

Sesi Kuliah Pertama

IF2281 Pemrograman Berorientasi Objek

Disusun Oleh
Inggriani Liem

Tujuan Sesi Ini

- Mahasiswa memahami alur perkuliahan “programming” secara “menyeluruh”, dan relasi antar kuliah yang pernah ditempuhnya :
 - Mahasiswa memahami relasi antara kuliah-kuliah pemrograman yang pernah ditempuhnya : IF1282, IF2182, IF2281
 - Mahasiswa memahami posisi kuliah IF2281 sebagai “capstone course” semua kuliah programming yang diperoleh sejak TPB
- Mahasiswa memahami tuntutan waktu dan komitmen serta cara belajar yang diharapkan dosen, sehingga proses pembelajaran hasilnya optimal

Kuliah “Pemrograman”

Untuk Mahasiswa Informatika

- Wajib:
 - IF1281 : Dasar Pemrograman
 - IF2182 : Algoritma dan Struktur Data
 - IF2281 : Pemrograman Berorientasi Objek
- Pilihan :
 - Ifxxx – Prmograman Internet
 - Ifxxx – Pemrograman Sistem
- Mengandung pemrograman : Basis Data, Logika Pemrograman
- Mengandung tugas membuat program :
 - hampir semua matakuliah

Prinsip pengajaran programming IF

- Lebih menekankan pada konsep ketimbang tools
- Mencakup aspek statis (design, implementasi kode) dan dinamis (run time behaviour, eksekusi).
- Kemampuan mahasiswa : konsep & practice
- Tools, platform, Bahasa dipilih sebagai alat eksekusi dan praktek dengan tujuan :
 - menolong “realisasi”, “praktek” mengenai pemahaman mahasiswa
 - Melatih mahasiswa agar kalau bekerja sudah tidak kikuk
 - Membentuk programmer yang independent thd bahasa
- Dengan makin kompleksnya tools & bahasa, maka “subset” pemilihan tools dan bahasa yang menjadi bagian kuliah menjadi penting. Intinya : bukan mengajarkan “Java” (sesuai training Sun sJAVA)

Sasaran “programmer”

- Programmer yang bukan hanya end user/GUI tapi “core developer”
- Programming yang mempunyai kebiasaan dalam Programming practice yang:
 - Berperilaku disiplin dan menerapkan “disiplin” pemrograman 😊😊😊
 - Programmer yang bekerja baik dalam tim (ada coding standard, SOP, versioning). Catatan : Bisa saja mengerjakan program outsource/open source secara freelance, yang sebetulnya kerja dalam team
 - Mampu memprogram secara terukur (memahami metrics dan mempunyai kemampuan time management yang baik)
- Semua dicapai secara berjenjang: kecil dan terbimbing, skala menengah, besar dengan komponen

IF1282 – Dasar Pemrograman

Materi kuliah

- Pemrograman Fungsional
 - Mahasiswa memahami semua konsep pemrograman secara “logik”, yaitu basis untuk membangun ADT dan membangun loop yang baik (berdasarkan definisi rekurens)
 - Praktek agar trampil memrogram dalam salah satu bahasa fungsional. LISP hanya alat praktek, supaya mahasiswa dapat memanipulasi list, pohon dsb tanpa peduli implementasi detail yang ruwet
- Pemrograman Prosedural:
 - pengenalan salah satu bahasa pemrograman (boleh Pascal atau C)

IF1282 – Dasar Pemrograman Practices

- Membentuk sikap dasar memprogram yang baik :
 - Bekerja di lab sebagai “kantoor”, berkomunikasi dengan “atasan langsung” (asisten) dan tidak langsung (dosen)
 - Tugas kecil dan jelas, rutin per minggu dan dinilai
 - Mengerjakan tugas-tugas yang spesifikasinya jelas

IF2182 Struktur Data

Materi

- Prinsip keseluruhan : OOP dengan bahasa non OOL
- Bagian Ia Algoritma :
 - Konsep yang ada dan sama di fungsional, di”translate”: type,
 - Konsep yang beda dijelaskan perbedaannya, misalnya : aksi versus fungsi, aplikasi versus call, kondisional,
 - Konsep yang tidak ada di pemrograman fungsional menjadi fokus: memori, variabel, konstanta, loop, mesin
- Bagian Ib Implementasi internal :
 - Pointer dan manipulasinya (bagian dari linked list)
 - Rekurens di konteks prosedural (variable lokal, dst)
- Bagian II : Struktur Data
 - Semua konsep sudah diajarkan di Fungsional, bahas implementasi memori dan pengkodean ke bahasa prosedural
 - Studi Kasus

IF2182 Struktur Data

Practices

- Skala tugas : kecil dan menengah
- Tugas :
 - Tugas diberikan setiap minggu namun tidak diperiksa/setor setiap minggu, untuk membuat
 - Ada waktu satu minggu untuk integrasi tugas kecil yang pernah dikerjakan
- Mahasiswa pernah bekerja dengan:
 - Kompilasi terpisah, pemecahan kode program yang sistematis
 - Versioning tools
- Minimal dua bahasa dan dua proyek “besar” serta memberikan latihan program maintenance (memodifikasi program dalam waktu singkat)

IF2281 OOP

Materi

- Prinsip :: OO tingkat lanjut, bukan hanya “enkapsulasi” yang dipelajari di IF 2182 Struktur Data
- Materi yang dicakup :
 - Dari pemrograman prosedural ke OOP
 - Konsep dasar OOP secara menyeluruh (“big picture”)
 - Enkapsulasi
 - Generik
 - Inheritance (single, multiple, repeated)
 - Polymorphism dan permasalahannya
 - Konkurensi sederhana [hanya Java]
 - GUI based programming, MVC [Java Swing]
 - Design kelas
 - Perbandingan bahasa-bahasa OO
- Mahasiswa harus praktek minimal dalam **dua bahasa OO** yang banyak di pasaran : contoh C++, Java
- Bahasa lain harus diperkenalkan, tanpa run, misalnya Eiffel, C#

IF2281 OOP

Practices

- Code reuse : dari hasil praktek Struktur data ; oleh sebab itu akan optimal kalau bahasanya lain, namun kompatibel. Misalnya C (SD), menjadi Java/C++ (OOP)
- Tugas :
 - Tugas kecil membuat class/komponen
 - Integrasi memakai clas/komponen yang sudah dibuat
- Minimal dua proyek, dengan satu proyek berdasarkan proposal yang diajukan mahasiswa untuk membangkitkan minatnya. Proposal harus memenuhi rambu-rambu “metriks” yang diberikan sehingga berkeadilan serta memberikan “sense” terhadap software metrics (banyaknya kelas, kompleksitas,..)

Penutup

- Kuliah OOP yang diajarkan dan dipraktekkan di IF tidak terlepas dari dua kuliah sebelumnya. Relasinya sangat erat, bahkan “berkesinambungan”
- Tugas yang pernah dikerjakan akan dipakai lagi di kuliah berikutnya (aspek code reuse)
- Skala program yang dikerjakan makin menaik (volume/LoC, kompleksitas, struktur)
- Rangkaian kuliah lebih bersifat pemrograman “tekstual”, mengajak mahasiswa mengendalikan alur eksekusi kode, dibandingkan dengan memprogram “visual” yang dikebanyakan kasus menitipkan perilaku eksekusi kepada platform dan tools
- Ditargetkan untuk membentuk “core developer”, bahkan untuk software system (bukan end user application)

Kenapa C++ dan Java ?

- Pada kuliah ini akan diajarkan C++ dan Java, dua buah bahasa yang cukup banyak dipakai dalam dunia industri
- Urutan pengajaran : C++ (untuk yang belum mengenal, “rumit) kemudian Java (relatif lebih mudah):
- STL (Standard templates Library) yang tersedia:
 - Diajarkan “bagaimana menggunakannya”, bukan “menggunakannya.
 - Membandingkan dengan sejenis yang pernah dibuatnya secara “naif”
- API Java hanya terbatas yang diajarkan, sisanya diharap akan dipakai di perkuliahan lain yang memang fokus, misalnya:
 - J2ME diajarkan di Mobile computing
 - JDBC disinggung di kuliah basis data
 - Parser dipakai di kuliah kompilasi, ...

Highlight

C++ versus Java

- Pointer
- Memory management serta run time behaviour
- Ctor, cctor, dtor, iostream, operator=
- Pemakaian keyword sama dengan arti beda-beda (seperti const)
- STL dikenalkan bagaimana cara pakai
- Beda konsep dengan reference Java
- Tidak perlu mengelola dtor, cctor, operator=
- Dipakai untuk introduksi pemrograman konkuren (lebih primitid)
- API dipilih sesuai tema tugas besar
- Mencakup konkurensi dan SWING