

REKAYASA PERANGKAT LUNAK

(Software engineering)

Mohamad Sidiq
Teknik Informatika
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro

Rencana Pelaksanaan Pembelajaran

| | |
|------------------|---|
| Nama Mata Kuliah | Rekasaya Perangkat Lunak |
| Satuan Kredit | 3 SKS (150 menit dalam seminggu) |
| Program Studi | Teknik Informatika S1 |
| Deskripsi | Mata kuliah ini menjelaskan tentang Konsep Dasar Perangkat Lunak, Daur Hidup, dan Tahapan Pembangunan Perangkat Lunak, mulai dari tahapan requirement, analisis, perancangan, konstruksi, pengujian, dan perawatan dengan menggunakan metode terstruktur yang dilengkapi studi kasus untuk setiap tahapannya. |
| Tujuan | Setelah mengikuti dan menyelesaikan mata kuliah ini, mahasiswa diharapkan dapat menggunakan konsep dan metode rekayasa perangkat lunak terstruktur. |

Rencana Pelaksanaan Pembelajaran

| No | Pokok Bahasan | Materi Bahasan |
|-----|--------------------------------------|--|
| 1 | Introduction to Software Engineering | <ul style="list-style-type: none">a. Perkembangan rekayasa perangkat lunakb. Perkembangan perangkat lunakc. Permasalahan perangkat lunakd. Jenis Aplikasi Perangkat Lunake. Mitos Perangkat Lunakf. Perbedaan Software, Software Engineering, System Engineering, Computer Systemg. Isu dari Tanggungjawab Profesional |
| 2,3 | Software Process | <ul style="list-style-type: none">a. Process Frameworkb. CMMI (Capability Maturity Model Integration)c. Process Assessmentd. Process Model (Waterfall Model, Increment Model, RAD Model), Evolutionary Process Model (Prototyping Model, Spiral Model)e. An Agile View of Process |

Rencana Pelaksanaan Pembelajaran

| No | Pokok Bahasan | Materi Bahasan |
|-----|-------------------------------|--|
| 4 | Software Engineering Practice | <ol style="list-style-type: none">Core and the essence of practice Software EngineeringPlanning PracticesModeling PracticesConstruction PracticesComputer-Based System ElementSystem Engineering HierarchySystem Model Process |
| 5,6 | Requirement Engineering | <ol style="list-style-type: none">Requirement Engineering Task (inception, elicitation, elaboration, negotiation, specification, validation, requirement management)Initiating The Requirements Engineering Process.Collaborative Requirements GatheringDeveloping Use Case |

Rencana Pelaksanaan Pembelajaran

| No | Pokok Bahasan | Materi Bahasan |
|----|--------------------|--|
| 7 | Analysis Modeling | <ol style="list-style-type: none">Requirement AnalysisAnalysis Modeling ApproachesData Modeling ConceptsObject Oriented Analysis |
| 8 | Design Modeling | <ol style="list-style-type: none">Scenario-Based ModelingFlow Oriented ModelingClass-Based ModelingClass-Responsibility (CRC) Modeling |
| 9 | Design Engineering | <ol style="list-style-type: none">Translating the analysis model into a software designDesign Process and Design QualityDesign Concepts (Abstract, Architecture, Modularity, Information Hiding, Functional Independence, Refinement)Design Model |

Rencana Pelaksanaan Pembelajaran

| No | Pokok Bahasan | Materi Bahasan |
|-----------|---------------------------------|--|
| 10, 11 | Creating an Architecture Design | <ol style="list-style-type: none">Software ArchitectureData designArchitecture Styles and PatternArchitecture DesignMapping Data Flow into a Software Architecture |
| 12 | Interface Design | <ol style="list-style-type: none">The Golden RulesUser Interface Analysis and DesignInterface AnalysisInterface Design Steps |
| 13 | Testing Strategy | <ol style="list-style-type: none">Strategy Approach to Software TestingTest Strategy for Conventional SoftwareTest Strategy for Object-Oriented SoftwareValidation TestingSystem Testing |

Rencana Pelaksanaan Pembelajaran

| No | Pokok Bahasan | Materi Bahasan |
|----|----------------|--|
| 14 | Testing Tactic | a. Software Testing Fundamentals b. Black-Box and White-Box c. Object-Oriented Testing Methods |

Referensi

| No | Judul | Pengarang | Penerbit |
|----|---|-------------------|-------------|
| 1 | Software Engineering A Practitioner's Approach, 8th edition | Roger S. Pressman | McGraw-Hill |
| 2 | Software Engineering, 10th edition | Ian Sommerville | Pearson |

Rencana Pelaksanaan Pembelajaran

Metode Pembelajaran & Penilaian

| No | Metode Pembelajaran |
|----|----------------------------------|
| 1 | Ceramah |
| 2 | Diskusi |
| 3 | Penugasan (Mandiri dan Kelompok) |
| 4 | Praktek Mandiri Tool untuk RPL |

| No | Komponen Penilaian | Persentase |
|----|-----------------------|------------|
| 1 | Kehadiran | 5 % |
| 2 | Tugas Mandiri | 10 % |
| 3 | Tugas Kelompok | 20 % |
| 4 | Ujian Tengah Semester | 30 % |
| 5 | Ujian Akhir Semester | 35 % |
| | Nilai Akhir Semester | 100 % |

Introduction to Software Engineering

(Pengantar Rekayasa Perangkat Lunak)

Introduction to Software Engineering

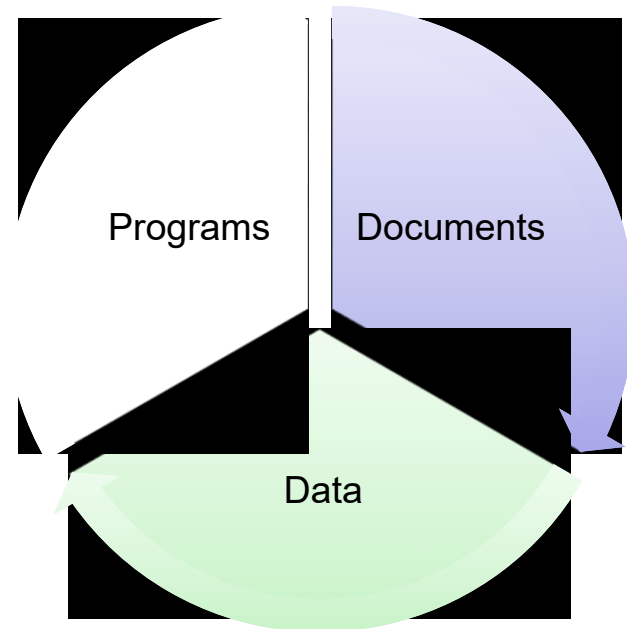
- Apakah perangkat lunak itu?
- Apakah peran perangkat lunak?
- Bagaimana perkembangan perangkat lunak?
- Apakah permasalahan perangkat lunak?
- Apakah jenis aplikasi perangkat lunak?
- Apakah perbedaan antara perangkat lunak (software), rekayasa perangkat lunak (software engineering) dan rekayasa sistem (system engineering)?
- Apa isu dan tanggungjawab professional perekraya perangkat lunak?

Perangkat Lunak (Software)

Software adalah sekelompok item atau objek yang membentuk konfigurasi di mana di dalamnya termasuk:

- Program : sekumpulan instruksi yang ketika dieksekusi akan memberi fungsi dan hasil yang diinginkan.
- Data : sekumpulan data yang memungkinkan program memanipulasi informasi.
- Dokumen : sekumpulan dokumen yang menggambarkan operasi dan penggunaan program.

Software

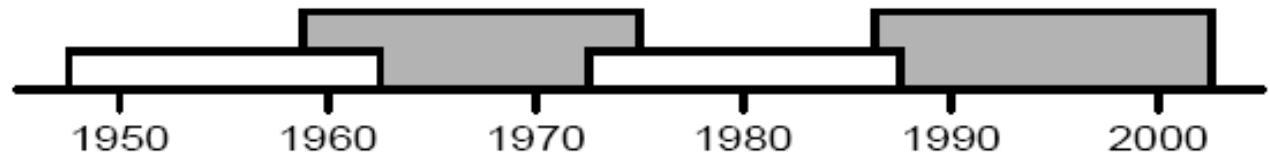


Peranan Perangkat Lunak

- **Perangkat lunak adalah sebuah produk**
 - Menyajikan potensi komputasi
 - Menghasilkan, mengelola, memperoleh, modifikasi, menampilkan atau mengirimkan informasi
- **Perangkat lunak adalah alat untuk menyajikan sebuah produk**
 - Mendukung atau menyediakan langsung fungsi sistem
 - Mengendalikan program lain (mis : operating system)
 - Mempengaruhi komunikasi (mis : networking software)
 - Membantu membangun PL lain (mis : software tools)

Perkembangan Perangkat Lunak

Evolusi software



Tahun-tahun awal :

- ◇ Batch orientation
- ◇ Limited distribution
- ◇ Customer software

Era kedua :

- ◇ Multi user
- ◇ Real time
- ◇ Database

Era ketiga

- ◇ Distributed system
- ◇ Embedded intelligence
- ◇ Low cost hardware
- ◇ Consumer infact

Era keempat :

- ◇ Expert system
- ◇ AI Machine
- ◇ Parallel architecture

Tahun-tahun Pertama (Era Pertama)

- **Batch Orientation**

Suatu orientasi di mana proses dilakukan setelah data dikumpulkan dalam satuan waktu tertentu, atau proses dilakukan setelah data terkumpul, lawan dari batch adalah Online atau Interactive Process. Keuntungan dari Interactive adalah mendapatkan data yang selalu up to date.

- **Limited distribution**

Suatu penyebaran software yang terbatas pada perusahaan-perusahaan tertentu.

- **Custom software**

Software yang dikembangkan berdasarkan keinginan perusahaan-perusahaan tertentu.

Era Kedua

□ **Multi user**

Suatu sistem di mana satu komputer digunakan oleh beberapa user pada saat yang sama.

□ **Real Time**

Suatu sistem yang dapat mengumpulkan, menganalisa dan mentransformasikan data dari berbagai sumber, mengontrol proses dan menghasilkan output dalam mili second.

□ **Database**

Perkembangan yang pesat dari alat penyimpan data yang OnLine menyebabkan muncul generasi pertama DBMS (DataBase Management System).

□ **Product Software**

Adalah software yang dikembangkan untuk dijual kepada masyarakat luas.

Era Ketiga

- **Distributed system**

Suatu sistem yang tidak hanya dipusatkan pada komputer induk (Host computer), daerah atau bidang lainnya yang juga memiliki komputer yang ukurannya lebih kecil dari komputer induk. Lawan dari distributed system adalah Centralized System.

- **Embedded Intelligence**

Suatu product yang diberi tambahan “Intelligence” dan biasanya ditambahkan microprocessor yang mutakhir. Contohnya adalah automobil, robot, peralatan diagnostic serum darah.

- **Low Cost Hardware**

Harga hardware yang semakin rendah, ini dimungkinkan karena munculnya Personal Computer.

- **Consumer Impact**

Adanya perkembangan komputer yang murah menyebabkan banyaknya software yang dikembangkan, software ini memberi dampak yang besar terhadap masyarakat.

Era Keempat

- **Expert system**

Suatu penerapan A.I. (Artificial Intelligence) pada bidang-bidang tertentu, misalnya bidang kedokteran, komunikasi, dll.

- **AI Machine**

Suatu mesin yang dapat meniru kerja dari sebagian otak manusia. Misalnya mesin robot, komputer catur.

- **Parallel Architecture**

Arsitektur komputer yang memungkinkan proses kerja LAN paralel, yang dimungkinkan adanya prosesor berbeda dalam satu komputer

Produk Perangkat Lunak

Produk perangkat lunak dibuat untuk pelanggan tertentu ataupun untuk pasar umum produk perangkat lunak ada dua jenis :

- ❑ **Generik** – dibuat untuk dijual ke suatu kumpulan pengguna yang berbeda atau pasar umum.
- ❑ **Bespoke (custom)** – dibuat untuk suatu pengguna tunggal sesuai dengan spesifikasi yang diinginkan pengguna.

Kategori Aplikasi Perangkat Lunak

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

Kategori Baru Perangkat Lunak

- ❑ Obiquitous computing — jaringan nirkabel.
- ❑ Netsourcing—Web sebagai mesin komputasi.
- ❑ Open source—kode program dibuka gratis kepada komunitas komputasi.
- ❑ Juga ...
 - ❑ Data mining
 - ❑ Grid computing
 - ❑ Cognitive machines
 - ❑ Software for nanotechnologies

Krisis Perangkat Lunak

Adalah sekumpulan masalah yang ditemukan dalam pengembangan software komputer. Masalahnya tidak hanya terbatas pada software yang tidak berfungsi sebagaimana mestinya, tetapi krisis software ini terdiri dari masalah yang berhubungan dengan :

1. Bagaimana mengembangkan software
2. Bagaimana memelihara software yang ada, yang berkembang dalam jumlah besar
3. Bagaimana mengimbangi permintaan software yang makin besar.

Masalah Perangkat Lunak

1. Estimasi jadwal dan biaya yang seringkali tidak tepat
2. Produktivitas orang-orang software (programmer) yang tidak dapat mengimbangi permintaan kebutuhan software
3. Kualitas software yang kurang baik.

Penyebab Masalah Perangkat Lunak

❑ **Karakteristik software itu sendiri**

- Software yang bersifat logika dibandingkan fisik, oleh karena itu mengukur software harus merupakan suatu kesatuan, tidak seperti hardware.
- Software bersifat tidak aus, menyebabkan kesalahan yang terjadi pada software. Umumnya terjadi pada tahap pengembangan.
- Manajer tingkat menengah dan tingkat atas yang tidak mempunyai latar belakang software, seringkali diberi tanggungjawab untuk mengembangkan software. Padahal tidak semua manajer itu dapat me-manage semua proyek. Praktisnya : software programmer atau software engineering mendapatkan latihan formal yang sedikit dalam hal teknik baru pengembangan software.

❑ **Kegagalan mereka yang bertanggungjawab dalam pengembangan software.**

Mitos Perangkat Lunak

Mitos: Jika sebuah perusahaan memutuskan untuk melakukan outsourcing suatu proyek software, perusahaan tidak perlu melakukan apapun dan cukup membiarkan perusahaan outsource untuk melakukan keseluruhan proyek.

Realitas: Apabila sebuah perusahaan tidak mengerti bagaimana cara mengatur dan mengontrol sebuah proyek software secara internal, usaha akan selau dibutuhkan meskipun outsourcing dilakukan.

Mitos: Kebutuhan software berubah secara terus-menerus, tetapi perubahan tersebut dapat dengan mudah diatasi, karena software itu fleksibel.

Realitas: Kebutuhan software memang akan terus berubah, namun dampak dari perubahan tersebut sangat bervariasi berdasarkan tahap dimana perubahan itu terjadi.

Mitos Perangkat Lunak

Mitos: Ketika program ditulis dan dapat dijalankan, pekerjaan kita selesai.

Realitas: Pada kenyataannya 60%-80% dari keseluruhan pekerjaan dilakukan setelah software diberikan kepada user untuk pertama kalinya.

Mitos: Satu-satunya produk yang dapat disampaikan dari proyek yang berhasil adalah program yang dapat dijalankan.

Realitas: Program hanya merupakan salah satu bagian dari konfigurasi software yang terdiri dari banyak elemen lainya (contoh: model, dokumentasi, perencanaan, dll).

Dll.....

Software Engineering (Rekayasa Perangkat Lunak)

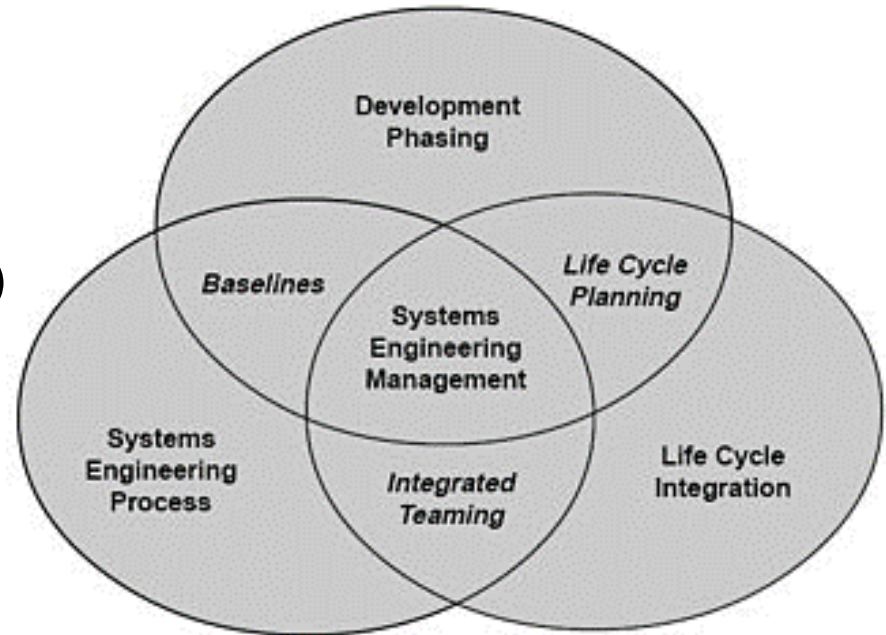
Rekayasa perangkat lunak (RPL) atau Software Engineering atau (SE) adalah satu bidang profesi yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak dan manajemen kualitas.

Software engineering adalah (*IEEE Computer Society*):

1. Penerapan suatu pendekatan yang sistematis, disiplin dan terkuantifikasi atas pengembangan, penggunaan dan pemeliharaan perangkat lunak.
2. Studi tentang pendekatan-pendekatan seperti pada (1)

System Engineering (Rekayasa Sistem)

- System engineering adalah sub disiplin rekayasa yang berkaitan dengan manajemen keseluruhan proyek rekayasa selama siklus hidup mereka (lebih fokus pada aspek fisik) yang terkait dengan logistik, koordinasi tim, kontrol mesin otomatis, proses kerja dan alat serupa .
- System engineering mempunyai kaitan dengan semua aspek pengembangan system berbasis komputer yang mencakup perangkat keras, perangkat lunak, dan yang terkait dengan proses bisnis.



Software Engineering VS System Engineering

Rekayasa sistem berkaitan dengan semua aspek dalam pembangunan sistem berbasis komputer termasuk hardware, rekayasa perangkat lunak dan proses.

Rekayasa perangkat lunak adalah bagian dari rekayasa sistem yang meliputi pembangunan pembangunan perangkat lunak, infrasktruktur, kontrol, aplikasi dan database pada sistem.

Para ahli sistem (system engineers) terlibat dalam spesifikasi sistem, desain arsitektural, integrasi dan peluncurannya.

Isu dan Tanggungjawab Profesional

❑ Confidentiality (Kerahasiaan)

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

❑ Competence (Kemampuan)

Engineer mestinya tidak salah menggambarkan tingkatan kemampuannya. Mereka mestinya tidak dengan sadar menerima pekerjaan yang diluar kemampuannya.

Isu dan Tanggungjawab Profesional

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

TUGAS MANDIRI (ke-1)

1. Carilah contoh-contoh software yang termasuk dalam kategori:

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

2. Carilah contoh-contoh framework/software/tool yang digunakan untuk mobile programming, disertai juga dengan penjelasan kegunaannya.

Tugas diemail ke msidiq2007@gmail.com

Next Software Process