Pertemuan 1:
Perangkat Lunak dan Rekayasa Perangkat Lunak

Meuthia Rachmaniah
Departemen Ilmu Komputer, FMIPA IPB

Software Engineering: A Practitioner's Approach, 9th Ed., 2020
Roger S. Pressman dan Bruce R. Maxim

Objectives/Self-Study Outline

1) Sifat Perangkat Lunak (PL)
2) *Defining the Discipline*
3) Proses Perangkat Lunak (Software Process)
4) Praktik Rekayasa Perangkat Lunak (RPL)
5) *How It All Start*
6) Rangkuman

Quick Look

What Is It?

- PL Komputer adalah *work product* yang dibangun dan didukung oleh professional PL selama bertahun-tahun.
  - *Work product* adalah program yang dieksekusi komputer pada ukuran dan arsitektur komputer yang beragam
  - RPL mencakup proses, koleksi metode-metode (praktik), dan *array of tools* untuk profesional membangun PL berkualitas tinggi

Who Does It

- Software Engineers: membangun dan mendukung PL. Software engineers menerapkan proses RPL
- Every Users: setiap orang pada industrialized world

Why Is It Important?

- RPL penting karena memungkinkan kita membangun sistem yang kompleks secara tepat waktu dan dengan kualitas tinggi
- Memaksakan disiplin kerja dengan ketat, namun masih memungkinkan untuk menyesuaikan pendekatan yang sesuai dengan kebutuhan

What Are The Steps

PL dibangun seperti Anda membangun produk yang sukses:

- menerapkan proses yang gesit (agile) dan mudah beradaptasi
- mengarah pada hasil berkualitas tinggi
- memenuhi kebutuhan orang-orang yang akan menggunakan produk

What Is The Work Product

- Sudut pandang Software Engineers: gugus program-program, konten (data), dan work product lainnya untuk mendukung PL komputer
- Sudut pandang Users: tool atau produk yang sedemikian rupa membuat dunia users menjadi lebih baik

How Do I Ensure That I've Done It Right

- Ikuti terus kuliah dan praktikum RPL, pilih ide-ide yang dapat diterapkan pada PL yang akan dibangun, dan terapkan/gunakan PL tsb dalam pekerjaan Anda

Tantangan Realita Pengembangan PL yang Dihadapi

Aspek dan Minat

- PL telah menjadi aspek dari setiap kehidupan
- Peningkatan jumlah orang memiliki minat pada fitur dan fungsi yang terdapat pada aplikasi PL
  - *Upaya bersama (seperti konser) harus dilakukan untuk memahami masalah sebelum solusi PL dikembangkan*

IT Requirements

- Persyaratan teknologi informasi yang dituntut oleh individu, bisnis, dan pemerintah tumbuh semakin kompleks setiap tahun
- PL canggih yang pernah diimplementasukan dalam lingkungan komputasi mandiri yang dapat diprediksi, sekarang tertanam (embedded) di dalam segala hal, mulai dari elektronik konsumen hingga perangkat medis hingga kendaraan otonom
- *Design has become a pivotal activity (Desain telah menjadi aktivitas penting)*

PL DALAM SEGALA BENTUKNYA DAN DI SEMUA DOMAIN APLIKASINYA HARUS DIREKAYASA

Ketergantungan pada PL

- Individu, bisnis, dan pemerintah semakin bergantung pada PL untuk pengambilan keputusan strategis dan taktis serta operasi dan control sehari-hari
- *Software should exhibit high quality (PL harus menunjukkan kualitas tinggi)*

Pertumbuhan Minat dan Tuntutan

- Nilai yang dirasakan dari aplikasi tertentu tumbuh, sehingga basis pengguna dan umur PL juga akan tumbuh
- Seiring bertambahnya basis pengguna dan waktu penggunaan, maka tuntutan untuk adaptasi dan peningkatan juga akan bertumbuh
- *Software should be maintainable (PL harus dapat dipelihara/maintain)*

Rekayasa Perangkat Lunak (RPL)

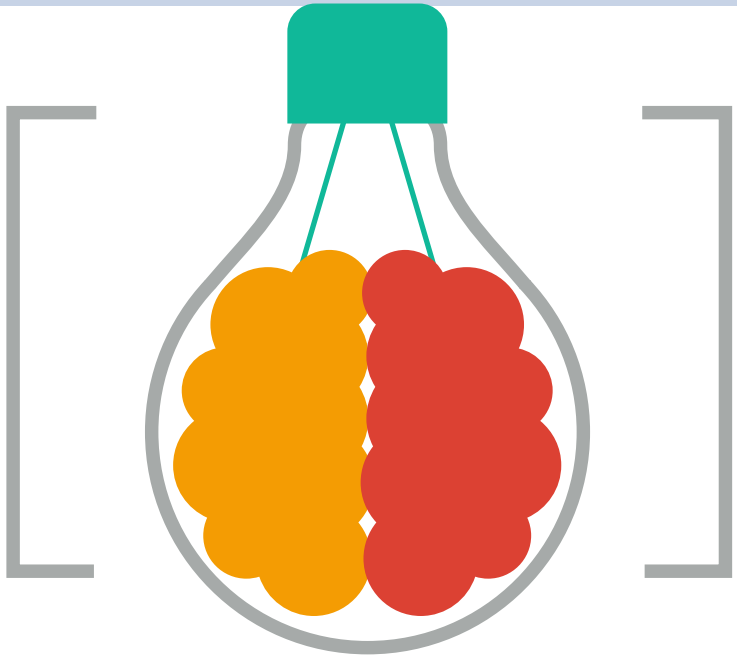
- Some realities:
  - *Upaya bersama (seperti konser) harus dilakukan untuk memahami masalah sebelum solusi PL dikembangkan*
  - *Design has become a pivotal activity (Desain telah menjadi aktivitas penting)*
  - *Software should exhibit high quality (PL harus menunjukkan kualitas tinggi*)
  - *Software should be maintainable (PL harus dapat dipelihara/maintain)*
- The seminal definition:
  - *[RPL adalah] penetapan dan penggunaan prinsip-prinsip rekayasa untuk mendapatkan PL ekonomis yang dapat diandalkan dan bekerja secara efisien pada mesin nyata (real machines)*

1) Sifat Perangkat Lunak

Dua peran PL:
1) Sebagai *Vehicle*
2) Sebagai Produk

PL sebagai *Vehicle* untuk mengirimkan produk

- Bertindak sebagai basis kontrol untuk komputer (sistem operasi)
- Mengkomunikasikan informasi (jaringan)
- Menciptakan dan mengendalikan program-program lain (*software tools and environment*)



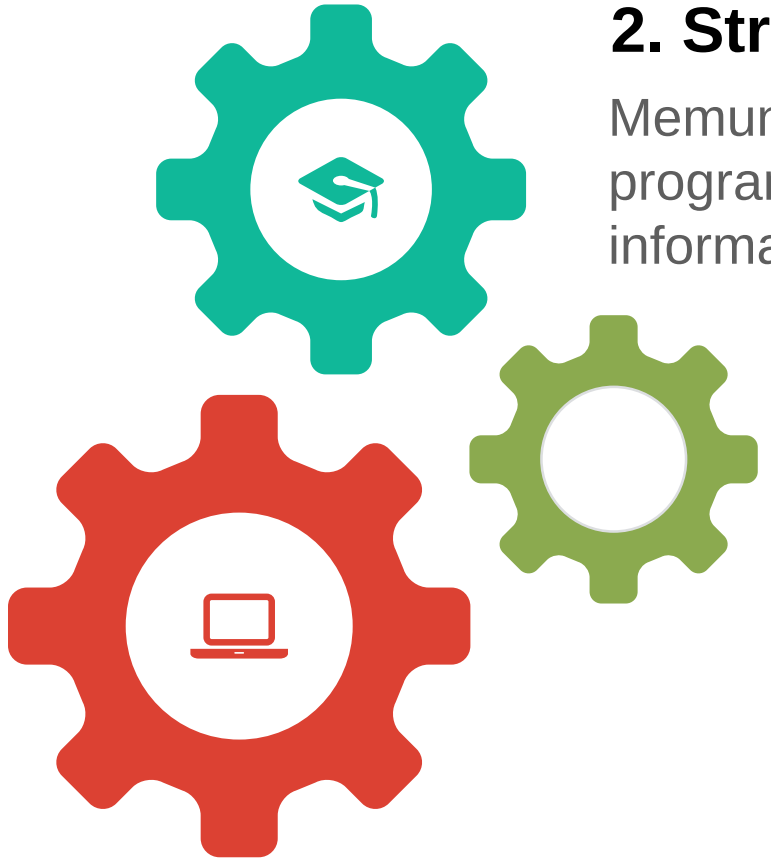
PL Komputer mengalami perubahan signifikan dalam 60 tahun terakhir:

- Kinerja hardware, arsitektur komputasi, peningkatan memory dan storage, variasi input/output
- Hasil luar biasa jika sukses, namun menimbulkan masalah besar jika gagal
- Dikembangkan oleh **tim spesialis**, bukan lagi oleh lone programmer

PL sebagai Produk

- PL memberikan potensi komputasi yang diwujudkan oleh hardware komputer atau oleh jaringan komputer yg dapat diakses oleh hardware lokal
- PL terdapat dalam mobile device, desktop, cloud, computer mainframe, autonomous machine
- PL adalah transformator informasi: menghasilkan, mengelola, memperoleh, memodifikasi, menampilkan, atau mentransmisikan informasi sederhana berupa bit tunggal atau kompleks seperti representasi augmented reality yang diperoleh dari lusinan sumber independen yang dihamparkan di dunia nyata

1.a Definisi Perangkat Lunak



1. Instruksi (Program-program Komputer)

Apabila dieksekusi menyediakan fitur-fitur, fungsi dan kinerja yang diinginkan

2. Struktur Data

Memungkinkan program-program untuk memanipulasi informasi secara memadai

Informasi Deskriptif

Dalam bentuk hardcopy dan bentuk virtual yang menjelaskan operasi dan penggunaan program-program

- Karakteristik PL berbeda dari hal-hal lain yang dibuat oleh manusia.
- PL adalah logik bukan elemen sistem fisik.
- PL memiliki satu karakteristik mendasar yang membuatnya sangat berbeda dari hardware: "***PL Tidak "Usang***".

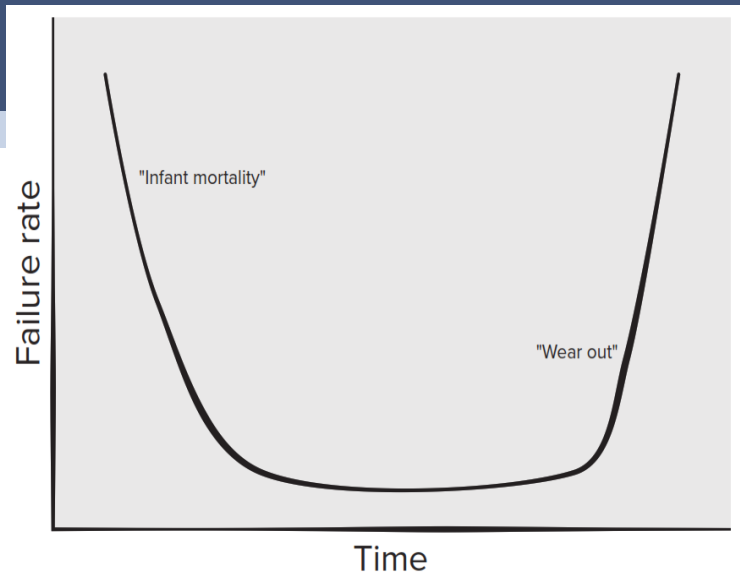


Fig.1.1 Kurva kegagalan **hardware**

- Disebut *bathtub curve relationship*
- Kegagalan relatif tinggi di awal (cacat desain/manufaktur), lalu dikoreksi, lalu cacat menurun sampai tingkat *steady-state*
- Seiring waktu, laju gagal meningkat lagi (efek kumulatif debu, vibrasi, penyalagunaan, temperature ekstrim, dll → WEAR OUT (usang)

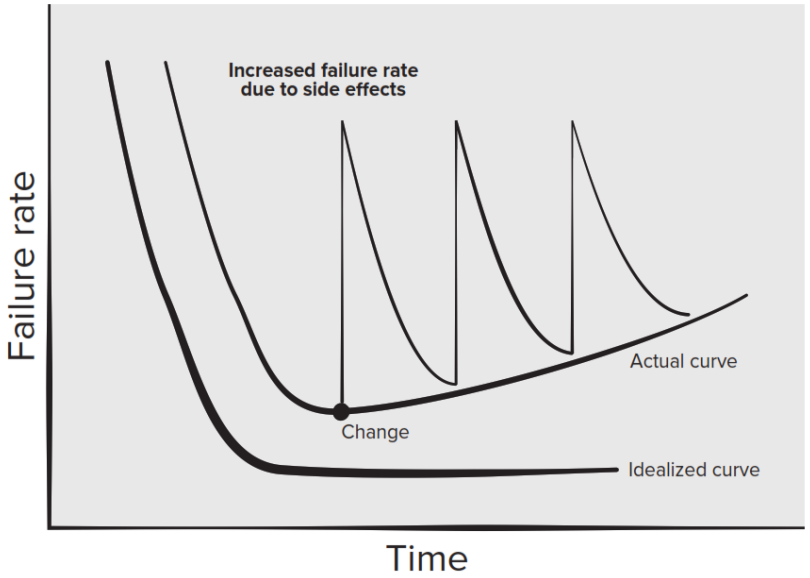


Fig.1.1 Kurva kegagalan PL (**software**)

- PL tidak rentan terhadap penyakit lingkungan yang menyebabkan hardware usang
- Secara teori, kurva tingkat kegagalan PL harus berbentuk "kurva ideal" (idealized curve)
- Awal program, cacat akan tinggi, lalu dikoreksi sehingga tingkat cacat mendatar
- PL mengalami deteriorate (menurun) sehingga perlu diubah (change), akan ada tingkat error (actual curve), dst.

1.b Domain Aplikasi Perangkat Lunak

1. System Software

Koleksi program-program sebagai layanan (service) untuk program-program lainnya:

- *Determinate*: PL sistem (misalnya, kompiler, editor, dan utilitas manajemen file) memproses kompleks pada struktur informasi tertentu
- ***Indeterminate***: aplikasi sistem (misalnya, komponen sistem operasi, driver, perangkat lunak jaringan, prosesor telekomunikasi) memproses sebagian besar data tak tentu

2. Application Software

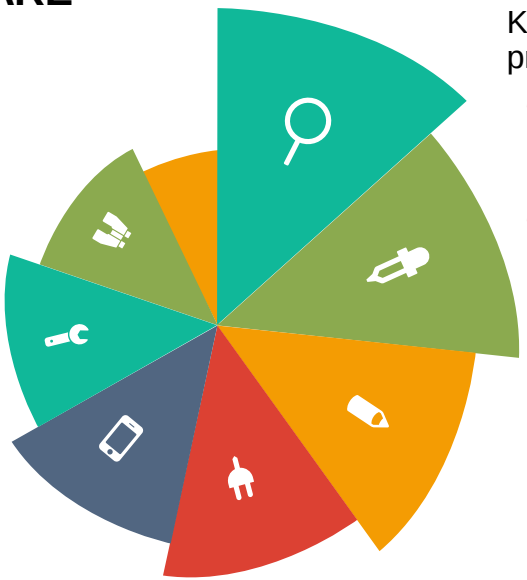
- Program yang berdiri sendiri yang memecahkan kebutuhan bisnis tertentu
- Memproses data bisnis atau teknis dengan cara yang memfasilitasi operasi bisnis atau pengambilan keputusan manajemen/teknis

3. Engineering/Scientific Software

- Serangkaian program "penghitungan angka" atau data sains yang luas seperti astronomi hingga vulkanologi, dari *stress analysis* otomotif hingga dinamika orbital, dari desain berbantuan komputer hingga kebiasaan belanja konsumen, dan dari analisis genetik hingga meteorologi

4. Embedded Software

- Terdapat di dalam produk atau sistem dan digunakan untuk mengimplementasikan dan mengontrol fitur dan fungsi untuk pengguna akhir dan untuk sistem itu sendiri
- Dapat melakukan fungsi terbatas dan esoteris (misalnya, kontrol tombol untuk oven microwave) atau menyediakan fungsi dan kemampuan kontrol yang signifikan (misalnya, fungsi digital dalam mobil seperti kontrol bahan bakar, tampilan dasbor, dan sistem pengereman)



5. Product-line Software

- Terdiri dari komponen yang dapat digunakan kembali (*reusable component*) dan dirancang untuk memberikan kemampuan khusus untuk digunakan oleh banyak pelanggan yang berbeda
- Dapat difokuskan pada pasar yang terbatas dan esoterik (misalnya, produk pengendalian persediaan/*inventory control product*) atau mencoba untuk mengatasi pasar konsumen massal.

6. Web/Mobile Application

- Berpusat pada jaringan mencakup beragam aplikasi dan aplikasi berbasis browser, komputasi awan, komputasi berbasis layanan, dan perangkat lunak yang berada di perangkat seluler

7. Artificial Intelligence Software

- Memanfaatkan heuristik untuk memecahkan problem kompleks yang tidak dapat dilakukan dengan perhitungan biasa atau analisis langsung.
- Contoh aplikasi: robotika, sistem pengambilan keputusan, pengenalan pola (gambar dan suara), pembelajaran mesin, pembuktian teorema, dan permainan.

PL – Kategori Baru

Netsourcing

Web sebagai engine komputasi

Ubiquitous Computing

Wireless Network (jaringan nirkabel)

Open Source

Source code “gratis” terbuka untuk komunitas komputasi (berkah, tetapi juga berpotensi kutukan!)

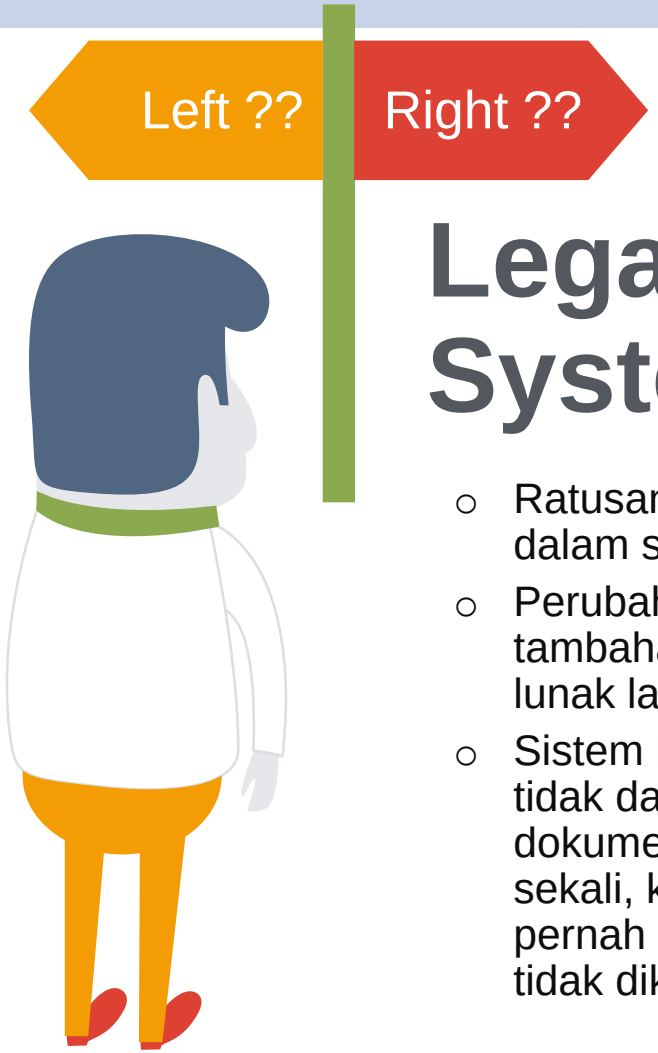
Open World Computing

Komputasi Pervasive, Terditribusi

Lainnya:

- Data Mining
- Grid Computing
- Cognitive Machines
- PL untuk teknologi Nano

1.c *Legacy* Perangkat Lunak



Legacy Software Systems ?

- Ratusan ribu program komputer termasuk dalam salah satu dari tujuh domain aplikasi PL
- Perubahan ini dapat menciptakan efek samping tambahan yang sering ada pada perangkat lunak lawas—kualitas buruk.
- Sistem lama terkadang memiliki desain yang tidak dapat diperluas, kode yang berbelit-belit, dokumentasi yang buruk atau tidak ada sama sekali, kasus pengujian dan hasil yang tidak pernah diarsipkan, dan riwayat perubahan yang tidak dikelola dengan baik

- Dikembangkan beberapa dekade yang lalu dan telah terus dimodifikasi untuk memenuhi perubahan kebutuhan bisnis dan platform komputasi.
- Proliferasi sistem semacam itu menyebabkan sakit kepala bagi organisasi besar yang menganggapnya mahal untuk dipelihara dan berisiko untuk berkembang

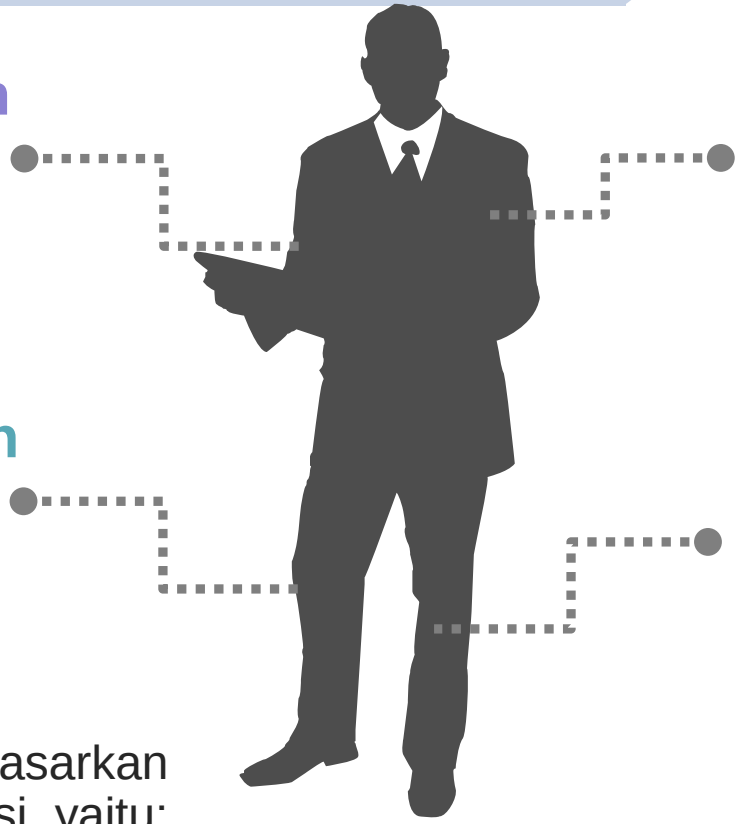
What To Do?

- ***Do Nothing*** setidaknya sampai sistem warisan harus mengalami beberapa perubahan signifikan
- ***Does not need to be fixed*** jika memenuhi kebutuhan penggunanya dan berjalan dengan andal

Alasan Legacy System harus berevolusi

Harus Disesuaikan
untuk memenuhi kebutuhan lingkungan atau teknologi komputasi baru

Harus Ditingkatkan
untuk menerapkan persyaratan bisnis baru



Harus Diperluas
untuk membuatnya bekerja dengan sistem atau database lain yang lebih modern

Harus Diarsitektur Ulang
untuk membuatnya layak dalam lingkungan komputasi yang berkembang..

Tujuan dari RPL modern:

- Merancang metodologi yang didasarkan pada gagasan evolusi, yaitu:
  - Gagasan bahwa system PL baru dapat dibangun dari yang lama
    - Semua harus berinteraksi dan bekerja sama satu sama lain

Karakteristik WebApps - I

- **Network Intensiveness**. WebApp terdapat dalam network dan harus melayani kebutuhan komunitas klien yang beragam
- **Concurrency**. Sejumlah bersar user dapat mengakses WebApp pada satu waktu bersamaan
- **Unpredictable Load**. Jumlah user pada WebApp dapat bervariasi magnitut besarnya dari hari ke hari
- **Performance**. Jika user WebApp harus menunggu lama (untuk akases, untuk pemrosesan di sisi server, untuk format dan display di sisi klien), maka WebApp akan ditinggalkan
- **Availability**. Walau ekspektasi tersedia 100% tidak mungkin diperoleh, namun user dari WebApp yang popular meminta akses "**24/7/365**"

Karakteristik WebApps - II

- **Data Driven**. Fungsi primer dari WebApp ialah penggunaan hypermedia untuk tekx, grafik, dan konten video
- **Content Sensitive**. Kualitas dan sifat estetika konten merupakan determinan penting untuk WebApp berkualitas
- **Continuous Evolution**. Aplikasi WebApp berevolusi secara konntinyu
- **Immediacy** (kebutuhan mendesak untuk meluncurkan PL ke pasar dengan cepat). WebApps sering menunjukkan waktu untuk memasarkan yang bisa dalam hitungan beberapa hari atau minggu
- **Security**. Karena WebApps tersedia melalui akses jaringan, sulit, jika bukan tidak mungkin, untuk membatasi populasi pengguna akhir yang dapat mengakses aplikasi
- **Aesthetic**. Bagian yang tak terbantahkan dari daya tarik WebApp adalah tampilan dan rasanya (***the look and feel***)

2) Defining the Discipline

Definisi RPL menurut IEEE:

1) Penerapan pendekatan yang sistematis, disiplin, terukur untuk pengembangan, pengoperasian, dan pemeliharaan PL; yaitu, penerapan rekayasa ke PL.
2) Studi tentang pendekatan seperti pada (1)

Lapisan-lapisan pada RPL

Fokus pada Kualitas

- Komitmen organisasi pada kualitas
- Total Quality Management (TQM) atau Six Sigma

Proses

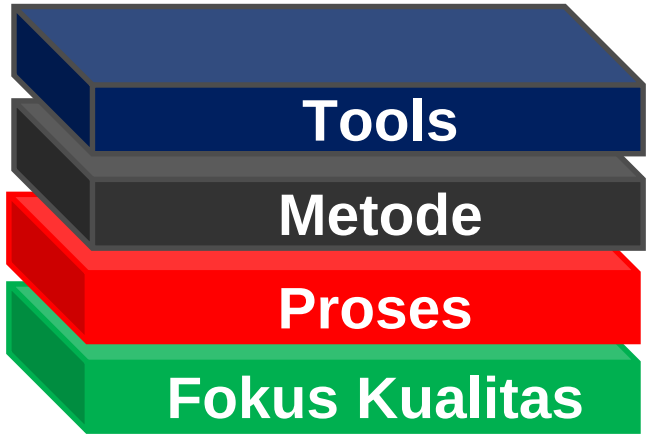
- Adalah perekat yang menyatukan lapisan teknologi dan memungkinkan pengembangan PL komputer yang rasional dan tepat waktu
- Mendefinisikan kerangka kerja (framework) yang harus ditetapkan untuk pengiriman yang efektif dari teknologi RPL
- Membentuk dasar untuk pengendalian manajemen proyek PL dan menetapkan konteks di mana metode teknis diterapkan, produk kerja/work product (model, dokumen, data, laporan, formulir, dll.) diproduksi, tonggak (benchmark) pencapaian ditetapkan, kualitas dipastikan, dan perubahan dikelola dengan baik

Metode

- Petunjuk teknis untuk membangun PL.
- Metode mencakup beragam tugas yang mencakup komunikasi, analisis persyaratan (requirements), pemodelan desain, konstruksi program, pengujian, dan dukungan.
- Metode RPLbergantung pada seperangkat prinsip dasar yang mengatur setiap bidang teknologi dan mencakup aktivitas pemodelan dan teknik deskriptif lainnya

Tools (Perangkat Bantu)

- Memberikan dukungan otomatis atau semi-otomatis untuk proses dan metode.
- Ketika tools terintegrasi sehingga informasi yang dibuat oleh satu tool dapat digunakan oleh tool lain, sebuah sistem untuk mendukung pengembangan PL , yang disebut RPL berbantuan komputer, dibuat



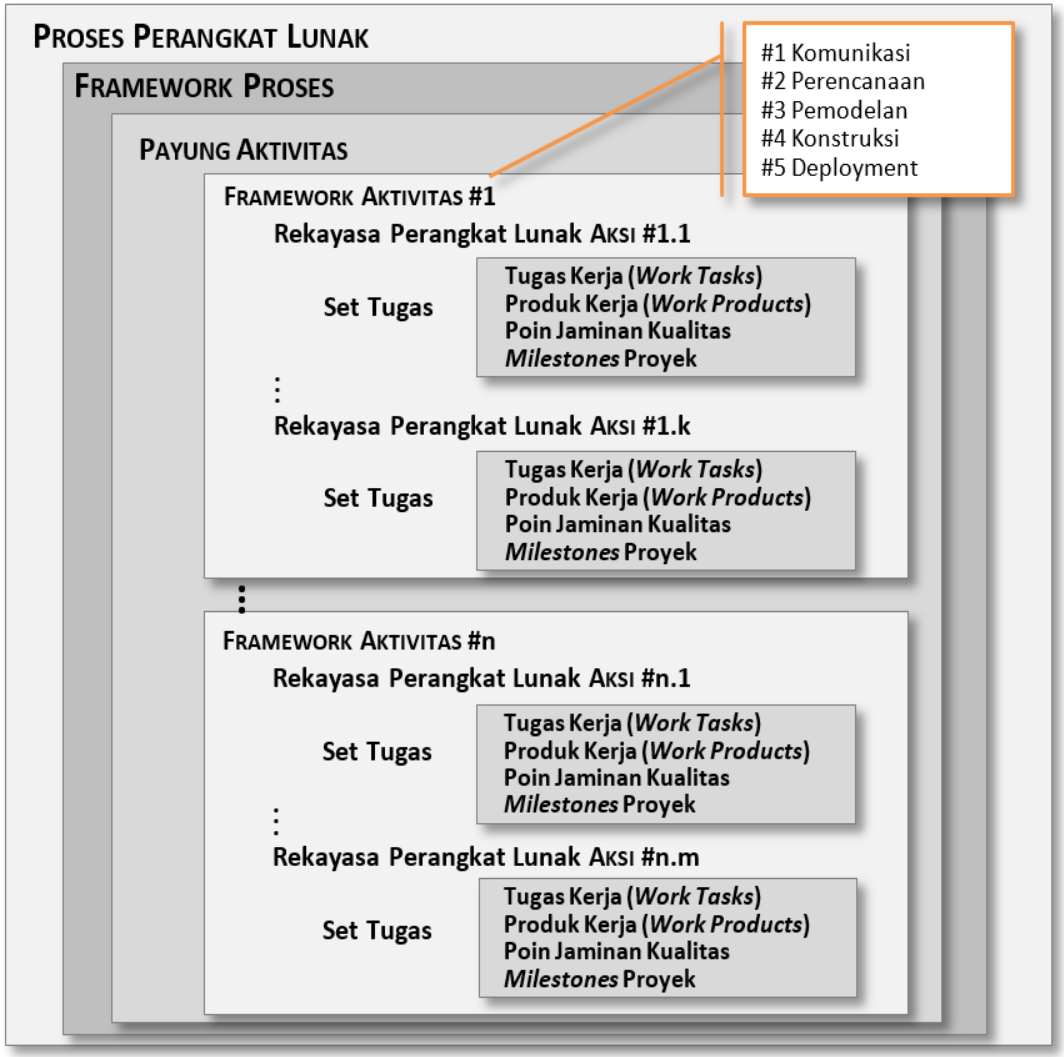
3) Proses Perangkat Lunak

- **Proses** adalah koleksi aktivitas, aksi (*actions*), dan tugas (*tasks*) yang dilakukan saat suatu produk kerja (*work product*) diciptakan
  - **Aktivitas** adalah upaya mencapai tujuan luas (mis. Komunikasi dengan stakeholders) dan menerapkannya (terlepas dari domain aplikasi, ukuran proyek, kompleksitas upaya, derajat ketelitiannya)
  - **Aksi/Actions** (mis.desain arsitektural) terdiri dari gugus tugas (*set of tasks*) yang menghasilkan produk kerja (*work product*) utama
  - **Tugas/Tasks** berfokus pada tujuan kecil, tetapi terdefinisi dengan baik (mis. melakukan tes unit) yang menghasilkan hasil yang nyata

Sebuah proses bukanlah resep kaku untuk bagaimana membangun PL komputer. Sebaliknya, ini adalah pendekatan yang dapat disesuaikan yang memungkinkan orang yang melakukan pekerjaan (tim PL) untuk memilih dan memilih rangkaian tindakan dan tugas kerja yang sesuai.

3.a Framewok Proses

- Menetapkan dasar untuk proses RPL yang lengkap dengan mengidentifikasi sejumlah kecil framework aktivitas yang berlaku untuk semua proyek PL, terlepas dari ukuran atau kompleksitasnya
- Mencakup serangkaian kegiatan payung (*Umbrella Activities*) yang berlaku di seluruh proses PL
- Framework Proses generik untuk RPL mencakup lima kegiatan: Komunikasi, Perencanaan, Pemodelan, Konstruksi, dan Deployment



Framework Proses Generik

Komunikasi

- Komunikasi dan kolaborasi dengan konsumen dan stakeholder
- Tujuan: untuk memahami tujuan stakeholder untuk proyek dan untuk mengumpulkan persyaratan yang membantu menentukan fitur dan fungsi PL

Perencanaan

- Membuat rencana proyek PL— mendefinisikan pekerjaan RPL dengan menggambarkan tugas teknis yang akan dilakukan, risiko yang mungkin terjadi, sumber daya yang akan diperlukan, produk kerja yang akan diproduksi, dan jadwal kerja

Pemodelan

- Model: Membuat "sketsa" sesuatu sehingga Anda akan memahami gambaran besarnya—seperti apa tampilannya secara arsitektur, bagaimana bagian-bagian penyusunnya cocok, dan banyak karakteristik lainnya.
- RPL: model untuk lebih memahami persyaratan PL dan desain yang akan mencapai persyaratan tersebut

Konstruksi

- Apa yang Anda desain harus dibangun.
- Aktivitas ini menggabungkan pembuatan koding (baik manual atau otomatis) dan pengujian yang diperlukan untuk mengungkap kesalahan dalam koding

Deployment

- PL (sebagai entitas yang lengkap atau sebagai tambahan yang diselesaikan sebagian) dikirimkan ke pelanggan yang mengevaluasi produk yang dikirimkan dan memberikan umpan balik berdasarkan evaluasi

- Penerapan lima framework aktivitas generik:
  - Pengembangan program sederhana dan kecil
  - Penciptaan aplikasi web
  - Rekayasa sistem berbasis komputer yang besar dan komplek

- Banyak proyek PL melakukan framework aktivitas secara iteratif seiring dengan berkembangnya proyek
  - Komunikasi, Perencanaan, Pemodelan, Konstruksi, dan Deployment diterapkan berulang kali pada sejumlah iterasi proyek
  - Setiap iterasi menghasilkan *software increment* berupa subset dari keseluruhan fitur dan fungsi PL

3.b Umbrella Activities

Aktivitas Framework Proses RPL dilengkapi oleh sejumlah Umbrella Activities

Pengukuran

Mendefinisikan dan mengumpulkan proses, proyek, dan ukuran produk yang membantu tim dalam memberikan PL yang memenuhi kebutuhan pemangku kepentingan;
dapat digunakan bersama dengan semua framework aktivitas dan umbrella acitivities lainnya

Review Teknis

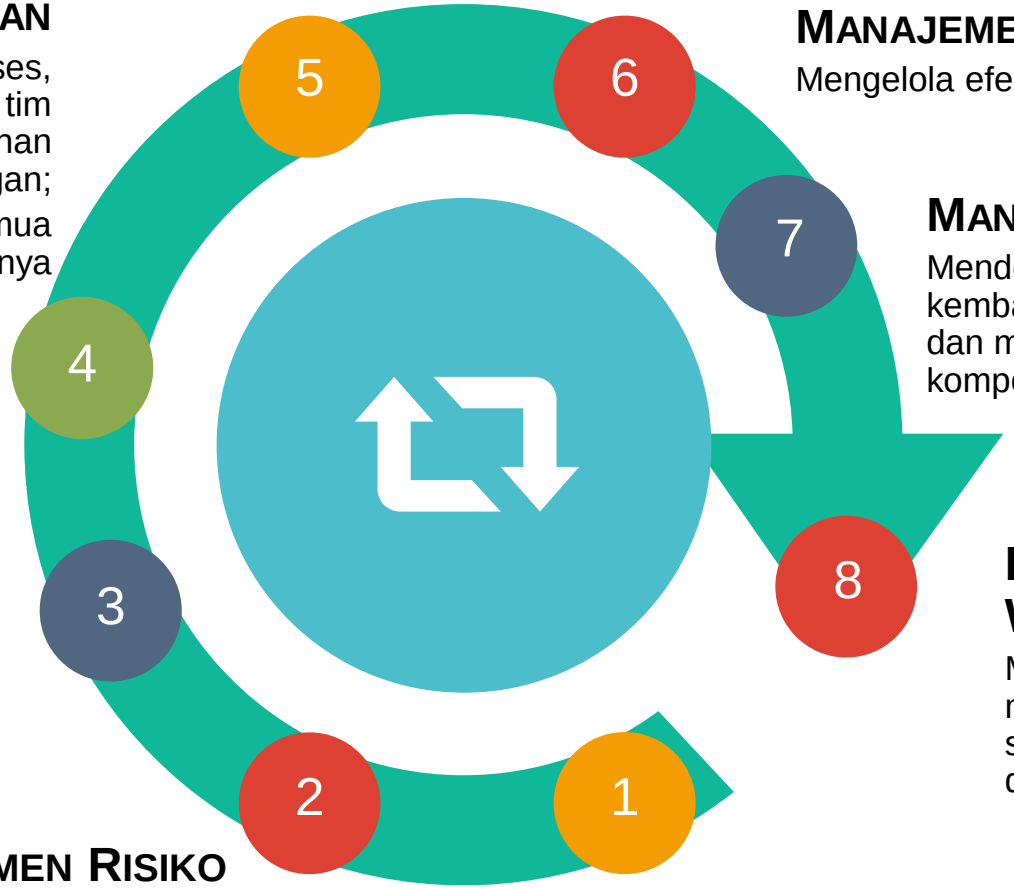
Menilai produk kerja RPL dalam upaya untuk mengungkap dan menghilangkan kesalahan sebelum disebarkan ke aktivitas berikutnya

Jaminan Kualitas PL (Software Quality Assurance)

Mendefinisikan dan melakukan aktivitas yang diperlukan untuk memastikan kualitas PL

Manajemen Risiko

Mengkaji risiko yang dapat mempengaruhi outcome proyek atau kualitas produk



Manajemen konfigurasi PL

Mengelola efek perubahan selama proses PL

Manajemen Reusability

Mendefinisikan kriteria untuk penggunaan kembali produk kerja (termasuk komponen PL) dan menetapkan mekanisme untuk mencapai komponen reusable

Preparasi dan Produksi Work Product

Meliputi aktivitas yang diperlukan untuk membuat produk kerja (*work product*) seperti model, dokumen, log, formulir, dan daftar

Pelacakan dan kontrol proyek PL

Tim PL menilai kemajuan terhadap rencana proyek dan mengambil tindakan yang diperlukan untuk mempertahankan jadwal

3.c Adaptasi Proses RPL

- Proses RPL harus agile dan adaptable (terhadap problem, proyek, tim, dan budaya organisasional)

Antar Proyek

Proses yang diadopsi untuk satu proyek mungkin berbeda secara signifikan dari proses yang diadopsi

Terdapat 9 Perbedaan

Perbedaan Proses dipengaruhi oleh:

1) Keseluruhan aliran aktivitas, aksi, dan tugas serta saling ketergantungan di antara mereka
2) Sejauh mana tindakan dan tugas didefinisikan dalam setiap Framework Activity
3) Sejauh mana work product diidentifikasi dan diperlukan
4) Cara penerapa kegiatan penjaminan mutu (quality assurance)
5) Cara aktivitas project tracking dan kendali diterapkan
6) Tingkat keseluruhan detail dan ketelitian proses yang lakukan
7) Sejauh mana pelanggan dan pemangku kepentingan lainnya terlibat dengan proyek
8) Tingkat otonomi yang diberikan kepada tim PL
9) Sejauh mana organisasi dan peran tim ditentukan

4) Praktik RPL - a) Esensi Praktik RPL

1. Understand the Problem (Komunikasi dan Analisis)

- Siapa yang memiliki kepentingan dalam pemecahan masalah? Artinya, siapa pemangku kepentingannya?
- Apa yang tidak diketahui? Data, fungsi, dan fitur apa yang diperlukan untuk menyelesaikan masalah dengan benar?
- Bisakah masalah dikotak-kotakkan? Apakah mungkin untuk merepresentasikan masalah yang lebih kecil yang mungkin lebih mudah dipahami?
- Bisakah masalah direpresentasikan secara grafis? Bisakah model analisis dibuat?

2. Rencanakan Solusi (Pemodelan dan Desain PL)

- Pernahkah Anda melihat masalah serupa sebelumnya? Apakah ada pola yang dapat dikenali dalam solusi potensial? Apakah ada PL yang mengimplementasikan data, fungsi, dan fitur yang diperlukan?
- Apakah masalah serupa sudah terpecahkan? Jika demikian, apakah elemen solusi dapat digunakan kembali?
- Bisakah submasalah didefinisikan? Jika demikian, apakah solusi mudah terlihat untuk submasalah?
- Dapatkah Anda mewakili solusi dengan cara yang mengarah pada implementasi yang efektif? Bisakah model desain dibuat?

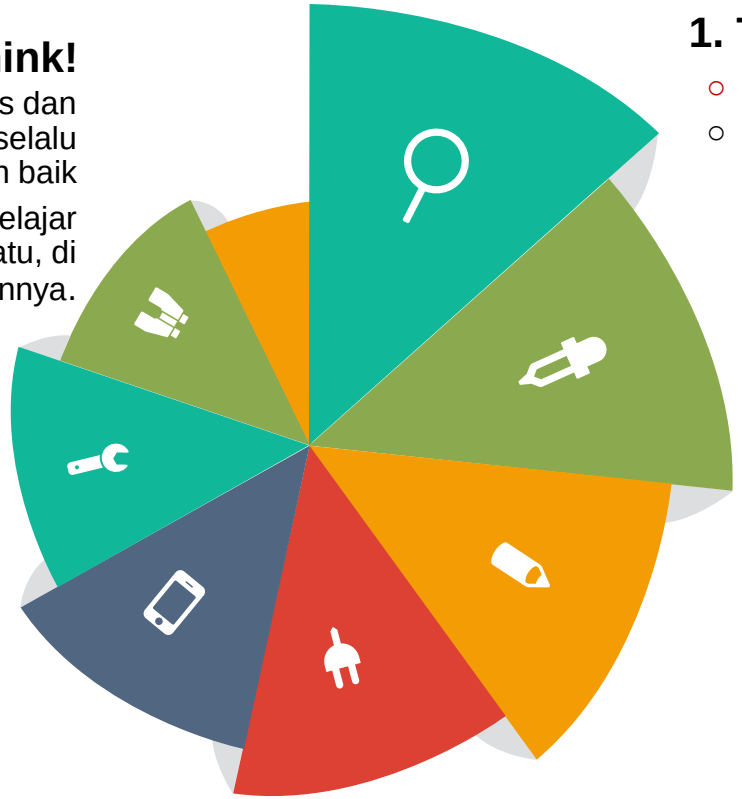
3. Jalankan Rencana (Code Generation)

- Apakah solusi sesuai dengan rencana? Apakah source code dapat dilacak ke model desain?
- Apakah setiap komponen bagian dari solusi terbukti benar? Apakah desain dan kode telah ditinjau, atau lebih baik, apakah bukti kebenaran telah diterapkan pada algoritme?

Periksa Akurasi Hasil (Testing dan QA)

- Apakah mungkin untuk menguji setiap komponen bagian dari solusi? Apakah strategi pengujian yang masuk akal telah diterapkan?
- Apakah solusi menghasilkan hasil yang sesuai dengan data, fungsi, dan fitur yang diperlukan? Apakah PL telah divalidasi terhadap semua persyaratan pemangku kepentingan?

4) PRAKTIK RPL - b) Prinsip-prinsip Umum Hooker



1. The Reason It All Exist

- To provide value to its users
- “Does this add real value to the system?” If the answer is no, don’t do it.

2. KISS (Keep It Simple, Stupid!)

- All design should be as simple as possible, but no simpler
- Desain yang lebih elegan biasanya yang lebih sederhana
- Seringkali dibutuhkan banyak pemikiran dan pengerjaan berulang kali untuk menyederhanakan desain

3. Maintain the Vision

- Visi yang jelas sangat penting untuk keberhasilan proyek PL
- Memiliki arsitek yang dapat memegang visi dan menegakkan kepatuhan membantu memastikan kesuksesan proyek PL

4. What Your Produce, Others Will Consume

- Selalu tentukan, rancang, dokumentasikan, dan implementasikan dengan mengetahui bahwa orang lain harus memahami apa yang Anda lakukan
- Potensi user sangat besar. Saat desain, pertimbangkan implementer. Saat Koding, pertimbangkan SDM yang akan memelihara dan melakukan debug

5. Be Open to the Future

- Spesifikasi berubah pada saat itu juga dan platform hardware sudah usang hanya beberapa bulan, masa pakai PL biasanya diukur dalam bulan, bukan tahun
- Selalu tanyakan “bagaimana jika”, dan persiapkan semua kemungkinan jawaban dengan menciptakan sistem yang memecahkan masalah umum, bukan hanya masalah spesifik

6. Plan Ahead for Reuse

- Reuse save time and effort. Reuse koding dan desain merupakan benefit dari teknologi object-oriented
- Perencanaan ke depan untuk r*eusable* mengurangi biaya dan meningkatkan nilai komponen *reusable* dan sistem di mana komponen-komponen tergabung

7. Think!

- Menempatkan pemikiran yang jelas dan lengkap sebelum bertindak hampir selalu menghasilkan hasil yang lebih baik
- Efek samping dari berpikir adalah belajar mengenali ketika Anda tidak tahu sesuatu, di mana Anda dapat meneliti jawabannya.

5) How It All Start

- *SafeHome:*
  - **Setiap proyek PL dipicu oleh beberapa kebutuhan bisnis —**
    - kebutuhan untuk memperbaiki cacat pada aplikasi yang ada;
    - kebutuhan akan kebutuhan untuk mengadaptasi ‘legacy system' dengan lingkungan bisnis yang berubah;
    - kebutuhan untuk memperluas fungsi dan fitur aplikasi yang ada, atau
    - kebutuhan untuk menciptakan produk, layanan, atau sistem baru.

6) RANGKUMAN

- PL adalah elemen kunci dalam evolusi sistem dan produk berbasis komputer dan salah satu teknologi terpenting di panggung dunia.
- Selama 60 tahun terakhir, PL telah berevolusi dari pemecahan masalah khusus dan alat analisis informasi menjadi industri itu sendiri.
- Namun masih terdapat kesulitan mengembangkan PL berkualitas tinggi tepat waktu dan sesuai anggaran.
- Perangkat lunak — program, data, dan informasi deskriptif — menangani beragam bidang teknologi dan aplikasi.
- PL lama terus menghadirkan tantangan khusus bagi mereka yang harus memeliharanya.
- RPL meliputi proses, metode, dan tools yang memungkinkan sistem berbasis komputer yang kompleks untuk dibangun secara tepat waktu dan berkualitas.
- Proses PL menggabungkan lima framework aktivitas — komunikasi, perencanaan, pemodelan, konstruksi, dan penyebaran (*deployment*) — yang berlaku untuk semua proyek PL.
- Praktik RPL adalah aktivitas pemecahan masalah yang mengikuti seperangkat prinsip inti.
- Saat Anda mempelajari lebih lanjut tentang RPL, Anda akan mulai memahami mengapa prinsip-prinsip ini harus dipertimbangkan saat memulai proyek PL apa pun

Chapter 1:

Perangkat Lunak dan Rekayasa Perangkat Lunak

Meuthia Rachmaniah
Departemen Ilmu Komputer, FMIPA IPB

Software Egineering: A Practitioners Approach
9th Edition, 2020

Roger S. Pressman
Bruce R. Maxim