

1. INTRODUCTION TO SOFTWARE ENGINEERING

1. What and Why Software Engineering ?



1.1 Software Engineering

(Rekayasa Perangkat Lunak)

- **Ekonomi** dari semua bangsa-bangsa maju tergantung pada perangkat lunak
- Semakin banyak **sistem yang dikendalikan** oleh perangkat lunak
- Rekayasa Perangkat Lunak mempunyai **kaitan dengan teori, metode, dan perkakas (tools)** untuk pengembangan perangkat lunak profesional
- Rekayasa Perangkat Lunak sudah menjadi bagian yang penting untuk menghadirkan **pendapatan nasional pada semua negara maju**

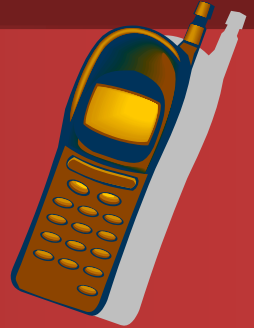
1.2 Software Costs

(Biaya-Biaya Perangkat Lunak)

- Biaya-biaya perangkat lunak sering mendominasi biaya-biaya sistem. Biaya-biaya perangkat lunak pada suatu PC sering lebih besar dari harga perangkat keras.
- Biaya-biaya perawatan perangkat lunak lebih besar dibanding dengan pengembangan perangkat lunak, karena sistem dengan masa pakai lama, biaya pemeliharaan mungkin beberapa kali biaya-biaya pengembangan.
- Rekayasa Perangkat Lunak mempunyai kaitan dengan biaya-biaya pengembangan perangkat lunak yang ekonomis.

1.3 FAQs about Software Engineering (*Pertanyaan-pertanyaan Seputar SE*)

- Apakah software itu?
- Apakah software engineering itu?
- Apa perbedaan antara software engineering dan computer science?
- Apa perbedaan antara software engineering dan system engineering?
- Apakah software process itu?
- Apakah software process model itu?



FAQs about Software Engineering

(Lanjutan)

- Apa saja yang merupakan biaya-biaya rekayasa perangkat lunak itu?
- Apa saja metode rekayasa perangkat lunak itu?
- Apakah CASE (Computer-Aided Software Engineering) itu?
- Apa saja atribut dari perangkat lunak yang baik?
- Apakah yang merupakan tantangan kunci dalam menghadapi rekayasa perangkat lunak?



What is software?

- **perintah** (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan;
- **struktur data** yang memungkinkan program memanipulasi informasi secara proporsional; dan
- **dokumen** yang menggambarkan operasi dan kegunaan program.
- Produk Perangkat lunak mungkin :
 - **Generic** (*Umum*) - yang dikembangkan untuk dijual ke bidang pelanggan berbeda;
 - **Bespoke/Custom** (*Pesanan*) - dikembangkan untuk pelanggan tunggal menurut spesifikasi mereka.



What is software engineering?

- *Software engineering* adalah suatu disiplin rekayasa (rancang-bangun) yang terkait dengan semua **aspek produksi** perangkat lunak.
- Engineer perangkat lunak mengadopsi pendekatan **sistematis** dan **terorganisir** untuk pekerjaan mereka dan menggunakan **teknik dan tools** yang disesuaikan dengan masalah yang dihadapi untuk dipecahkan, batasan pengembangan, dan sumber daya tersedia.

IEEE Definition

(IEEE = Institute of Electrical and Electronic Engineers)

Software engineering adalah:

1. **Aplikasi** dari sebuah pendekatan yang bersifat kuantifiabel, disiplin, dan sistematis bagi pengembangan, operasi, dan pemeliharaan perangkat lunak.
2. **Studi** tentang pendekatan-pendekatan seperti pada (1)

Bidang Penelitian Software Engineering mengacu pada kedua hal tsb.



What is the difference between *software engineering* and *computer science*?

- *Computer science* mempunyai kaitan dengan *theory and fundamentals*; *software engineering* mempunyai kaitan dengan *practicalities of developing and delivering useful software*.
- *Computer science* sekarang ini tidak cukup lengkap untuk bertindak sebagai tiang penyokong *software engineering*.



What is the difference between *software engineering* and *system engineering*?

- *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.
- *System engineers* mencakup spesifikasi sistem, desain arsitektur, pengintegrasian, dan penyebaran.



What is a software process?

- Software process merupakan himpunan **aktivitas tujuan pengembangan atau evolusi** perangkat lunak.
- Aktivitas umum dalam semua proses perangkat lunak adalah:
 - *Specification* (Spesifikasi)- hal-hal yang diperlukan oleh sistem dan batasan pengembangannya.
 - *Development* (Pengembangan)- produksi sistem perangkat lunak.
 - *Validation* (Pengesahan) - pemeriksaan perangkat lunak sesuai dengan keinginan pelanggan.
 - *Evolution* (Evolusi) - pengubahan perangkat lunak sesuai dengan permintaan pelanggan.



What is a software process model?

- Software process model merupakan **representasi** sederhana suatu *software process*, yang diperkenalkan dari suatu perspektif spesifik.
- Contoh perspektif proses adalah
 - *Workflow Perspektif* - Urutan aktivitas
 - *Data-Flow Perspektif* - Arus Informasi
 - *Role/Action Perspektif* – Peran dan Aksi
- Proses umum model
 - *Waterfall*
 - *Evolutionary development*
 - *Formal transformation*
 - *Integration from reusable components*



What are the costs of software engineering?

- Perkiraan kasar adalah **60% untuk biaya pengembangan**, sedangkan **40% untuk biaya pengujian**. Untuk custom software, biaya-biaya evolusi sering melebihi biaya-biaya pengembangan.
- Biaya-biaya berubah-ubah tergantung pada **jenis sistem** yang dikembangkan dan kebutuhan atribut sistem seperti **kehandalan dan reliabilitas** sistem.
- Distribusi biaya-biaya tergantung pada **model pengembangan** yang digunakan.



What are software engineering methods?

Software engineering methods merupakan **pendekatan terstruktur** dalam pengembangan perangkat lunak yang meliputi model sistem, notasi, aturan, desain *advice*, dan panduan proses.

- **Model Descriptions** (Uraian Model)
Uraian tentang model grafis yang harus diproduksi.
- **Rules** (Aturan-aturan)
Batasan yang berlaku pada model sistem.
- **Recommendations** (Rekomendasi)
Rekomendasi untuk praktik desain yang baik.
- **Process guidance** (Panduan Proses)
Aktivitas yang mengikuti.



What is CASE (*Computer-Aided Software Engineering*)?

- CASE adalah *System software* yang digunakan untuk mendukung otomatisasi aktivitas proses perangkat lunak. CASE sering digunakan untuk mendukung metode.
- Upper-Case
Tools untuk mendukung aktivitas proses awal kebutuhan dan desain.
- Lower-Case
Tools untuk mendukung aktivitas selanjutnya seperti *programming, debugging, dan testing*.



What are the attributes of good software?

Software perlu memiliki fungsi kebutuhan dan kemampuan yang diperlukan oleh pemakai dan harus *maintainable*, *dependable*, *efficient*, dan *usable*.

- **Maintainability**

Software harus dapat ditingkatkan dan diubah sesuai dengan kebutuhan.

- **Dependability**

Software harus dapat dipercaya (trustworthy).

- **Efficiency**

Software seharusnya tidak membuat penggunaan sumber daya sistem menjadi boros.

- **Usability**

Software harus dapat dipakai oleh para pemakai yang direncanakan.



What are the key challenges facing software engineering?

Tantangan : mengatasi sistem warisan (*legacy systems*), meningkatnya heterogenitas (*Heterogeneity*) sistem, dan tuntutan permintaan percepatan penyerahan(*Delivery*) sistem.

- **Legacy systems**

Sistem warisan (sistem lama) harus dirawat dan dibaharui.

- **Heterogeneity**

Sistem terdistribusikan dalam bentuk **campuran antara perangkat keras dan lunak**.

- **Delivery**

Adanya peningkatan **tekanan** untuk penyerahan perangkat lunak lebih cepat.

1.4 Professional and Ethical Responsibility

- *Software engineering* melibatkan tanggung-jawab **lebih luas** dibanding hanya aplikasi kecakapan **teknis**.
- *Software engineer* harus bertindak secara **etis, bertanggung jawab, dan jujur** jika mereka diharapkan untuk terhormat sebagai seorang profesional.
- Perilaku etis tidak hanya sekedar menegakkan hukum saja tetapi harus lebih dari itu (lih. hal. berikutnya).

Issues of professional responsibility

- **Confidentiality** (Kerahasiaan)

Engineer seharusnya **menghormati kerahasiaan** dari klien mereka tanpa tergantung dengan ya atau tidaknya suatu persetujuan kerahasiaan formal ditandatangani.

- **Competence** (Kemampuan)

Engineer mestinya tidak salah menggambarkan tingkatan kemampuannya. Mereka mestinya tidak dengan sadar menerima pekerjaan yang **di luar kemampuannya**.

Issues of professional responsibility (lanjutan)

- **Intellectual property rights** (Hak milik intelektual)
Engineers harus **sadar akan hukum lokal** yang mengatur penggunaan dari properti intelektual seperti hak paten, hak cipta, dll. Mereka harus seksama untuk memastikan bahwa intelektual properti klien harus dilindungi.
- **Computer misuse** (Penyalahgunaan Komputer)
Software engineers mestinya tidak menggunakan kecakapan teknis mereka untuk **menyalahgunakan komputer orang lain**. Penyalahgunaan komputer dari yang relatif sepele (misal untuk bermain game) sampai yang serius (pemberian virus).

