikan:
  - ✓ Architecture
  - ✓ Technical Setting
  - ✓ Standards

Disain 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.

Disain Bahasa Pemrograman

Goals:

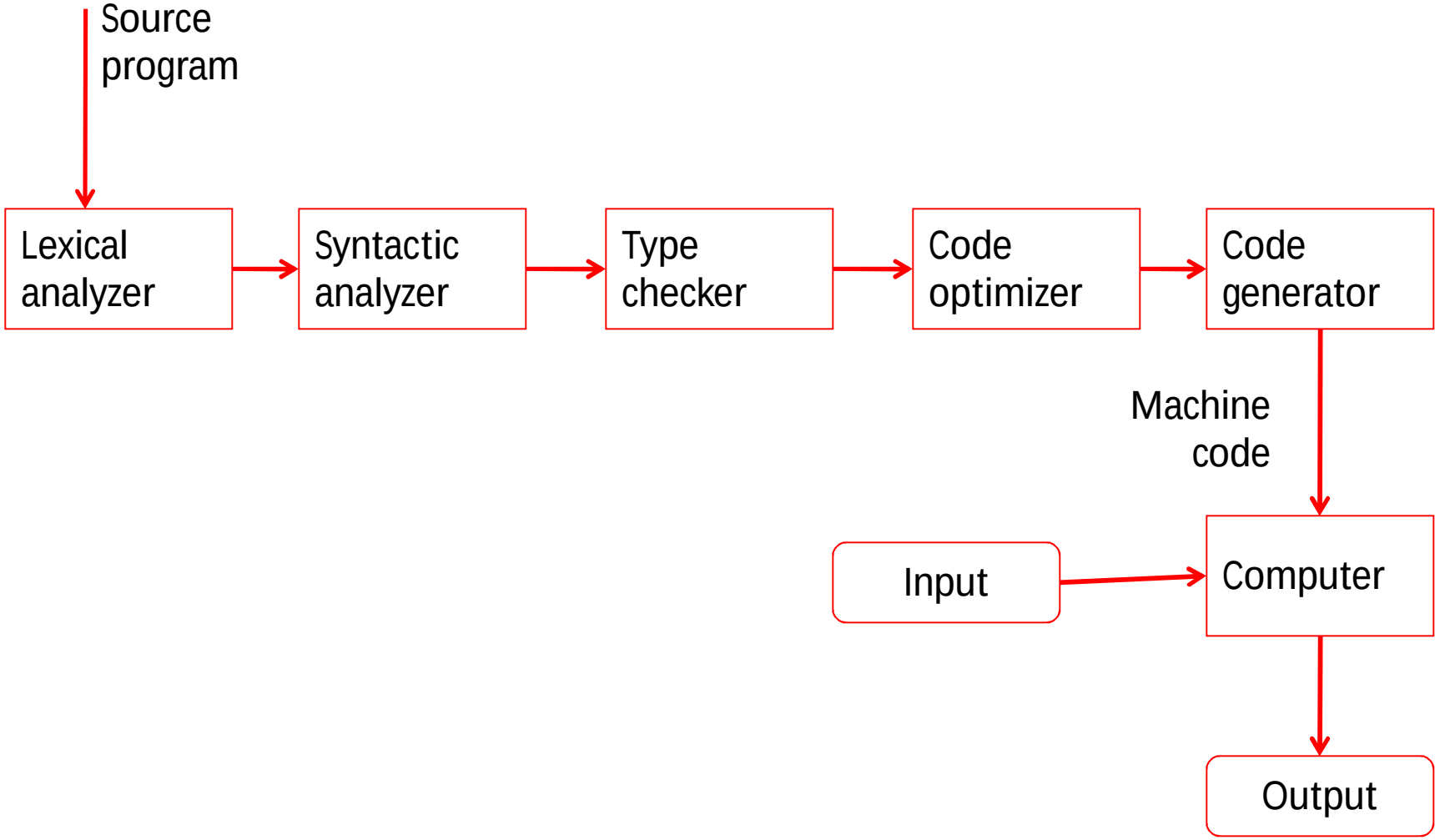
- Simplicity and Readability, program harus mudah ditulis, dan mudah dibaca oleh programmer umumnya.
- Clarity about Binding, memiliki batasan definisi dan waktu yang jelas, misalnya reserved words, ukuran memori suatu tipe data, run time, dsb.
- Reliability, program akan melakukan hal yang sama ketika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

Compilers & Virtual Machines

Bahasa program dianalisis dan selanjutnya diterjemahkan ke dalam bentuk yang dapat dipahami mesin:

- Dijalankan oleh komputer – “real machine” → compiling
- Dijalankan oleh interpreter – software yang mensimulasikan “virtual machine” dan menjalankan dalam “real machine” → interpreting

Compilers


Source program
Lexical analyzer
Syntactic analyzer
Type checker
Code optimizer
Code generator
Machine code
Input
Computer
Output

Virtual Machines & Interpreters