

REKAYASA PERANGKAT LUNAK

PENDAHULUAN

MATERI:

- Pendahuluan
- Pengantar Rekayasa Perangkat Lunak
- Proses Pembangunan Perangkat Lunak
- Rekayasa Kebutuhan Perangkat Lunak
- Analisis dan Desain Perangkat Lunak (Terstruktur)
- Pengujian Perangkat Lunak
- Pemeliharaan Perangkat Lunak

REFERENSI

- Roger S. Pressman, *“Software Engineering: A Practitioner’s Approach, 6th edition”*.
- Ian Sommerville, *“Software Engineering, 8th edition”*.
- Kendall, *“System Analysis and Design, 8th edition”*.
- Slide perkuliahan.

Rekayasa adalah penerapan ilmu dan teknologi untuk menyelesaikan permasalahan manusia. Hal ini diselesaikan lewat pengetahuan, matematika dan pengalaman praktis yang diterapkan untuk mendesain objek atau proses yang berguna. Para praktisi teknik profesional disebut insinyur.

Contoh :

1. Rekayasa Genetika
2. Rekayasa Cuaca

PENDAHULUAN

Why we need to learn software

engineering? What is software and software

engineering?

Software process

Criteria of Good

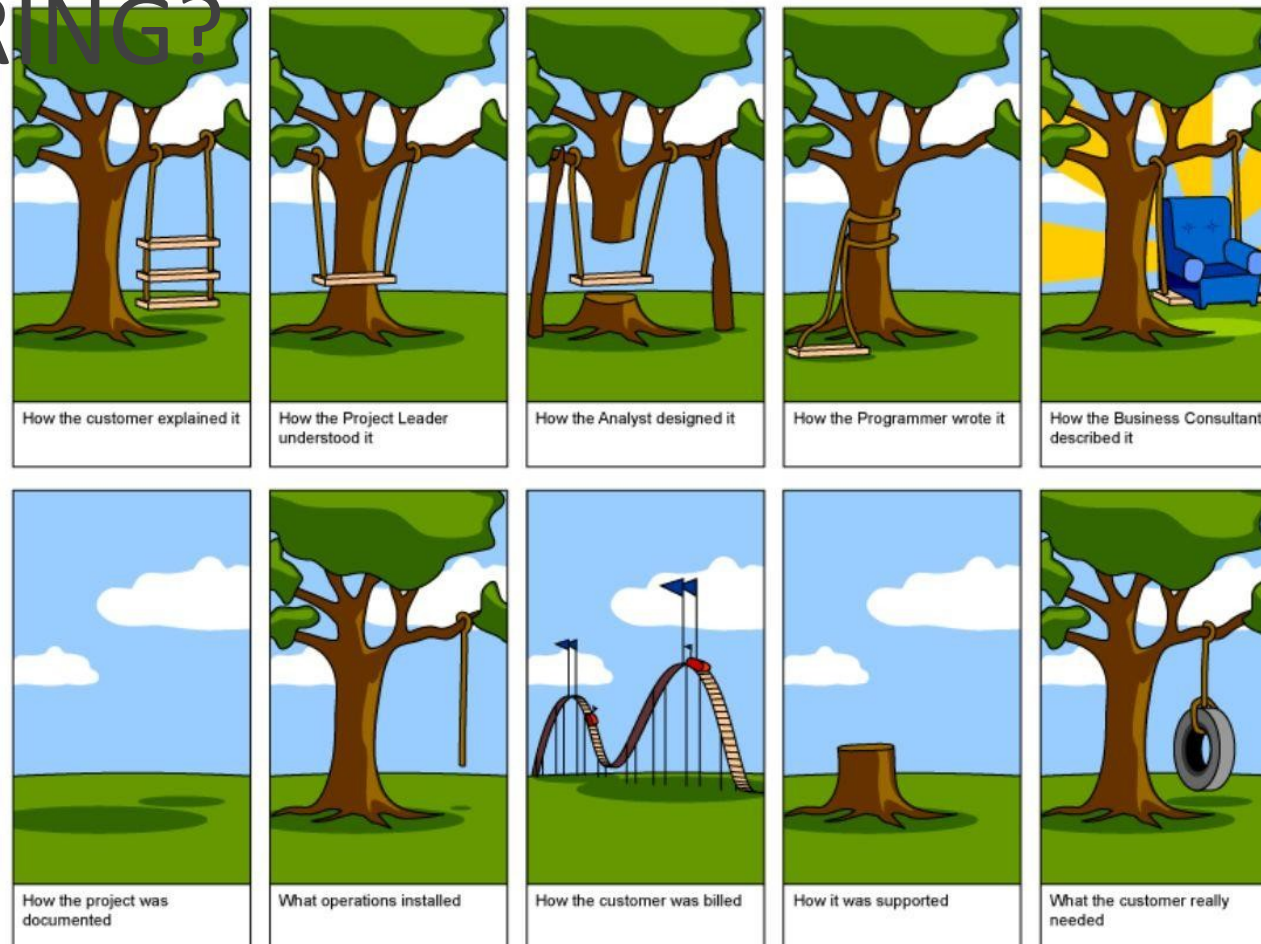
Software CASE Tools

Rekayasa Perangkat Lunak adalah satu bidang profesi yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak dan manajemen kualitas

WHY WE NEED TO LEARN SOFTWARE ENGINEERING?

Tujuan dan juga fungsi yang pertama adalah agar seseorang yang mempelajari rekayasa **perangkat lunak** ini mampu mengembangkan sebuah **perangkat lunak** yang berguna dan juga bermanfaat bagi user-nya. Sebuah **perangkat lunak** tentu saja tidak akan digunakan oleh user apabila tidak memiliki fungsi yang spesifik.

WHY WE NEED TO LEARN SOFTWARE ENGINEERING?



WHAT IS SOFTWARE?

- Instructions (computer programs) that when executed provide desired features, function, and performance;
- Data structures that enable the programs to adequately manipulate information;
- Documentation that describes the operation and use of the programs.

WHAT IS SOFTWARE?

- Software is developed or engineered, it is not manufactured in the classical sense.
- Software doesn't "wear out."
- Although the industry is moving toward component-based construction, most software continues to be custom-built.
- Generic → Public software & Bespoke (custom) → Private software

SOFTWARE APPLICATIONS

- system software
- application software
- engineering/scientific software
- embedded software
- product-line software
- WebApps (Web applications)
- AI software
- Etc.

Apa Itu System Software?

System software adalah jenis *software* yang dirancang untuk berkomunikasi dengan *hardware* atau bahkan *application software*. Ini bertindak sebagai *interface* antara *hardware* dan program komputer di mana ia mengkoordinasikan tugas-tugas antara dua komponen sistem komputer. Ada berbagai jenis *system software*. Mereka termasuk *operating system*, *language processors* dan *device drivers*. *System software* tipikal memastikan bahwa *hardware* menangani tugasnya dengan sukses. Misalnya, *operating system* melakukan manajemen memori dan penanganan file untuk *hardware*. Driver mengendalikan kinerja monitor dan perangkat lain seperti printer.

Apa Itu Application Software?

Application Software adalah jenis *software* yang dirancang untuk menyelesaikan tugas-tugas spesifik dalam *operating system*. Beberapa dari tugas ini termasuk menangani dokumen, menangani perhitungan, mengedit gambar, mengedit video, dan lainnya. Contoh *application software* termasuk pengolah kata, *spreadsheet*, *database software*, editor gambar.



Perbedaan Antara System Software Dengan Application Software

Subjek

System Software

Application Software

Pemakaian

System software digunakan untuk mengelola dan mengendalikan *hardware* komputer dan *application software*.

Application software melakukan tugas tertentu tergantung pada bagaimana ia dirancang. Ini tidak dapat melakukan tugas lain yang berada di luar cakupannya.

Instalasi

System software diinstal ke komputer ketika *operating system* diinstal.

Application software diinstal hanya ketika pengguna membutuhkannya. Misalnya, kamu hanya akan menginstal *software* edit foto ketika kamu ingin mengedit beberapa foto.

Waktu eksekusi

System software akan mulai berjalan setelah kamu menyalakan komputer. Ini akan terus berjalan hingga kamu mematikan komputer.

Application software hanya akan berjalan jika diminta. Kamu akan memulai pengolahan kata ketika kamu perlu membuka atau menyiapkan beberapa dokumen.

Abstraksi

Dari sudut pandang umum, pengguna tidak berinteraksi dengan *system software*. Ini karena software berfungsi di latar belakang.

Untuk *application software*, pengguna akan memiliki interaksi langsung dengan *software*. Bahkan, pengguna mengontrol cara *software* beroperasi.

Ketergantungan

System software berjalan secara independen. Pengoperasiannya tidak tergantung pada perilaku *software* lain. Sebaliknya, itu adalah salah satu yang menyediakan kondisi untuk *software* lain untuk berjalan secara efektif.

Application software tidak dapat berjalan secara independen. Mereka tergantung pada kondisi yang disediakan oleh *operating system*.

Kompleksitas

Memprogram *system software* cukup rumit. Ini karena membutuhkan programmer untuk memiliki pengetahuan penuh tentang *system hardware* yang mendasarinya dan bahasa tingkat rendah.

Pemrograman *application software* tidak rumit. Seorang programmer hanya perlu memiliki pengetahuan tentang *system software* yang mendasarinya dan bahasa tingkat tinggi

LEGACY SOFTWARE

Why must it change?

- software must be **adapted** to meet the needs of new computing environments or technology.
- software must be **enhanced** to implement new business requirements.
- software must be **extended to make it interoperable** with other more modern systems or databases.
- software must be **re-architected** to make it viable within a network environment.

WHAT IS SOFTWARE ENGINEERING?

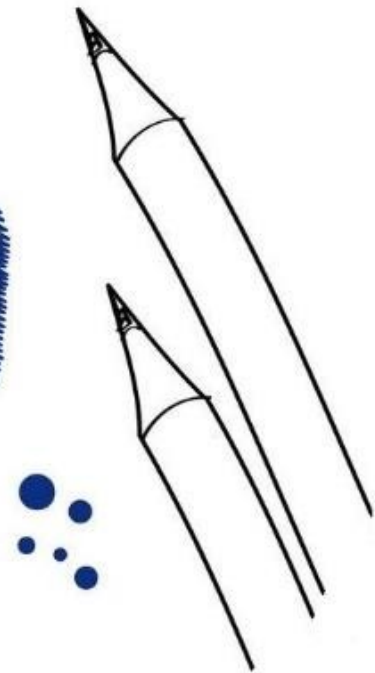
- Disiplin ilmu rekayasa atau teknik yang berkaitan dengan semua aspek dalam membuat perangkat lunak.
- The IEEE definition:
 - Software Engineering: “(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1)”.

WHAT IS SOFTWARE ENGINEERING?

PLANNING ANALYZE DESIGN

**SOFTWARE
ENGINEERING**

VALIDATION
AND VERIFICATION



SOFTWARE PROCESS

- Serangkaian aktifitas yang tujuannya adalah pembangunan atau evolusi perangkat lunak dengan aktifitas:
 - Spesifikasi
 - Pembangunan
 - Validasi
 - Evolusi



CRITERIAS OF GOOD SOFTWARE

Maintainability

- Software must evolve to meet changing needs

Dependability

- Software must be trustworthy

Efficiency

- Software should not make wasteful use of system resources

Usability

- Software must be usable by the users for which it was designed

CASE (Computer Aided Software Engineering) TOOLS

- **Computer-aided software engineering (CASE)** is the domain of software tools used to design and implement applications.
- CASE tools are similar to and were partly inspired by computer-aided design (CAD) tools used for designing hardware products.
- CASE tools are used for developing high-quality, defect-free, and maintainable software.
- CASE software is often associated with methods for the development of information systems together with automated tools that can be²⁵

CASE (Computer Aided Software Engineering) TOOLS

- Tools untuk mendukung aktifitas proses awal

Lower-CASE

- Tools untuk mendukung aktifitas selanjutnya seperti programming, debugging, dan testing.