Pendahuluan Perkuliahan Metode Formal

Kuliah Metode Formal Semester Ganjil 2015-2016

M. Arzaki

Fakultas Informatika
Telkom University

FIF Tel-U

Agustus 2015

Bahasan

1. Motivasi

Bahasan

1. Motivasi
2. Deskripsi Perkuliahan dan Pengajar

Bahasan

1. Motivasi
2. Deskripsi Perkuliahan dan Pengajar
3. Apa, Kapan, dan Dimana

Bahasan

1. Motivasi
2. Deskripsi Perkuliahan dan Pengajar
3. Apa, Kapan, dan Dimana
4. Aturan Penilaian, Presensi, dan Evaluasi

Bahasan

1. Motivasi
2. Deskripsi Perkuliahan dan Pengajar
3. Apa, Kapan, dan Dimana
4. Aturan Penilaian, Presensi, dan Evaluasi
5. Referensi Materi Kuliah dan Topik yang Dibahas

Bahasan

1. Motivasi
2. Deskripsi Perkuliahan dan Pengajar
3. Apa, Kapan, dan Dimana
4. Aturan Penilaian, Presensi, dan Evaluasi
5. Referensi Materi Kuliah dan Topik yang Dibahas
6. Lain-lain

Bahasan

Apa itu Metode Formal?

Metode Formal

Diambil dari website laboratorium FMSE UI:

Metode formal (*formal methods*) merupakan sebuah teknik berbasis logika matematika untuk membuat spesifikasi sebuah sistem komputer (*software* maupun *hardware*) secara tidak ambigu, rancu, dan dapat diverifikasi.

Pemakaian metode formal dimotivasi oleh penerapan analisis logika dan matematika yang mampu menjamin kebenaran dari sebuah desain. Kebenaran implementasi desain tersebut dijamin dengan kebenaran bukti matematis dari satu atau beberapa formula.

Mengapa ada kuliah Metode Formal?

Perkuliahan Metode Formal (CIG4F3) di program sarjana teknik informatika merupakan suatu kuliah pengantar yang ditujukan untuk mahasiswa tingkat tiga dan empat yang akan memilih topik penelitan untuk tugas akhirnya. Mahasiswa diharapkan telah menempuh perkulialahan logika matematika dengan baik.

Setelah menempuh perkuliahan ini, mahasiswa diharapkan memiliki pengetahuan dan keterampilan dasar dalam memakai kerangka formal pada rekayasa perangkat lunak.

Masalah Spesifikasi Sistem

Masalah Spesifikasi Sistem

Seorang *software engineer* diminta oleh manajernya untuk membuat suatu sistem informasi dengan spesifikasi berikut:

Masalah Spesifikasi Sistem

Masalah Spesifikasi Sistem

Seorang *software engineer* diminta oleh manajernya untuk membuat suatu sistem informasi dengan spesifikasi berikut:

1. Ketika *system software* di-*upgrade*, *user* tidak dapat mengakses *file system*;
2. Jika *user* dapat mengakses *file system*, maka *user* dapat menyimpan *file* baru;
3. Jika *user* tidak dapat menyimpan *file* baru, maka *system software* tidak sedang di-*upgrade*.

Apakah sistem informasi dengan spesifikasi di atas dapat dibuat?

Tragedi Ariane 5

Gambar diambil dari https://www.ima.umn.edu/~arnold/disasters/ariane.html.

Diambil dari https://www.ima.umn.edu/~arnold/disasters/ariane.html.

> *On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in a failure. Only about 40 seconds after initiation of the flight sequence, at an altitude of about 3700 m, the launcher veered off its flight path, broke up and exploded.*

The failure of the Ariane 501 was caused by the complete loss of guidance and altitude information 37 seconds after start of the main engine ignition sequence (30 seconds after lift-off). This loss of information was due to specification and design errors in the software of the inertial reference system.

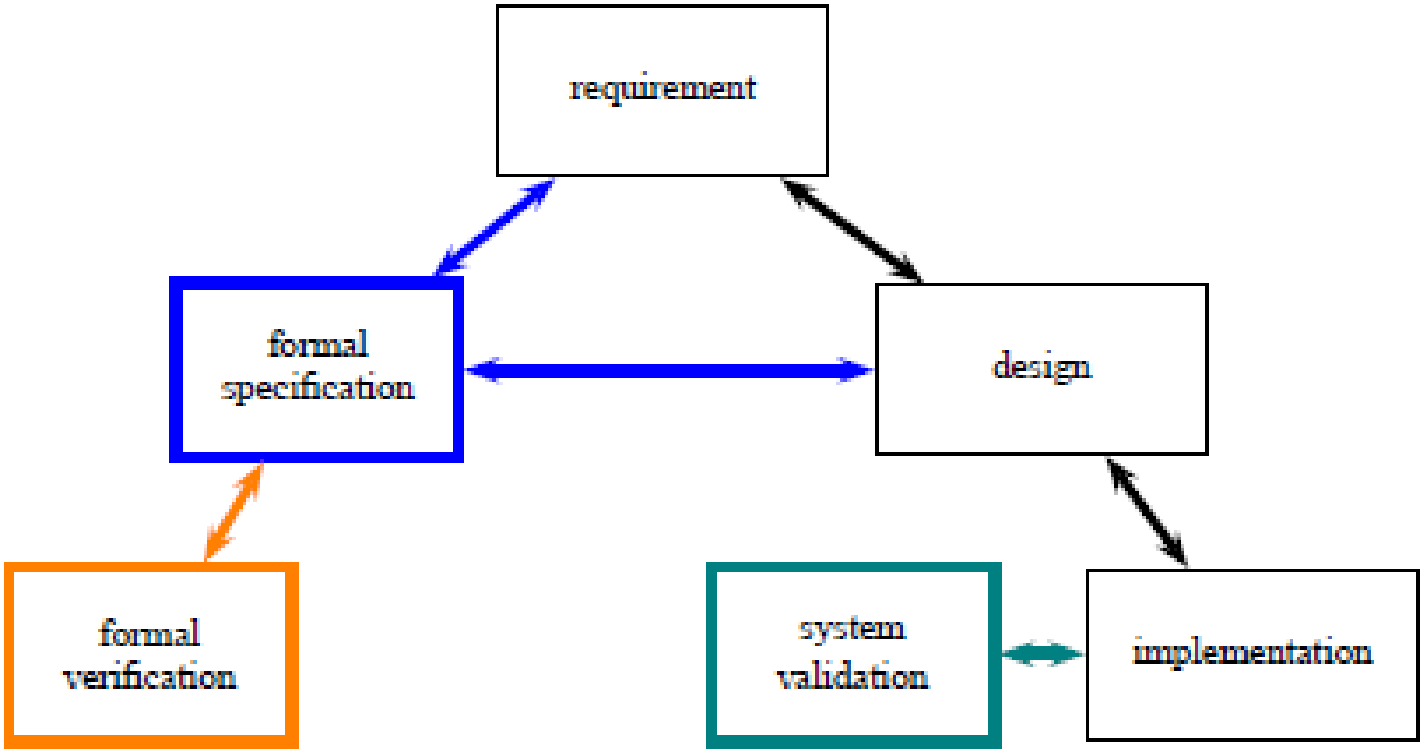
The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a* **value greater than** *what could be represented by a 16-bit signed integer.*

SRI: *Système de Référence Inertielle or Inertial Reference System*.

Apa yang Dilakukan dengan Metode Formal

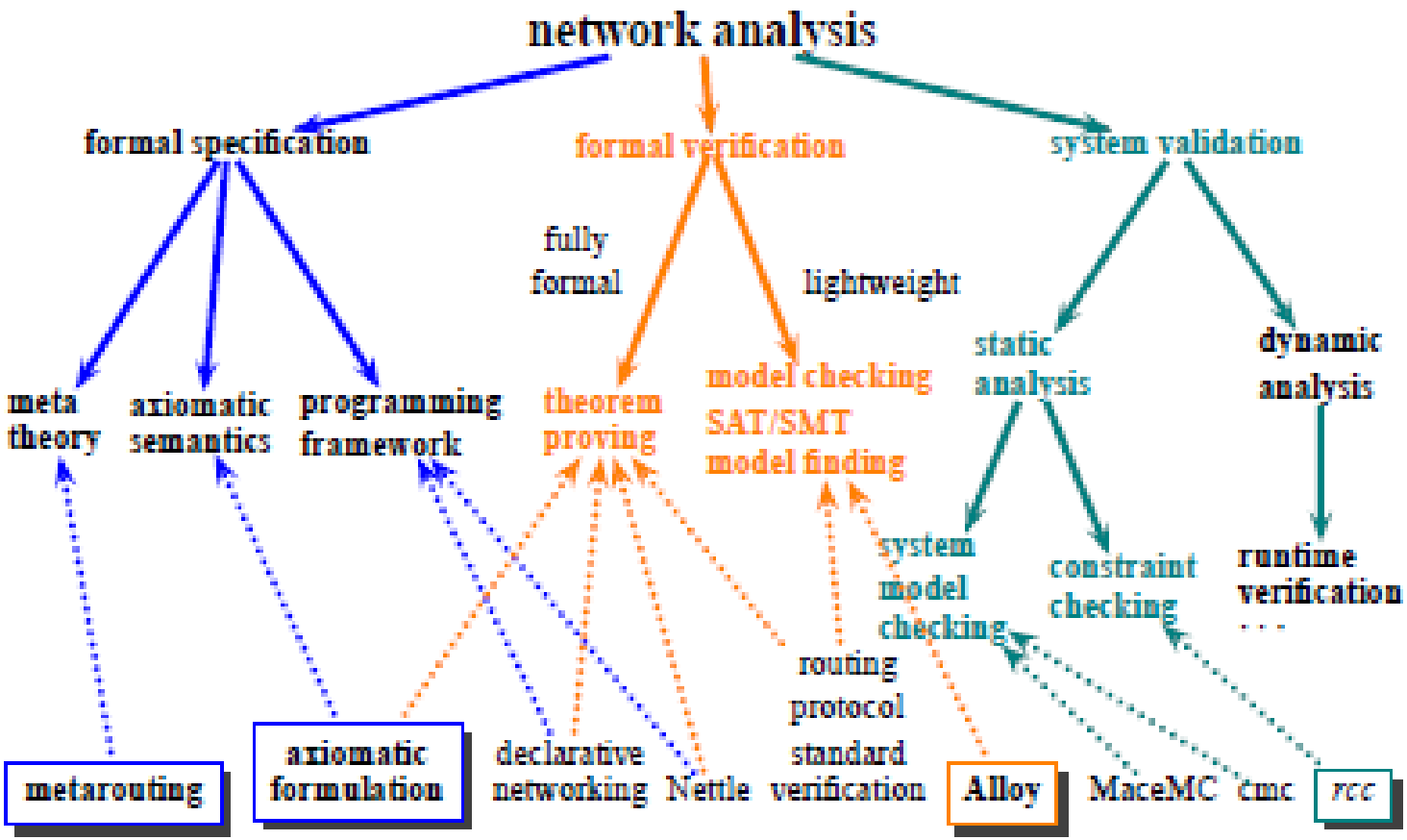
1. Pemeriksaan *software correctness* dan *reliability*-nya.
2. Verfikasi program.
3. *Program refinement*.
4. *Theorem proving* untuk spesifikasi sistem.
5. *Model checking* untuk spesifikasi sistem.
6. Formalisasi protokol keamanan.

Kerangka Kerja (*Framework*) Metode Formal



Gambar diambil dari *Formal Analysis of Network Protocol* (Anduo Wang, 2010).

Metode Formal dalam Analisis Jaringan (*Network Analysis*)



Gambar diambil dari *Formal Analysis of Network Protocol* (Anduo Wang, 2010).

Teknik Verifikasi Formal

Teknik verifikasi formal untuk suatu sistem terdiri atas tiga bagian utama:

1. Kerangka untuk memodelkan sistem (*framework for modelling systems*) untuk memberikan deskripsi sistem yang bersangkutan.
2. Bahasa spesifikasi (*specification language*) untuk mendeskripsikan sifat-sifat atau persyaratan yang harus dipenuhi.
3. Metode verfikasi (*verification method*) untuk menentukan apakah deskripsi sistem menenuhi spesifikasi yang telah ditetapkan.

Teknik Verifikasi Formal

Teknik verifikasi formal untuk suatu sistem terdiri atas tiga bagian utama:

1. Kerangka untuk memodelkan sistem (*framework for modelling systems*) untuk memberikan deskripsi sistem yang bersangkutan.
2. Bahasa spesifikasi (*specification language*) untuk mendeskripsikan sifat-sifat atau persyaratan yang harus dipenuhi.
3. Metode verfikasi (*verification method*) untuk menentukan apakah deskripsi sistem menenuhi spesifikasi yang telah ditetapkan.

Pendekatan verifikasi dapat dilakukan dengan cara *proof-based* atau *model-based*.

Teknik Verifikasi Formal

Teknik verifikasi formal untuk suatu sistem terdiri atas tiga bagian utama:

1. Kerangka untuk memodelkan sistem (*framework for modelling systems*) untuk memberikan deskripsi sistem yang bersangkutan.
2. Bahasa spesifikasi (*specification language*) untuk mendeskripsikan sifat-sifat atau persyaratan yang harus dipenuhi.
3. Metode verfikasi (*verification method*) untuk menentukan apakah deskripsi sistem menenuhi spesifikasi yang telah ditetapkan.

Pendekatan verifikasi dapat dilakukan dengan cara *proof-based* atau *model-based*.

1. *Proof-based* (berbasis pembuktian matematis). Sistem dideskripsikan dalam suatu himpunan Γ yang terdiri atas formula-formula logika. Jika spesifikasi sistem dinyatakan dalam formula ψ, maka verifikasi berbasis pembuktian matematis merupakan verifikasi untuk memeriksa apakah dari formula-formula pada Γ dapat disimpulkan kesimpulan ψ.

Teknik Verifikasi Formal

Teknik verifikasi formal untuk suatu sistem terdiri atas tiga bagian utama:

1. Kerangka untuk memodelkan sistem (*framework for modelling systems*) untuk memberikan deskripsi sistem yang bersangkutan.
2. Bahasa spesifikasi (*specification language*) untuk mendeskripsikan sifat-sifat atau persyaratan yang harus dipenuhi.
3. Metode verfikasi (*verification method*) untuk menentukan apakah deskripsi sistem menenuhi spesifikasi yang telah ditetapkan.

Pendekatan verifikasi dapat dilakukan dengan cara *proof-based* atau *model-based*.

1. *Proof-based* (berbasis pembuktian matematis). Sistem dideskripsikan dalam suatu himpunan Γ yang terdiri atas formula-formula logika. Jika spesifikasi sistem dinyatakan dalam formula ψ, maka verifikasi berbasis pembuktian matematis merupakan verifikasi untuk memeriksa apakah dari formula-formula pada Γ dapat disimpulkan kesimpulan ψ.
2. *Model-based* (berbasis pemeriksaan model). Sistem dideskripsikan dengan suatu model $\mathcal{M}$ yang memiliki berhingga *state*. Jika spesifikasi sistem dinyatakan dalam formula ψ, maka verifikasi berbasis model merupakan verifikasi untuk memeriksa apakah model $\mathcal{M}$ memenuhi formula ψ.

Bagaimana Verifikasi Formal Dilakukan?

Verifikasi formal sistem dapat dilakukan dengan cara *fully automatic* (dilakukan dengan bantuan *tools*), *fully manual* (dilakukan oleh manusia), atau diantara keduanya.

Bagaimana Verifikasi Formal Dilakukan?

Verifikasi formal sistem dapat dilakukan dengan cara *fully automatic* (dilakukan dengan bantuan *tools*), *fully manual* (dilakukan oleh manusia), atau diantara keduanya.

Verifikasi sistem dapat dilakukan untuk seluruh spesifikasi sistem (*full-verification*) atau properti-properti tertentu pada sistem (*property-verification*).

Bagaimana Verifikasi Formal Dilakukan?

Verifikasi formal sistem dapat dilakukan dengan cara *fully automatic* (dilakukan dengan bantuan *tools*), *fully manual* (dilakukan oleh manusia), atau diantara keduanya.

Verifikasi sistem dapat dilakukan untuk seluruh spesifikasi sistem (*full-verification*) atau properti-properti tertentu pada sistem (*property-verification*).

Verifikasi formal sistem dapat dilakukan untuk *hardware* maupun *software*.

Bagaimana Verifikasi Formal Dilakukan?

Verifikasi formal sistem dapat dilakukan dengan cara *fully automatic* (dilakukan dengan bantuan *tools*), *fully manual* (dilakukan oleh manusia), atau diantara keduanya.

Verifikasi sistem dapat dilakukan untuk seluruh spesifikasi sistem (*full-verification*) atau properti-properti tertentu pada sistem (*property-verification*).

Verifikasi formal sistem dapat dilakukan untuk *hardware* maupun *software*.

Verfikasi formal sistem dapat dilakukan sebelum sistem dibuat (*pre-development*) atau setelah sistem dikontruksi (*post-development*).

Bahasan

Deskripsi Perkuliahan

- Nama mata kuliah: **Metode Formal**
- Kode mata kuliah: **CIG4F3**
- Status: **mata kuliah pilihan KK ICM**
- Bobot SKS: **3 SKS**
- *Pre-requisite* (prasyarat): **Logika Matematika (utama)**, Dasar Algoritma dan Pemrograman, Algoritma dan Struktur Data, Matematika Diskret
- *Co-requisite* (penunjang): Rekayasa Perangkat Lunak, Desain dan Analisis Algoritma

Tentang Pengajar Metode Formal

- Nama Lengkap: Muhammad Arzaki
- Tempat, tahun lahir: Surabaya, 1987
- Pendidikan:
  - SMAN 8 Bandung (Juli 2002 – Juni 2005)
  - Program Sarjana Matematika ITB (Agustus 2005 – Oktober 2009)
  - Program Magister Ilmu Komputer UI (Agustus 2010 – Januari 2012)
- Riwayat Riset dan Pengajaran:
  - *Research assistant* di *Formal Methods in Software Engineering Lab*, Fasilkom UI (September 2010 – Januari 2012)
  - *Research associate* di *Formal Methods in Software Engineering Lab*, Fasilkom UI (Februari 2012 – Agustus 2013)
  - *Teaching staff* untuk program sarjana ilmu komputer UI (Februari 2012 – Agustus 2013)
  - *Research associate* dan *teaching staff* di Fakultas Informatika Telkom University (Januari 2015 – sekarang).
- Ruang kerja: Ruang E 104 (Gedung Kultubai Utara/ Gedung E ruang 104).
- Email kontak: <mylastname>@telkomuniversity.ac.id

Bahasan

Apa, Kapan, dan Dimana

Slot jadwal kuliah reguler

- Selasa, pukul 12:30 – 14:30 di E 302
- Jumat, pukul 16:30 – 18:30 di A 208 B

Durasi kuliah dalam satu pertemuan adalah 45 menit – 90 menit. Jadwal responsi dilakukan pada salah satu *slot* yang tersedia di ruang kelas yang telah ditetapkan.

Bahasan

Aturan Penilaian dan Presensi

Nilai akhir terdiri atas komponen-komponen berikut:

- PR/ Tugas: **25%** (direncanakan 5 kali, masing-masing 5%)
- UTS: **35%**
- UAS: **35%**
- Lain-lain (presensi, keaktifan di kelas, keaktifan *e-learning*): **5%**.

Berdasarkan aturan institusi, mahasiswa wajib hadir minimal 75% dari seluruh pertemuan yang diadakan oleh dosen pengampu. Ketidakhadiran yang dikarenakan sakit harus disertai dengan surat dokter. Tidak ada ujian susulan (UTS/ UAS), kecuali karena alasan sakit, alasan keluarga yang mendesak, atau tugas dari institusi (lomba kegiatan mahasiswa yang bersifat resmi). Soal ujian susulan dapat lebih sulit daripada soal ujian reguler.

Ujian (UTS dan UAS) Metode Formal

Selama kuliah, mahasiswa diharuskan memiliki catatan kuliah secara individu yang ditulis dengan tulisan tangan (bukan hasil fotokopi atau tulisan/ ketikan orang lain). Catatan kuliah tersebut akan digunakan ketika ujian. Tidak diperkenankan untuk saling meminjam catatan dalam ujian.

Indeks Nilai Akhir

Indeks nilai akhir (NA) ditentukan oleh konversi berikut

$$
\begin{aligned}
80 < \text{NA} &\Rightarrow \text{nilai akhir } \mathbf{A} \\
70 < \text{NA} \leq 80 &\Rightarrow \text{nilai akhir } \mathbf{AB} \\
65 < \text{NA} \leq 70 &\Rightarrow \text{nilai akhir } \mathbf{B} \\
60 < \text{NA} \leq 65 &\Rightarrow \text{nilai akhir } \mathbf{BC} \\
50 < \text{NA} \leq 60 &\Rightarrow \text{nilai akhir } \mathbf{C} \\
40 < \text{NA} \leq 50 &\Rightarrow \text{nilai akhir } \mathbf{D} \\
\text{NA} \leq 40 &\Rightarrow \text{nilai akhir } \mathbf{E}
\end{aligned}
$$

Aturan indeks nilai akhir dapat berubah sesuai kesepakan kelas dan dosen pengampu. Tidak ada remedial berupa tugas tambahan atau ujian bila indeks nilai akhir telah keluar.

Bahasan

Referensi Materi Perkuliahan

Referensi perkuliahan yang dibuat oleh dosen (*slide* atau *handout*) akan diunggah secara berkala ke idea.telkomuniversity.ac.id, demikian pula dengan tugas maupun hasil-hasil evaluasi. Mahasiswa diharapkan mempelajari materi perkuliahan dari sumber-sumber berikut:

1. Anne Kaldewaij. *Programming: The Derivation of Algorithms*. Prentice Hall. 1990.
2. Mordechai Ben-Ari. *Mathematical Logic for Computer Science, 2nd Edition*. Springer Verlag. 2001.
3. Jean-François Monin and Michael G. Hinchey. *Understanding Formal Methods*. London: Springer Verlaag. 2003.
4. Michael Huth and Mark Ryan. *Logic in Computer Science: Modeling and Reasoning about System, 2nd Edition*. Cambridge University Press. 2004. (Referensi utama).
5. T. H. Cormen, et al. *Introduction to Algorithms, 3rd Edition*. MIT Press. 2009.
6. Michael Fischer. *Practical Formal Methods Using Temporal Logics*. John Wiley and Sons, Ltd. 2011.
7. Kenneth H. Rosen. *Discrete Mathematics and Its Applications, 7th Edition*. McGraw-Hill. 2012.

Rencana kegiatan per pekan dapat dilihat pada RPS yang diunggah ke `idea.telkomuniversity.ac.id`. Pembahasan logika proposisi dan logika predikat akan dilakukan dengan *slide* perkuliahan logika matematika yang dipakai pada semester ganjil 2015–2016.

Topik yang Dibahas

Materi yang rencananya akan dibahas:

1. Logika proposisi: sintaks tabel kebenaran, formula logika proposisi, semantik formula logika proposisi, inferensi pada logika proposisi.
2. Logika predikat: sintaks formula logika predikat, semantik formula logika predikat, inferensi pada logika predikat.
3. LTL (*linear-time temporal logic*): sintaks LTL, semantik LTL, pemodelan sistem dengan LTL.
4. CTL (*computation tree logic*): sintaks CTL, semantik CTL, pemodelan sistem dengan CTL.
5. Logika Hoare (*Hoare logic*) untuk verifikasi program imperatif.

Tools dan bahasa pemrograman yang akan digunakan.

1. NuSMV (*symbolic model checker*).
2. Prolog untuk pemrograman deklaratif (tentatif).
3. Eiffel untuk mempelajari *loop invariant* (tentatif).

Bahasan

Lain-lain

Pertanyaan atau masalah yang belum dibahas dalam rencana perkuliahan ini akan dibahas dan didiskusikan ketika masa perkuliahan berlangsung.