## Bab 2

### Metode Formal

* Apa itu metode formal (Definisi, Penemu, Filosofi)  
   Metode Formal didefinisikan sebagai matematika dari sistem perangkat lunak dan perangkat keras komputer (Holloway, XXXX). Metode Formal juga bisa dianggap sebagai matematika terapan yang digunakan untuk memodelkan dan menganalisis sistem ICT (Teknologi Komunikasi dan Informasi) (Baier, XXXX). Menurut Monin, sebenarnya istilah “Teknik Formal” lebih pantas untuk digunakan dibandingkan “Metode Formal” karena teknik ini belum memiliki metodologi yang baku. Namun karena istilah Metode Formal lebih populer maka untuk selanjutnya dalam tulisan ini istilah Metode Formal lah yang akn digunakan (Monin, XXXX).
* Kenapa metode formal [Holloway1997](#fn-holloway1997)
* Software Engineering jauh lebih tidak reliable dari other eng

Pengembangan perangkat lunak saat ini sudah terkenal sebagai sebuah proses yang lambat dalam memberikan hasil serta sulit diprediksi dan tidak bisa diandalkan dalam operasi (Holloway, 1997).

* Kenapa metode formal jarang dipake
* Contoh metode formal yang sudah dipake
* Kelemahan Metode Formal

#### Justifikasi Logis [Huth2004](#fn-huth2004)

* Aturan Logis
* Logika Propositional
* Logika Predikat
* SAT and SMT Solver

#### Spesifikasi Program

* Desain sebelum kode
* Test Driven Development

#### Verifikasi Program

* Verifikasi dengan Model Checking (Model Based)
* Verifikasi dengan Semantic Entailment (Proof Based)

### Pemrograman Fungsional

* Paradigma Pemrograman [Harper2017](#fn-harper2017)
* Paradigma Pemrograman Fungsional [Sabry1998](#fn-sabry1998)
* Paradigram Pemrograman Fungsional Murni
* Kenapa Fungsional Murni itu excellent buat Metode Formal [Turner1985](#fn-turner1985) [Hughes1989](#fn-hughes1989) [Butler1995](#fn-butler1995)
* Perbandingan dengan Paradigma lain

#### Haskell [O'Sullivan2008](#fn-o'sullivan2008)

* Definisi, Penemu, Filosofi [Hudak2007](#fn-hudak2007)
* Fungsional Murni
* Lazy Computing
* Hard Typing
* Program yang dibuat menggunakan Haskell (Hackage)
* How to Specify and Verify (Haskell for Specification)

#### Liquid Haskell [Pena2017](#fn-pena2017)

* Definisi, Penemu, Filosofi  
  Liquid Haskell merupakan sebuah type-checker untuk bahasa fungsional Haskell yang bertujuan untuk membantu memudahkan pemrogram untuk memverifikasi program mereka dengan usaha sesedikit mungkin. Untuk itu type-checker ini menggunakan sebuah teknologi bernama Liquid Types yang merupakan singkatan dari Logically Qualified Data Types. Pada dasarnya, teknologi ini menggunakan SMT Solver untuk secara instan memverifikasi sebuah program yang sudah dituliskan spesifikasinya dalam Liquid Types. Dengan itu, tidaklah lagi diperlukan waktu yang panjang untuk memverifikasi program dengan menuliskan persamaan matematika dengan tangan karena hal itu sudah bisa ditangani oleh komputer.
* Liquid Types
* Usage of SMT Solver in Liquid Haskell
* Kelemahan Liquid Haskell
* Refinement, Inference, and Polymorphism
* Liquid haskell Case Studies

#### Concurrency in Haskell