

Bahasa Pemrograman :: Pendahuluan

Julio Adisantoso
ILKOM IPB

20 Pebruari 2011

Deskripsi

Konsep bahasa pemrograman (functional programming, logic programming, concurrent programming, imperative programming, object oriented programming), jenis dan ciri; aturan-aturan dalam struktur pemrograman; elemen-elemen bahasa pemrograman: notasi, deklarasi, scope. Functional programming: ciri-ciri, sintaks, list, dan beberapa aplikasi lainnya. Logic programming : ciri-ciri, relasi, aturan, fakta, goal, prolog search tree, place holder, cut, list, database. Object oriented programming: pengertian obyek, kelas dan data, operator overloading, inheritance, fungsi virtual, polymorphism, stream.

Tujuan dan Strategi

Tujuan

Setelah mengikuti matakuliah ini, mahasiswa diharapkan mampu memecahkan masalah yang berbeda dengan menggunakan bahasa pemrograman yang sesuai.

Strategi

Kuliah diberikan kepada mahasiswa S1 IPB yang mengambil mata kuliah ini sebagai mayor (G6: Ilmu Komputer), baik reguler maupun penyelenggaraan khusus (ekstensi). Perkuliahan dilakukan sebanyak 14 kali pertemuan kuliah tatap muka dan 12 kali praktikum. Metode perkuliahan adalah kombinasi antara ceramah, diskusi, dan latihan. Sedangkan praktikum dilaksanakan dengan metode tutorial dan latihan. Mahasiswa wajib mengikuti perkuliahan minimal 80 persen, dan praktikum 100 persen. Mahasiswa pengulang matakuliah Bahasa Pemrograman diwajibkan mengikuti keseluruhan kegiatan kuliah dan praktikum selama satu semester.

Tugas dan Bacaan

Tugas

Tugas pemrograman diberikan pada saat kuliah dan wajib dikerjakan oleh mahasiswa, dan hasilnya dikumpulkan sesuai dengan waktu yang telah ditetapkan. Keterlambatan pengumpulan tugas akan mengurangi nilai sebesar 10 satuan per hari dengan batas maksimum keterlambatan adalah 5 hari. Mahasiswa yang melakukan kecurangan dan/atau plagiarism akan mendapatkan sanksi tegas.

Bacaan

- 1 Programming Languages: Principles and Paradigms. 2nd Ed. Allen B. Tucker and Robert E. Noonan.
- 2 C: How to Program. 3rd Ed. Deitel, H. M. and P. J. Deitel. Pearson Education Prentice Hall. Prentice-Hall, New Jersey.
- 3 Java: How to Program. 3rd Ed. Deitel, H. M. and P. J. Deitel. Pearson Education Prentice Hall. Prentice-Hall, New Jersey

Tata Tertib

- Sesuai dengan ketentuan yang terdapat pada Buku Panduan Sarjana IPB
- Hadir paling lambat 15 menit. Mahasiswa **TIDAK DIPERKENANKAN** masuk kelas setelah 15 menit kuliah dimulai.
- Berpenampilan dan berbusana sopan serta rapi.
- Tidak menggunakan sandal atau sejenisnya, serta tidak mengoperasikan handphone, laptop, atau sejenisnya.
- Tidak ada ujian dan penugasan susulan atau perbaikan
- **TIDAK DIPERKENANKAN PINDAH KELAS PARALEL**

Kriteria Penilaian

- Nilai akhir (NA) adalah nilai kumulatif dari nilai ujian tengah semester (UTS), ujian akhir semester (UAS), dan nilai LL (tugas atau praktikum atau kehadiran). Metode dan bobot nilai sebagai berikut:
 - UTS+UAS tertulis, bobot total 70%
 - UTS+UAS praktek, bobot total 20%
 - LL dinilai melalui aktivitas mahasiswa (praktikum, kehadiran), dan nilai tugas, dengan bobot total 10%
- Selang nilai untuk menetapkan huruf mutu A, B, C, D, atau E ditentukan berdasarkan nilai rata-rata dan standard deviasi dengan menggunakan sebaran normal. Ketentuan ini berlaku sama untuk semua mahasiswa baru atau pengulang.

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

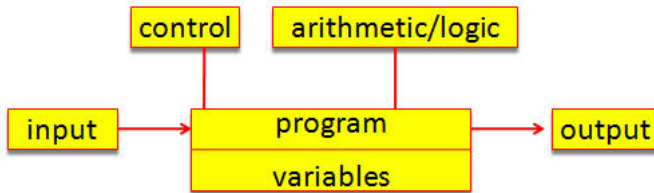
- Prinsip perancangan bahasa : (1) Sintaks, (2) Nama dan Tipe, (3) Semantik.
- Sintaks menjelaskan bagaimana struktur program yang benar. Struktur bahasa pemrograman modern didefinisikan menggunakan bahasa formal yang disebut context-free-grammar.
- Nama dan Tipe menunjukkan bagaimana aturan penamaan 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, assignment, 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#

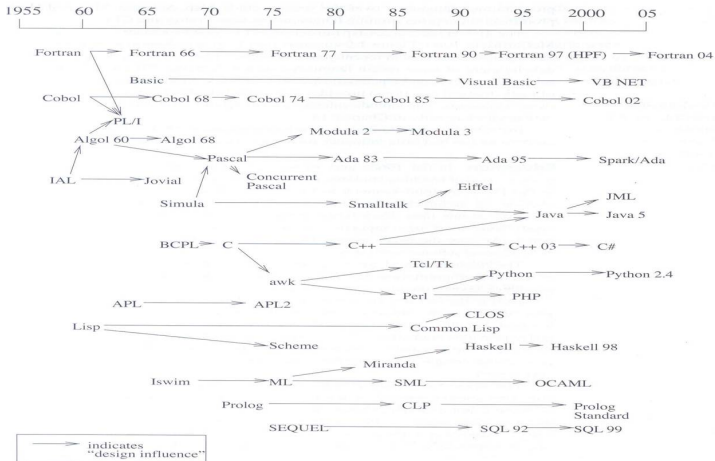
Functional Programming

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



Desain Bahasa

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

Kendala Desain

- **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.

Tujuan

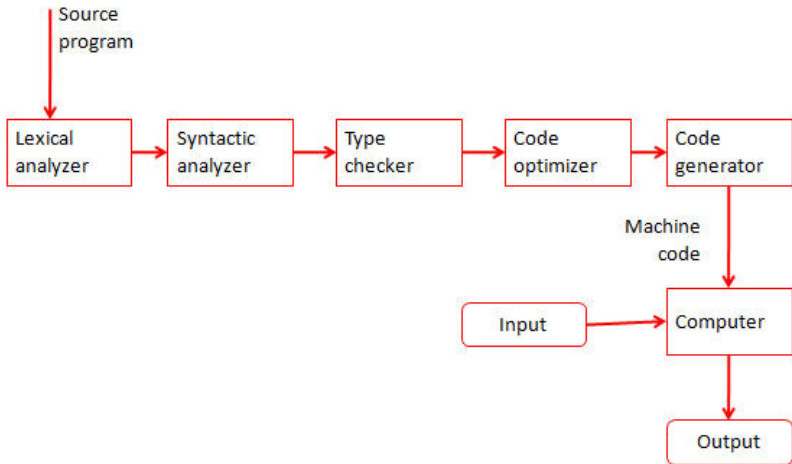
- **Simplicity and Readability**, program harus mudah ditulis, dan mudah dibaca oleh programmer umumnya.
- **Clarity about Binding**, memiliki batasan definisi dan waktu yang jelas, misalnya reserved words, ukuran memori suatu tipe data, run time, dsb.
- **Reliability**, program akan melakukan hal yang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

Compilers and Virtual Machines

Bahasa program dianalisis dan selanjutnya diterjemahkan ke dalam bentuk yang dapat dipahami mesin, salah satu dari:

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

Compiler



Virtual Machines and Interpreters

