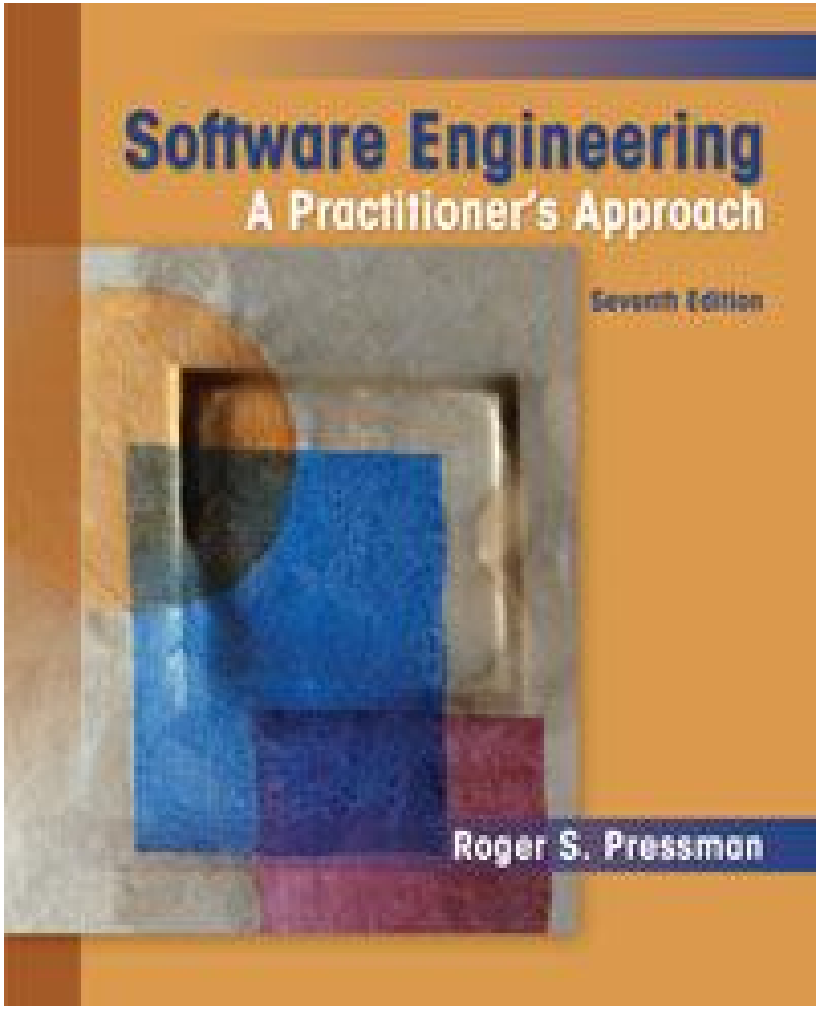
PERANGKAT LUNAK & REKAYASA PERANGKAT LUNAK

Defri Kurniawan M.Kom

Refrensi


Software Engineering
A Practitioner's Approach
Seventh Edition
Roger S. Pressman

content

- *Why Software Engineering*
- Perangkat Lunak (PL)
  - Definisi
  - Jenis-jenis berdasarkan Market, Platform, Licensi
- Rekayasa Perangkat Lunak (RPL)
  - Definisi
  - Lapisan
  - Mitos

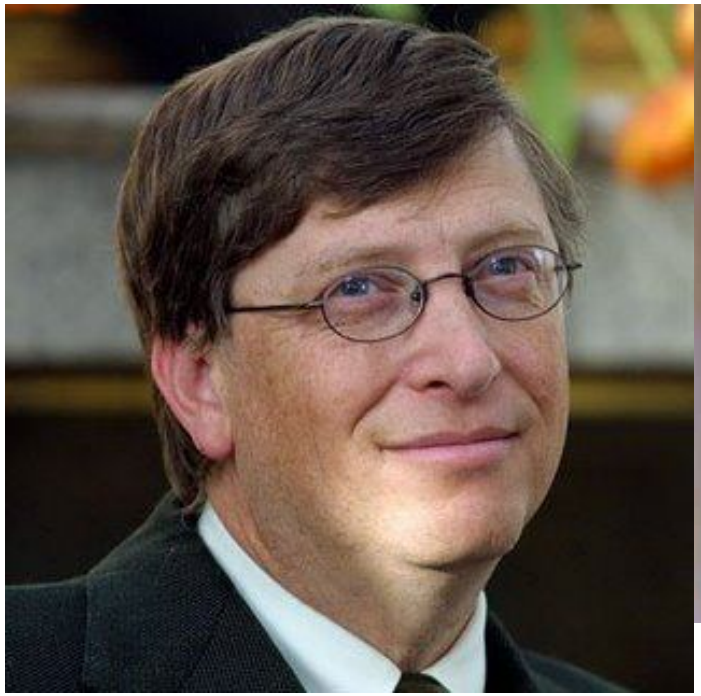
WHY SOFTWARE ENGINEERING???

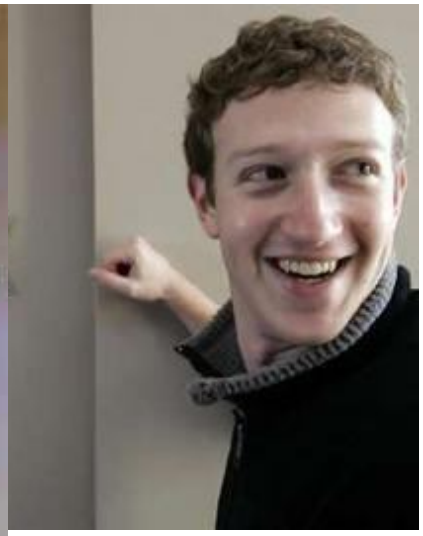
Why Software Engineering???

- Ingin mengembangkan produk (*software*) berkualitas
- Segala Aspek kehidupan dikendalikan oleh perangkat lunak
- Banyak pengusaha dunia sukses di bidang *Software Engineering*
- Usia Bidang kajian *Software Engineering* masih muda & terus berkembang

Pengusaha Dunia *Software Engineering*

- Who's the NEXT??





Perkembangan Ilmu Software Engineering

1. Peter J Dennings yang memimpin task force disiplin ilmu computing memasukkan software engineering sebagai satu disiplin ilmu (Dennings, 1999)
2. IEEE Computer Society membentuk tim khusus untuk menyusun pohon ilmu Software Engineering (Software Engineering Body of Knowledge, SWEBOK) *http://swebok.org*
3. Software Engineering termasuk nama jurusan atau fakultas yang diakui menurut IEEE Computing Curricula 2005

Matriks Dennings 1999

1. Algoritma dan Struktur Data
2. Bahasa Pemrograman
3. Arsitektur Komputer
4. Sistem Operasi dan Jaringan
5. Software Engineering
6. Database dan Sistim Retrieval Informasi
7. Artificial Intelligence dan Robotik
8. Grafik
9. Human Computer Interaction
10. Ilmu Komputasi
11. Organizational Informatics
12. BioInformatik

(Peter J. Dennings, 1999)

SWEBOK 2004

Software Requirement
Software Requirements Fundamentals Requirement Process Requirements Elicitation Requirements Analysis Requirements Validation Practical Considerations

Software Design
Software Design Fundamentals Key Issues in Software Design Software Structure and Architecture Software Design Quality Analysis and Evaluation Software Design Noations Software Design Strategies and Methods

Software Construction
Software Construction Fundamentals Managing Construction Practical Considerations

Software Testing
Software Testing Fundamentals Test Levels Test Techniques Test Related Measures Test Process

Software Maintenance
Software Maintenance Fundamentals Key Issues in Software Maintenance Maintenance Process Techniques for Maintenance

Software Configuration Management
Management of the SCM Process Software Configuration Identification Software Configuration Control Software Configuration Status Accounting Software Configuration Auditing Software Release Management and Delivery

Software Engineering Management
Initiation and Scope Definition Software Project Planning Software Project Enactment Review and Evaluation Closure Software Engineering Measurement

Software Engineering Process
Process Implementation and Change Process Definition Process Assesment Process and Product Measurement

Software Quality
Software Quality Management Software Quality Management Process Practical Considerations

Software Engineering Tools and Methods
Software Tools: Software Requirements Tools Software Design Tools Software Construction Tools Software Testing Tools Software Maintenance Tools Software Configuration Management Tools Software Engineering Management Tools Software Engineering Process Tools Software Quality Tools Miscellaneous Tool Issues Software Engineering Methods: Heuristic Methods Formal Methods Prototyping Methods

Knowledge Areas of the Related Disciplines
Computer Engineering Computer Science Management Mathematics Project Management Quality Management Software Ergonomic Systems Engineering

IEEE Computing Curricula 2005

1. Computer Engineering (*CE*, *Teknik Komputer*)
2. Computer Science (*CS*, *Ilmu Komputer*)
3. Information Systems (*IS*, *Sistem Informasi*)
4. Information Technology (*IT*, *Teknologi Informasi*)
5. Software Engineering (*SE*, *Rekayasa Perangkat Lunak*)

IEEE Computing Curricula 2005

Computer Engineering (CE)

pengembangan sistem terintegrasi(software dan hardware)

Computer Engineer

Information System (IS)

analisa kebutuhan proses bisnis serta desain siste

System Analyst

Computer Science (CS)

konsep computing dan pengembangan software

Computer Scientist

Information Technology (IT)

pengembangan dan maintenance infrastruktur IT

Network Engineer

Software Engineering (SE)

pengembangan software dan pengelolaan tahapan SDLC

Software Engineer

Perangkat Lunak - Definisi

- Perangkat Lunak adalah program komputer, prosedur, dokumentasi terkait dan data yang berkaitan dengan pengoperasian sistem komputer (*IEEE definition*)

- Perangkat Lunak komputer merupakan produk yang dibuat oleh para profesional perangkat lunak dalam jangka waktu panjang

Perangkat Lunak - Definisi

- Definisi *software* dari IEEE hampir identik dengan definisi ISO.
- Sesuatu dapat dikatakan *software*, jika memiliki 4 komponen seperti:
  - *Computer Program (the "code")*
  - *Procedures*
  - *Documentation*
  - *Data necessary for operating system computer*

(ISO, 1997, Sec. 3.11 an ISO/IEC 9000-3 Sec. 3.14)

Perangkat Lunak - Definisi

- Perangkat Lunak adalah (1) Instruksi-instruksi (program komputer) yang ketika dijalankan menyediakan fitur-fitur, fungsi-fungsi, dan kinerja-kinerja yang dikehendaki, (2) struktur data yang memungkinkan program-program memanipulasi informasi, dan (3) informasi deskriptif pada salinan tercetak dan bentuk-bentuk maya yang menggambarkan pengoperasian dan penggunaan program-program (Roger S.Pressman, 2010)

Perangkat Lunak – Jenis (Market)

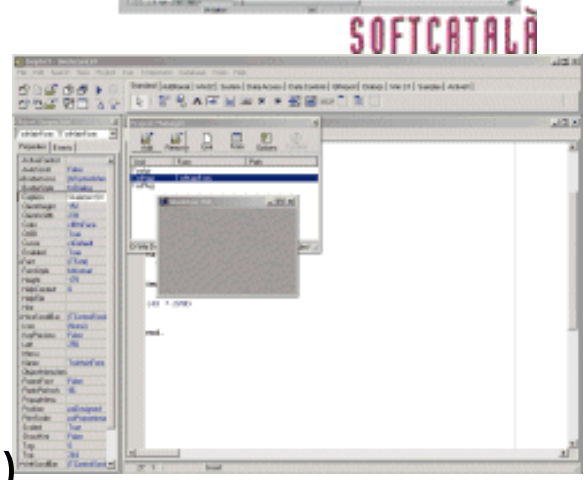
Software berdasarkan market

- *Software* Generik

  Perangkat lunak standar yang diproduksi oleh perusahaan pengembang dan dijual pada pasar terbuka ke siapapun yang bisa membelinya (*Shrink-wrapped*)

- *Software* Pesanan

  Perangkat lunak yang dikembangkan khusus dan disesuaikan dengan kebutuhan pelanggan



(Ian Sommerville, Software Engineering 9 Ed., 2012)

Perangkat Lunak – Jenis (Platform)

<u>*Software* berdasarkan Platform:</u>

- *Software* Sistem
- *Software* *Real-Time*
- *Software* Bisnis
- *Software* Teknik dan Ilmu Pengetahuan
- *Software* Tertanam (*Embedded Software*)
- *Software* Komputer Personal
- *Software* Kecerdasan Buatan
- *Software* *Mobile*

(Roger Pressman, Software Engineering,: A Practitioner's Approach 7Ed., 2009)

Perangkat Lunak – Jenis (Licensi)

Software Berdasarkan Licensi

1. *Proprietary Software*

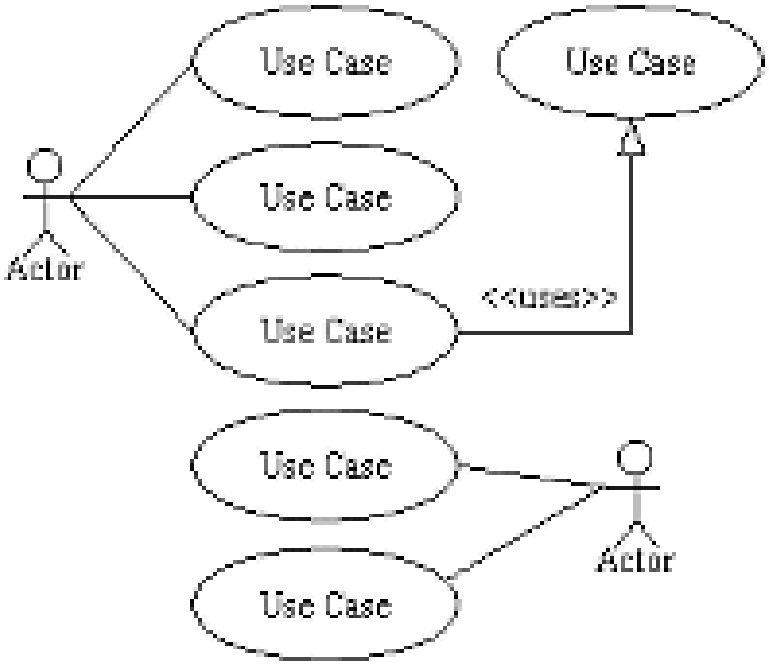
   Software yang *source code*-nya tertutup dan didistribusikan dengan suatu format lisensi yang membatasi pihak lain untuk menggunakan, memperbanyak dan memodifikasi

2. *Open Source Software*

   Software yang *source code*-nya terbuka dan didistribusikan dalam suatu format lisensi yang memungkinkan pihak lain secara bebas memperbanyak dan memodifikasi *source code* didalamnya

Rekayasa Perangkat Lunak - Definisi

Disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi, desain, konstruksi, testing sampai pemeliharaan setelah digunakan



Rekayasa Perangkat Lunak - Definisi

- Rekayasa Perangkat Lunak meliputi di dalamnya suatu proses, kumpulan metode-metode dan sederetan perkakas (*tools*) yang memungkinkan para profesional mengembangkan perangkat lunak berkualitas tinggi

Rekayasa Perangkat Lunak - Definisi

- IEEE [IEE93a] menyatakan: Rekayasa Perangkat Lunak pada dasarnya merupakan: (1) aplikasi dari suatu pendekatan yang sistematik, disiplin dan dapat diukur pada pengembangan, operasi, dan perawatan perangkat lunak: yaitu penerapan rekayasa pada perangkat lunak. (2) Studi pendekatan-pendekatan seperti pada (1).

Rekayasa Perangkat Lunak - Lapisan

- Rekayasa Perangkat Lunak yang sesungguhnya merupakan teknologi yang berlapis

Gambar Lapisan-lapisan (*layers*)
Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak - Lapisan

Perkakas menyajikan dukungan-dukungan bagi proses dan metode

Metode menyajikan prosedur-prosedur teknis untuk mengembangkan/ membangun perangkat lunak. Mencakup di dalamnya analisis kebutuhan, pemodelan, rancangan, pembuatan program, pengujian & dukungan terhadap pengguna

Segala pendekatan rekayasa (termasuk rekayasa perangkat lunak) harus bersandar pada komitmen organisasi pada peningkatan kualitas

Proses mendifinisikan *framework* (kerangka kerja) yang harus dibangun & membentuk dasar bagi kendali menegement proyek perangkat lunak

System Engineering, Software Engineering dan Sistem Komputer

- *System engineering* mempunyai kaitan dengan semua aspek pengembangan sistem berbasis komputer yang mencakup perangkat keras, perangkat lunak ,dan yang terkait dengan proses bisnis.

- *Software engineering* berkonsentrasi pada komponen perangkat lunak sistem yang lebih besar

Sistem Komputer

- **Sistem komputer** adalah elemen-elemen yang terkait untuk menjalankan suatu aktifitas dengan menggunakan komputer.

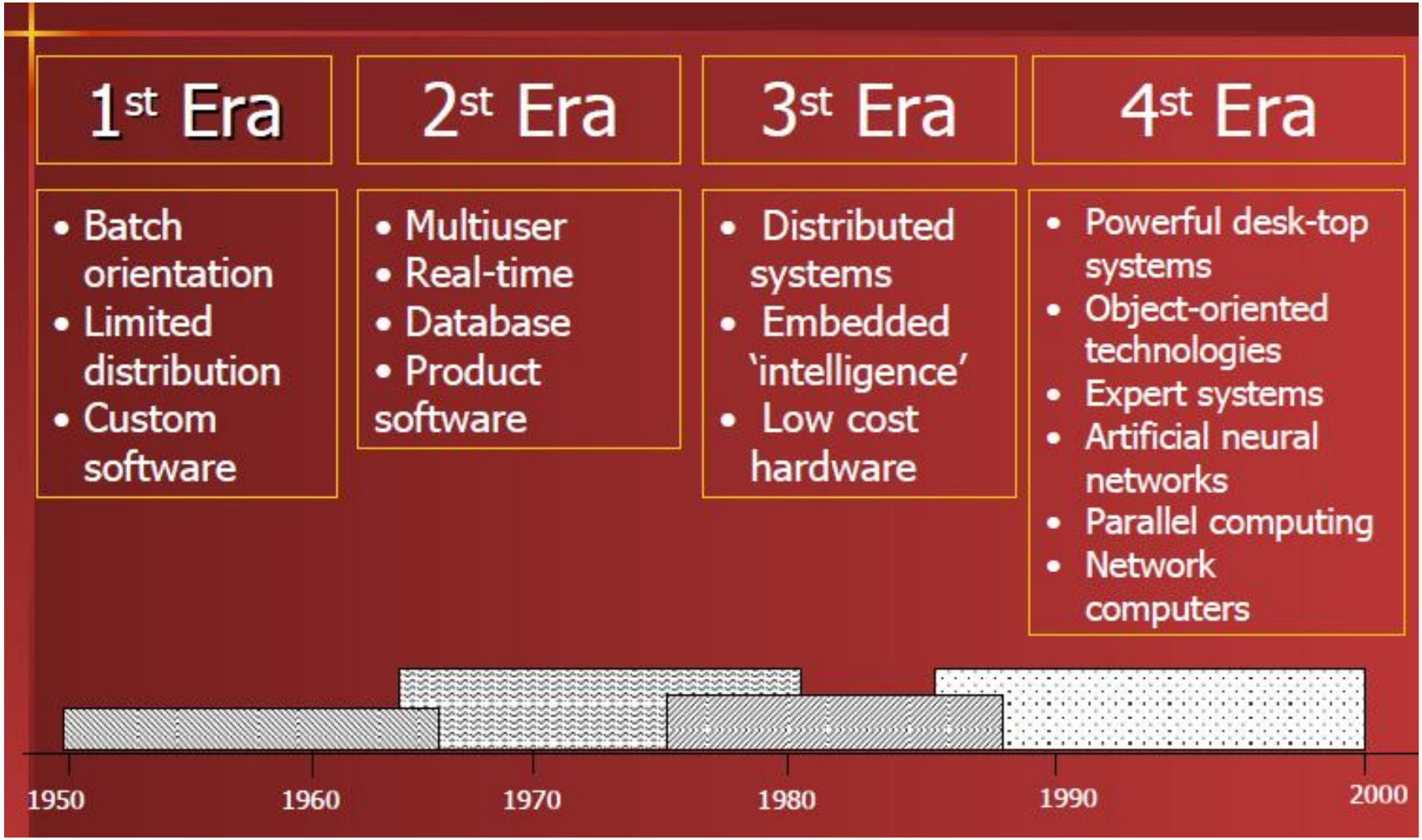
- **Sistem Komputer** pada dasarnya **terdiri dari tiga komponen utama** yaitu :

  - Perangkat Keras (Hardware)

  - Perangkat Lunak (Software)

  - Brainware (User/Sumber Daya Manusia)

Perkembangan Perangkat Lunak



Permasalahan Perangkat Lunak

- Kemajuan perangkat keras terus berlanjut
- Tidak dapat memenuhi kebutuhan bisnis dan pasar yang begitu cepat
- Masyarakat bergantung pada perangkat lunak, dan mempengaruhi ekonomi bila perangkat lunak gagal
- Belum banyak dijumpai perangkat lunak yang berkualitas tinggi (masih berjuang)
- Buruknya desain dan sumber daya yang tidak memadai

Mitos-mitos Perangkat Lunak

- Software Miths (mitos-mitos perangkat lunak) adalah asumsi-asumsi permasalahan yang kebenarannya tidak dapat dipertanggungjawabkan berkaitan dengan pengembangan perangkat lunak
- Tiga kelompok yang terkait dalam pengembangan perangkat lunak
  - Management : manajer yang bertanggungjawab terhadap pengembangan perangkat lunak
  - Customer : pelanggan yang memesan perangkat lunak
  - Practitioner's : praktisi yang mengembangkan perangkat lunak

Mitos Management

- Dengan memiliki buku berisi standard dan prosedur yang banyak untuk pengembangan perangkat lunak, maka pekerjaan pasti lancar.
  - *Buku-buku itu memang lengkap, tapi apakah digunakan ? Apakah praktisi perangkat lunak sadar dengan keberadaannya. Apakah cocok dengan pengembangan yang modern ? Apakah benar-benar lengkap ?*
- Untuk menghasilkan perangkat lunak yang berkualitas, maka kita perlu membeli komputer terbaru.
  - *Untuk mendapatkan perangkat lunak yang berkualitas, CASE tools lebih penting daripada perangkat keras.*
- Bila terlambat maka tambahlah jumlah programer
  - *Penambahan programmer semakin menambah keterlambatan.*

Mitos-mitos Customer

- Tujuan sistem secara umum cukup untuk memulai menulis program, rincian belakangan saja.
  - *Definisi awal yang buruk merupakan sebab utama gagalnya kerja perangkat lunak*
  - *Rincian kebutuhan sistem sangat penting:*
    - *fungsi*
    - *performance*
    - *antar-muka*
    - *batasan rancangan*
    - *kriteria validasi*
    - *dll*
- Perangkat lunak bersifat fleksibel, perubahan kebutuhan mudah diakomodasi oleh pengembang perangkat lunak
  - *Dampak perubahan sangat bergantung pada tahap mana perubahan terjadi*

Mitos-mitos Praktisi

- Program selesai, pekerjaan selesai
  - *50% - 70% usaha dihabiskan setelah program diserahkan ke user untuk pertama kalinya.*
- Kualitas hanya bisa diketahui setelah program berjalan (running)
  - *Kualitas dapat dijaga sejak PL dikembangkan.*
- Yang diserahkan ke user adalah program
  - *Yang diserahkan adalah program, dokumen, dan data.*

***