

Pengenalan Rekayasa Perangkat Lunak (RPL)

Budi Irawan



facebook.com/deerawan



[@masbugan](https://twitter.com/masbugan)



blog.budiirawan.com

History of RPL

Era 1940s

- Komputer pertama dibuat
- Bidang Computer Science mulai berkembang
- Karakteristik perangkat lunak:
 - Berukuran kecil
 - Dibuat oleh satu sampai dua orang
 - Fungsi programnya masih spesifik
 - Input dan output sederhana

Komputer Era 1940s

ENIAC

(Electronic Numerical Integrator Analyzer and Computer)



Digunakan oleh United States Army, dengan harga => **\$500,000**

Era 1960s

- Komputer mulai berkembang
- Pengembangan software dari segi ukuran dan kompleksitas meningkat, akibatnya:
 - Delivery telat
 - Tidak sesuai requirement/specs
 - Banyak error setelah digunakan

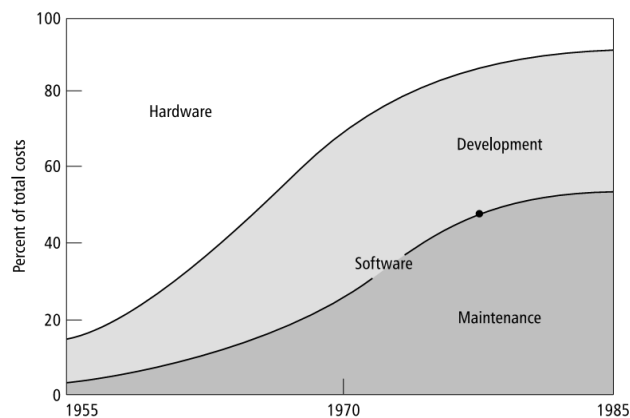
1968 dan 1969

- Dua konferensi pertama untuk software
- Diselenggarakan oleh NATO
- Istilah "Software Engineering" pertama kali muncul
- Bertujuan untuk mengatasi "Software Crisis"
- Yang dibahas "apakah software punya teori dan teknik teruji yang bisa digunakan"

Era 2013 (Saat ini)

- Komputer ada dimana-mana
- Bidang Computer Science sudah populer
- Karakteristik perangkat lunak:
 - Berukuran kecil - besar
 - Dibuat oleh 1 – 30 orang atau lebih
 - Team-nya bisa lintas benua
 - Fungsi programnya beragam cth: e-commerce, banking, akademik
 - Input dan output kompleks

Grafik Software Cost



(Source: B.W. Boehm, Software Engineering, IEEE Transactions on Computers, 1976 IEEE.)

So, what?

Biaya pengembangan software menjadi suatu isu yang harus diperhatikan



FAKTA

Perusahaan US Airline pernah rugi \$50M hanya gara2 sebuah error di sistem reservasinya

Apa itu Rekayasa Perangkat Lunak (RPL)?

Definisi RPL

Pembentukan dan penggunaan teknik serta prinsip-prinsip dalam rangka untuk memperoleh perangkat lunak yang ekonomis, handal dan bekerja secara efisien pada perangkat keras (**Konferensi NATO**)

Suatu pendekatan yang sistematis, disiplin dan terukur dalam pengembangan, pengoperasian dan pemeliharaan perangkat lunak (**IEEE Standard Glossary of Software Engineering**)

Definisi RPL Versi Vicky

Disiplin knowledge yang berharmonisasi dengan aspek pembuatan software untuk menghindari terjadinya kudeta dan konspirasi
(Vicky aka Hendrianto)

Mengapa RPL penting?

Untuk menciptakan perangkat lunak yang baik

Karakteristik Perangkat Lunak yang Baik

Karakteristik	Deskripsi
Maintainability	Perangkat lunak seharusnya dibuat sehingga mampu beradaptasi dengan perubahan dari customer
Dependability dan Security	Perangkat lunak seharusnya memiliki keamanan, kehandalan dan keselamatan
Efficiency	Perangkat lunak seharusnya menggunakan sumber daya sistem secara efisien seperti memori, prosesor, dsb serta sumber daya finansial
Acceptability	Perangkat lunak harus bisa digunakan sesuai dengan penggunaannya termasuk cocok dengan sistem yang mereka gunakan

Masalah Umum Pengembangan Perangkat Lunak

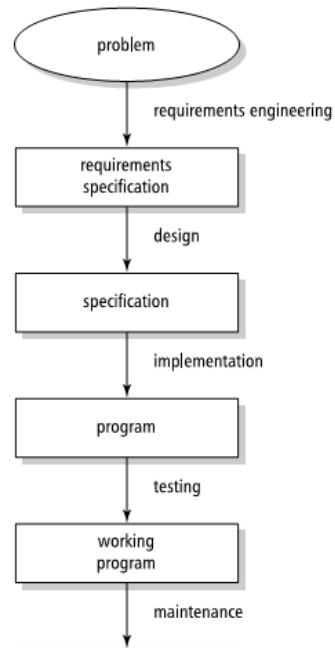
- Telat dalam delivery ke pengguna
- Kesulitan melihat progress pengembangan
- Biaya yang membengkak
- Terjadi banyak error atau bug

Fase Pengembangan Perangkat Lunak



Fase2 Basic RPL

Bisa jadi alurnya bertambah kompleks atau sederhana



1) Problem

Apa yang menjadi kendala dan masalah bagi pengguna

Which is a problem?

- Staf menggunakan excel untuk pendaftaran mahasiswa
- Staf kewalahan karena mahasiswa yang daftar terlalu banyak
- Mahasiswa tidak bisa mendapatkan informasi jadwal ujian masuk
- Mahasiswa tertarik untuk kuliah di jurusan komputer

2) Requirement

- Melihat apa yang dibutuhkan pengguna
- Melihat masalah secara lengkap
- Mendeskripsikan fungsi perangkat lunak yang ingin dikembangkan

Requirement

- Mendeskripsikan jumlah dokumentasi yang dibutuhkan
- Mendeskripsikan kebutuhan response time dan kecepatan sistem

Feasibility Study

- Fase requirement meliputi **Feasibility Study**
- **Feasibility Study** bertujuan melihat apakah ada solusi yang secara ekonomi dan teknis dapat dilakukan terhadap permasalahan yang dihadapi

Fase Requirement menghasilkan dokumen
yang disebut
Requirements Specification

3) Design

- Membuat model dari keseluruhan sistem
- Masalah dibagi2 menjadi kecil yang disebut **components**
- Fungsi components harus dispesifikasikan
- Penghubung antar components harus dispesifikasikan yang disebut dengan **interfaces**

Fase Design menghasilkan dokumen
yang disebut
Technical Specification

4) Implementation

- Implementasi mengacu pada dokumentasi desain dan requirement
- Implementasi dalam bentuk kode
- Kodenya harus well-documented, reliable, easy to read, fleksibel dan benar

Fase Implementation menghasilkan
perangkat lunak

5) Testing



Testing merupakan fase setelah implementation



Testing merupakan fase yang dilakukan dari fase requirement hingga selesai proses pengembangan

Dalam fase requirement misalnya testing
dibuat untuk memvalidasi fitur
perangkat lunak

6) Maintenance

- Memperbaiki bug atau error
- Penambahan atau perubahan requirement dari pengguna

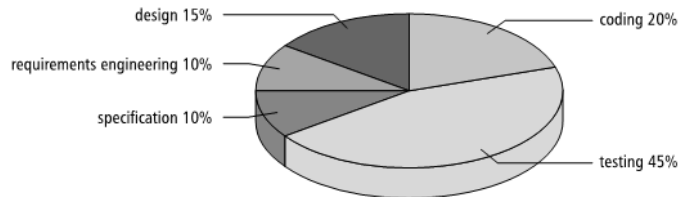
Jenis Aktivitas Maintenance

- **Corrective**
Membetulkan error/bug yang muncul
- **Adaptive**
Melakukan adaptasi perangkat lunak terhadap perubahan environment misal penggantian ke hardware baru

Jenis Aktivitas Maintenance

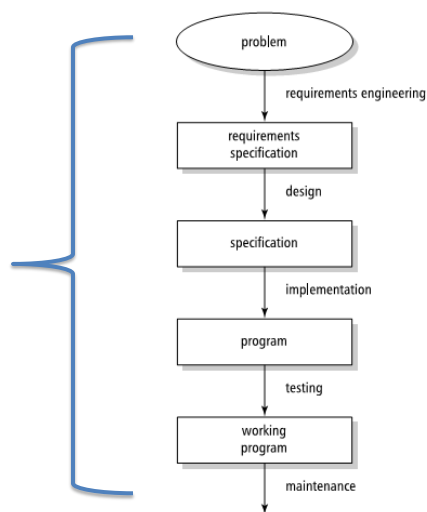
- **Perfective**
Melakukan adaptasi perangkat lunak terhadap perubahan atau penambahan requirement
- **Preventive**
Melakukan adaptasi perangkat lunak terhadap perubahan environment misal penggantian ke hardware baru

Komposisi Fase



Project

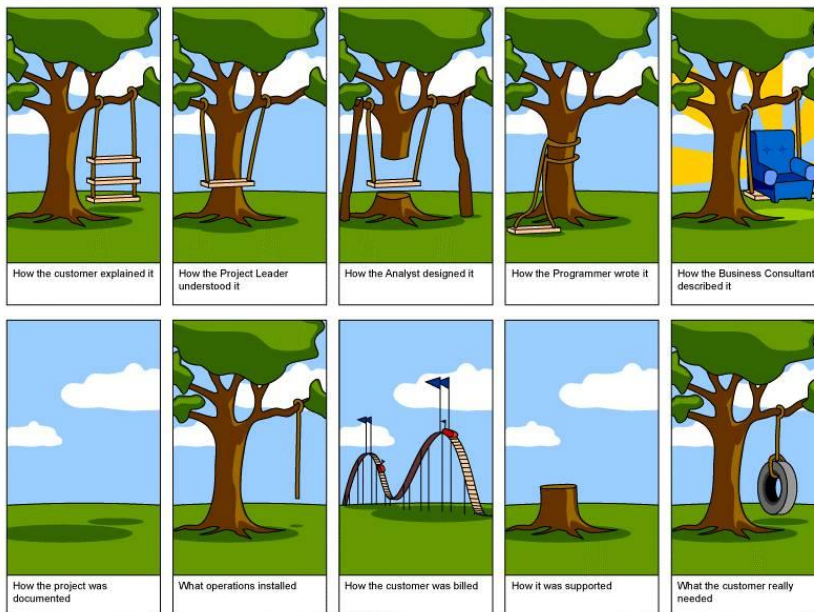
Sebuah project meliputi fase2 di samping ini



Project Management

Aktivitas untuk mengelola seluruh kegiatan tiap fase supaya menghasilkan perangkat lunak yang tepat waktu, tepat fungsi dan tepat biaya

Jangan sampai seperti ini!



Etika RPL



Profesi software engineer
memiliki kode etik yang tercantum dalam
ACM/IEEE-CS

8 Prinsip Kode Etik

1) PUBLIC

Harus konsisten dengan kepentingan publik

2) CLIENT AND EMPLOYER

Bertindak yang terbaik sesuai kepentingan klien dan atasan mereka, konsisten dengan kepentingan publik

3) PRODUCT

Memastikan produk yang dihasilkan memenuhi Standar profesionalisme yang ada

8 Prinsip Kode Etik...

4) JUDGMENT

Menjaga integritas dan kemandirian dalam setiap pembuatan keputusan

5) MANAGEMENT

Manager dan pemimpin harus terus mempromosikan pendekatan beretika dalam manajemen pengembangan dan pemeliharaan perangkat lunak

8 Prinsip Kode Etik...

6) PROFESSION

Menjaga integritas dan reputasi profesinya konsisten dengan kepentingan publik

7) COLLEAGUES

Bersikap adil dan mendukung rekan-rekan kerjanya

8) SELF

Selalu belajar terkait profesi mereka dan melakukan pendekatan beretika dalam proses pembelajarannya

Beberapa contoh etika

- Memastikan bahwa perangkat lunak telah melalui testing yang cukup
- Memastikan bahwa perangkat lunak aman dan berkualitas tinggi
- Sebagai manager tidak mengizinkan software engineer bertindak keluar dari etika profesi

What we have learned?

What we Have Learned?

- Sejarah RPL
- Apa itu RPL?
- Mengapa RPL penting?
- Masalah dalam perangkat lunak
- Fase dasar RPL
- Kode etik profesi

Your Mission

The Mission

- Membaca silabus mata kuliah
- Mencari dan mulai membaca buku referensi



Thank You