

Pertemuan 1

PENGENALAN REKAYASA PERANGKAT LUNAK

Pokok Bahasan dalam RPL :

- ❖ RPL sebagai produk dan sebagai produk
- ❖ Konsep manajemen proyek
- ❖ Proses pembangunan PL dan metrik proyek
- ❖ Perencanaan proyek PL(Perangkat Lunak)
- ❖ Manajemen resiko dlm pelaksanaan proyek
- ❖ Penjadwalan dan penelusuran proyek pembangunan PL
- ❖ Jaminan kualitas PL(Perangkat Lunak)
- ❖ Manajemen konfigurasi PL
- ❖ Rekayasa sistem ke arah CB

Pokok Bahasan dalam RPL (lanjutan)

- ❖ Konsep dan prinsip analisis
- ❖ Pemodelan analisis
- ❖ Konsep dan prinsip desain
- ❖ Metode desain
- ❖ Implementasi pembangunan
- ❖ Teknik pengujian perangkat
- ❖ Strategi perancangan PL
- ❖ CASE tool pembangunan PL

Buku Referensi :

- Pressman, RS., 2008, Software Engineering: A Practitioner's Approach, New York: McGraw-Hill
- Sommerville, I, 2007, Software Engineering, Addison Wesley

Rekayasa Perangkat Lunak

- ☐ Perangkat Lunak? (Software??)
- ☐ Rekayasa Perangkat lunak-RPL?
- ☐ (Software engineering-SE??)
- ☐ Rekayasa sistem-RS? (system engineering-SyE??)
- ☐ Evolusi Perangkat Lunak
- ☐ Computer Science vs RPL
- ☐ RPL vs RS ??
- ☐ Pelaku yang berhubungan dengan Rekayasa Perangkat Lunak
- ☐ Mitos yang ada berkembang
- ☐ Tantangan dalam Pengembangan Perangkat Lunak

Definisi Perangkat Lunak (PL)

❑ IEEE-Standar Glossary of Software Engineering Terminology, 1990:

“Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.”

Maksudnya :

Perangkat lunak merupakan kumpulan dari program, prosedur, dan dokumen data lain yang saling berhubungan yang merepresentasikan masalah di dunia nyata yang dikonfigurasi dalam sebuah bentuk aplikasi yang harus dikerjakan komputer

Produk Perangkat Lunak

- Perangkat lunak <> produk perangkat keras
- Produk perangkat lunak dikembangkan (developed) atau direkayasa (engineered).
- Tidak dipabrikkan seperti Perangkat keras.
- Perangkat lunak secara pemakaian tidak pernah AUS layaknya perangkat keras

Produk Perangkat Lunak (2)

Perangkat Lunak

Produk Generik Umum

Produk Pesanan

- ❑ Perbedaan PENTING antara 2 bentuk perangkat lunak :

| Produk Generik | Produk Pesanan |
|---|---|
| Pada produk generik, organisasi yang mengembangkan perangkat lunak mengontrol spesifikasi perangkat lunak | Pada produk pesanan, spesifikasi biasanya dikembangkan dan dikontrol oleh organisasi yang membeli perangkat lunak tersebut. |
| diproduksi oleh organisasi pengembang dan dijual ke pasar terbuka → biasa disebut software | dipesan oleh pelanggan tertentu & dikembangkan khusus bagi pelanggan |

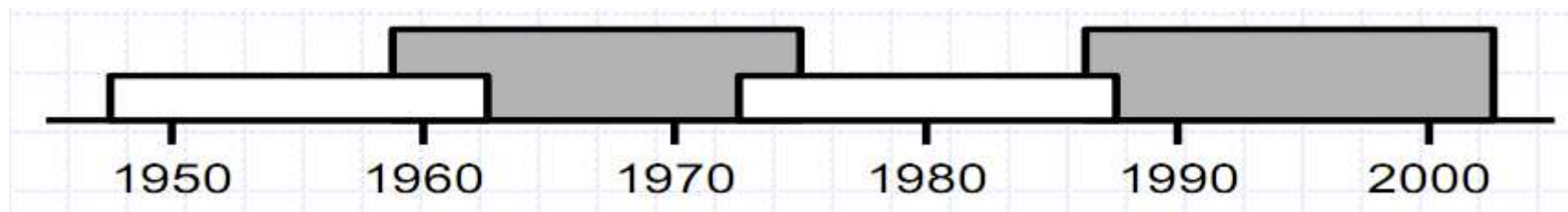
- ❑ Karakteristik perangkat lunak yang baik:
 - ❖ Usability
 - ❖ be reliable
 - ❖ maintainability
 - ❖ Efficiency
 - ❖ eye cathcing user interface
 - ❖ long life time
 - ❖ Mempunyai kinerja sesuai fungsi yang dibutuhkan pemakai

Jenis-jenis aplikasi Perangkat Lunak

- ☐ Perangkat Lunak Sistem (System software)
- ☐ Perangkat lunak waktu nyata (Realtime Software)
- ☐ Perangkat Lunak Teknik Dan Ilmu Pengetahuan (Scientific & Engineering Software)
- ☐ Embeded System (yang ditanamkan ke sistem)
- ☐ Perangkat Lunak Pengolah Data (Data Processing)
- ☐ Perangkat Lunak Sistem Informasi (Information System)
- ☐ Perangkat Lunak Sensor
- ☐ Perangkat Lunak Komunikasi (Communicaion Software)
- ☐ Perangkat Lunak Pengolah Grafis
- ☐ Perangkat Lunak Kecerdasan

Evolusi Perangkat Lunak

- ❑ Perangkat lunak pertama kali diciptakan tahun 1945
- ❑ Fokus utama pembuatannya untuk mengembangkan praktik dan teknologi dalam meningkatkan produktivitas para praktisi pengembang PL dan kualitas aplikasi yg dapat digunakan oleh pemakai
- ❑ Evolusi dipicu adanya tuntutan bisnis dan lingkungan kerja yang berkembang sangat dinamis



Evolusi Perangkat Lunak (2)

❖ Era I (1945 – 1960):

- ❑ Munculnya teknologi perangkat keras di tahap awal
- ❑ Penggunaan perangkat lunak yg berorientasi batch
- ❑ Distribusi perangkat lunak masih terbatas
- ❑ Didominasi perangkat lunak model custome
- ❑ Munculnya istilah software engineering (akhir 1950-an/awal 1960-an)
- ❑ Belum didefinisikan secara jelas tentang aspek–software engineering

Evolusi Perangkat Lunak (3)

- ❖ Era II (1960 – 1970)
 - ❑ Disebut era krisis perangkat lunak (software crisis).
 - ❑ Penggunaan perangkat lunak sudah meluas
 - ❑ Telah hadir perusahaan yang membangun software (software house)
 - ❑ Perangkat lunak sdh mengenal multiprogram, multiuser, real-time, dan penggunaan database.

Evolusi Perangkat Lunak (4)

- ❖ Era II (Lanjutan)
- ❑ Banyak project PL yg gagal
 - ✓ Over budget/anggaran
 - ✓ Berakibat rusak fisik dan kematian
 - ✓ Meledaknya Roket Ariane , kesalahan perintah dlm PL
- ❑ Dua konferensi ttg software engineering:
 - ✓ Disponsori Komite Sains NATO
 - ✓ Tahun 1968 dan 1969
 - ✓ Profesi resmi bidang software engineering

Evolusi Perangkat Lunak (5)

❖ Era III (1975 – 1985)

- ☐ Pengembangan sistem mengarah ke konsep sistem terdistribusi.
- ☐ Penerapan sistem embeded intelligence
- ☐ Harga perangkat keras sudah jauh lebih murah sehingga pemakaian meluas
- ☐ Pemanfaatan jaringan global dan lokal serta sudah diperkenalkan komunikasi digital

Evolusi Perangkat Lunak (6)

- ❖ Era IV (1985 – 2000)
 - ☐ Kemampuan PC sudah setara dengan komputer mainframe
 - ☐ Penerapan teknologi yang berorientasi pada objek
 - ☐ Implementasi sistem pakar
 - ☐ Jaringan saraf tiruan
 - ☐ Komputasi paralel
 - ☐ Jaringan komputer sudah semakin canggih

Evolusi Perangkat Lunak (7)

- ❖ Era V (2000 – sekarang)
 - ☐ Penggunaan media digital
 - ☐ Media web berkembang pesat
 - ☐ Wireless sudah meluas
 - ☐ Teknologi meluas hingga di mobile computing, mobile programming
 - ☐ Perangkat keras sudah semakin kecil namun powerfull
 - ☐ Dilakukan berbagai penelitian yang menghasilkan model proses/paradigma pengembangan PL utk mengatasi krisis PL

Krisis Perangkat Lunak

❖ Masalah yang muncul:

- Estimasi jadwal dan biaya yang seringkali tidak tepat
- Produktivitas orang-orang software yang tidak dapat mengimbangi permintaan software
- Kualitas software yang kurang baik.
- Kurangnya pengetahuan tentang: Bagaimana mengembangkan software, memelihara software yang ada.
- Bagaimana mengimbangi permintaan software yang makin besar.

Mitos Dalam Perangkat Lunak (Management)

❖ **Mitos1:**

- ☐ Kita tidak perlu mengubah pendekatan terhadap pengembangan software, karena jenis pemrograman yang kita lakukan sekarang ini sudah kita lakukan 10 tahun yang lalu.
- ☐ Realitasnya : Walau hasil program sama, produktivitas dan kualitas software harus ditingkatkan dengan menggunakan pendekatan software developments

Mitos Dalam Perangkat Lunak (Management) (2)

❖ **Mitos2:**

- ☐ Kita sudah mempunyai buku yang berisi standarisasi dan prosedur untuk pembentukan software.
- ☐ Realitasnya : Memang buku tersebut ada, tetapi apakah buku tersebut sudah dibaca atau buku tersebut sudah ketinggalan jaman (out of date).

❖ **Mitos3:**

- ☐ Jika kita tertinggal dari jadwal yang ditetapkan, kita menambah beberapa programmer saja. Konsep ini sering disebut Mongolian harde concept.

Mitos dalam perangkat lunak (Customer)

❖ Mitos1:

- ❑ Pernyataan tujuan umum sudah cukup untuk memulai penulisan program. Penjelasan yang lebih rinci akan menyusul kemudian.

- ❑ Realitasnya : Definisi awal yang buruk adalah penyebab utama kegagalan terhadap usaha-usaha pem-bentukan software. Penjelasan yang formal dan terinci tentang informasi fungsi performance interface, hambatan desain dan kriteria validasi adalah penting. Karakteristik di atas dapat ditentukan hanya setelah adanya komunikasi antara customer dan developer.

Mitos dalam perangkat lunak (Customer)

❖ Mitos2:

- ☐ Kebutuhan proyek yang terus menerus berubah dapat dengan mudah diatasi karena software itu bersifat fleksibel.

- ☐ Realitasnya : memang benar bahwa kebutuhan software berubah, tetapi dampak dari peru-bahan berbeda dari waktu ke waktu.

Mitos Dalam Perangkat Lunak (Praktisi)

❖ Mitos1:

- ☐ Tidak ada metode untuk analisis disain dan testing terhadap suatu pekerjaan, cukup menuju ke depan terminal dan mulai coding.
- ☐ Realitasnya : Metode untuk analisis desain dan testing diperlukan dalam pengembangan software.

❖ Mitos2:

- ☐ Segera setelah software digunakan, pemeliharaan dapat diminimalisasikan dan diatasi dengan cara “CATCH AS CATCH CAM”.
- ☐ Realitasnya : Diperlukan budget yang besar dalam maintenance software. Pemeliharaan software harus diorganisir, direncanakan dan dikontrol seolah-olah sebagai suatu proyek besar dalam sebuah organisasi.

Mitos dalam perangkat lunak (Management)

❖ **Mitos1:**

- ☐ Kebutuhan proyek yang terus menerus berubah dapat dengan mudah diatasi karena software itu bersifat fleksibel.
- ☐ Realitasnya : memang benar bahwa kebutuhan software berubah, tetapi dampak dari peru-bahan berbeda dari waktu ke waktu.

Definisi Rekayasa Perangkat Lunak (RPL)

RPL atau *Software Engineering (SE)* → Disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan. Perangkat Lunak yang dibuat harus mampu:

- ✓ Tepat waktu
- ✓ Tepat anggaran
- ✓ Meningkatkan kinerja
- ✓ Mengoperasikan prosedur sistem dengan benar

Perbedaan RPL dengan Computer science

| Computer Science | Software Engineering |
|---|--|
| Computer science lebih memperhatikan teori & metode komputerisasi, sedangkan software engineering menyangkut masalah praktikal pembuatan dan delivery perangkat lunak | Software engineering merupakan bagian dari system engineering, dimana sistem engineering memperhatikan semua aspek pembuatan sistem berbasis komputer termasuk perangkat keras, perangkat lunak & proses |

Perbedaan RPL dengan Rekayasa Sistem (RS)?

| Rekayasa Sistem | Rekayasa Perangkat Lunak |
|---|--|
| Rekayasa Sistem (RS) berkaitan dengan semua aspek dalam pembangunan sistem berbasis komputer termasuk hardware, rekayasa PL dan proses. | ❖ RPL adalah bagian dari rekayasa sistem yang meliputi pembangunan PL, infrasktruktur, kontrol, aplikasi dan database pada sistem. |

Tantangan dalam Rekayasa Perangkat Lunak

- ☐ Tantangan warisan
- ☐ Tantangan heterogenitas
- ☐ Tantangan pengiriman

Pelaku Dalam RPL

❖ **Manajer**

(Manajer proyek, Manajer konfigurasi, Manajer penjamin kualitas PL, Manajer bidang lainnya sesuai kebutuhan)

❖ **Software Developer**

(Analis sitem, Desainer, Programmer, Inspektor PL, Pengontrol perubahan)

❖ **Pendukung**

(Staf administrasi, Humas, Pencatat teknis, Administrator database, Administrator jaringan)