



Rencana Kuliah IF2210

Pemrograman Berorientasi Objek

Semester 2-2014/2015

Riza Satria Perdana
Inggriani



Course Overview

- Memberikan pemahaman konsep, teknik dan tools untuk membentuk keahlian membangun perangkat lunak dengan paradigma berorientasi objek
- Mahasiswa diharapkan tidak bingung dengan masalah sintaks. Oleh karena itu ditargetkan minimal mahasiswa harus menguasai 3 atau 4 bahasa pemrograman berorientasi objek, dua di antaranya harus “fasih”.



Scope

- Teori, paradigma dan Konsep OOP dan lokasi OOP dalam tatanan *SW Life cycle*
- Praktek dalam minimal 2 bahasa [ditambah satu bahasa sebagai *overview*]
 - Konkuren dalam JAVA saja
 - Aplikasi berbasis UI – JAVA saja
- *Case Study* – untuk mencakup desain
- Tugas kecil dan tugas besar



Course Outcomes

- Setelah mengikuti kuliah ini, mahasiswa diharapkan mampu:
 - Memahami konsep OOP {akan didefinisikan 7 level OOP, mulai class/object s.d. Concurrent & distributed object}, reuse, library
 - Merancang dan mengimplementasi kelas
 - Memakai kelas-kelas yang disediakan
 - Merancang dan mengimplementasi aplikasi/SW secara OO
 - Memahami implementasi OOP dalam minimal 3 bahasa dan satu bahasa notasi: C++, JAVA, C#, Eiffel



CourseTopics [Part-1]

- OO paradigm – imperative paradigm
- Class, Object, attribute, method
- Overview of passing parameter
- Reference and container
- Object creation & destruction: ctor, cctor, dtor,
- Object manipulation: assignment , comparaison
- Generic class – template
- Abstract class, root class, ...
- Exception and assertion



Course Topics [Part 2]

- Inheritance, multiple inheritance, repeated inheritance
- Polymorphism
- STL, API – data structure, iterator, algorithm
- Concurrent object, thread
- Language special feature

Concept will be implemented in existing feature of OOL



Komponen Penilaian

- Komponen penilaian dengan persentase
 - Kuis 5-10%
 - UTS 20-35% [teori, praktek]
 - UAS 30-40% [teori, praktek]
 - Tugas kecil 10-15%
 - Tugas Besar 15-20%
 - Kehadiran: menjadi pertimbangan
- Usahakan terbaik...



Penentuan nilai akhir

- Mahasiswa harus hadir UTS dan UAS sesuai jadwal. Jika tidak hadir UTS/UAS, nilainya NOL. Kesempatan ujian ulangan jika tidak hadir hanya akan diberikan dengan alasan khusus yang akan ditentukan setelah penentuan kelulusan.
- Tugas pemrograman dengan autograder dibobot sesuai pengumuman praktikum/kuis/ujian, dan hanya dibuka pada jam yang ditentukan.
- Tidak ada kesempatan mengulang kuis/demo tugas. Jika tak hadir kuis/demo, maka nilainya Nol. **Kuis dapat dilakukan sewaktu-waktu tanpa pemberitahuan pada jam kuliah yang terjadwal atau jam praktikum**
- Semua tugas disetor melalui LMS Oddyseus, yang dibuka pada waktu yang ditentukan. Tugas yang terlambat disetor (tak mungkin), nilainya NOL.
- Jika diketahui mencontek, atau melakukan ketidak-jujuran, maka nilainya NOL.



Nilai Akhir

- Nilai diberikan dengan skala absolut, bukan rata-rata kelas dan ditentukan dari Final Score sebagai berikut:

Final Score	Nilai Akhir
85 s.d 100 dan (*)	A
75 sampai 85	AB
70 sampai 75	B
65 sampai 70	BC
55 sampai 65	C
40 sampai 55	D
< 40	E

(*) mahasiswa kehadirannya >90% mulai minggu 1, dan tidak ada komponen nilai bernilai Nol, dan tidak pernah menimbulkan persoalan. Jika ada suatu cacat diturunkan ke AB atau B.



Communication

- Lokasi akses course material dan media komunikasi
 - Oddyseus: <http://oddyseus.if.itb.ac.id>
- Komunikasi dengan dosen
 - Lewat email atau forum Oddy
 - Komunikasi tentang kuliah melalui forum atau ketua kelas. Hanya jika sangat personal, dapat menghubungi secara personal
- Komunikasi peserta
 - Lewat milis kuliah if2210@students.if.itb.ac.id



Kehadiran di Kelas dan Lab....

- Membawa:
 - kertas A-4 (siap untuk kuis),
 - Pensil dan penghapus (disarankan membawa pensil yang tak mengotori ruangan dengan serutan)
 - Ballpoint merah dan {biru atau hitam}
 - Laptop terinstal kompiler C++ dan JAVA, siap untuk latihan dan hanya dikeluarkan/diaktifkan jika mendapat izin
- Berbaju rapi, bersih dan bersepatu
- Tidak terlambat dari jam yang ditentukan oleh dosen [toleransi dari jam yang ditentukan jadwal ITB]
- Membawa KTM dan KSM (siap diperiksa, terutama jika kuis/ujian) harus dapat menunjukkan identitas akademik
- Khusus di lab, harus taat peraturan lab
- Last but not least (ini bukan peraturan): hadir badan dan jiwa 😊 😊 😊
Hadir bukan karena harus mengisi “Daftar Hadir”.
- Meninggalkan kelas/lab dalam keadaan rapi dan bersih



Language specific topics

Java

- Interface
- Java API
- Thread
- Junit
- Javadoc

C++

- Pointer and reference
- Friend class
- Operator=, operator+, other operator overloading
- STL
- Doxygen



Referensi : Bahasa C++

- C++
 - Stroustrup: “The C++ Programming”, 2nd ed.
 - Lippman: C++ Primer, 2nd ed
 - Nicolai M. Josuttis: “The C++ Standard Library: a tutorial and Reference”, 2nd ed



Referensi – OOAD, OO Concept & Eiffel

- *Object-Oriented Software Construction*, Second Edition, by [Bertrand Meyer](#), Prentice Hall, 1997, [ISBN 0-13-629155-4](#)
- [http://en.wikipedia.org/wiki/Eiffel %28programming language%29](http://en.wikipedia.org/wiki/Eiffel_%28programming_language%29)
- Grady Booch, “Object oriented Analysis and Design with Application”, The Benjamin/Cumming Publishing Company, 1995



Referensi - JAVA

- Memakai Bahasa Java
 - <http://docs.oracle.com/javase/tutorial/>
- Mengenal Bahasa Java
 - <http://docs.oracle.com/javase/specs/>
- Textbook :
 - James Gosling et all
- Program-program kecil bahasa Java



Design, Bahasa OO dan Autograder

- Aspek desain akan ditambah dengan dua case study, di mana aspek desain dan “*software reuse*” ditekankan.
- Bahasa akan diajarkan satu per satu, mulai C++ dulu. Bahasa JAVA diajarkan lebih cepat . Beberapa konsep yang sama/berbeda “diajarkan” agar lebih dimengerti.
- Aspek spesifik bahasa akan dicakup sedapat mungkin
- Pemakaian autograder sistem Oddyseus untuk tugas kecil, dan kompilasi class pembangun aplikasi. Autograder adalah sebuah program yang akan menilai hasil eksekusi program berdasarkan *test case*. Hasil eksekusi program anda akan dibandingkan dengan hasil eksekusi program solusi.
- Setiap minggu anda harus mencek Oddyseus.



Ceklist Kompetensi

- Lihat spreadsheet detail yang diberikan dan akan diupload



Penutup

- Belajar memrogram seperti naik sepeda, anda harus banyak berlatih
- Belajar bahasa pemrograman tidak sesulit belajar bahasa natural. Hanya diperlukan disiplin untuk mengingat aturan
- Memrogram skala besar, perlu IDE.
Memrogram skala kecil, anda harus mampu melakukannya dengan *pencil and paper*.
- Kamis, 22 Januari 2015 praktikum dimulai

Tanya Jawab – Q&A

