

# **Resume Pengantar Rekayasa Perangkat Lunak**

Nama :

1. Faza Akbar H (171511009)
2. Jaka Ahmad Z (171511012)
3. Refdinal Tubagus (171511023)

## **Chapter 1**

### **1. Introduction**

#### **1.1 Professional software development**

Banyak penggunaan software saat ini di berbagai bidang seperti finansial, infrastruktur, militer, dll.

Ada 2 faktor yang membuat maraknya software gagal, yakni:

1. Bertambahnya permintaan (Permintaan terus berkembang bahkan seperti permintaan yang tak mungkin terwujud)
2. Rendahnya ekspetasi (tidak adanya penggunaan teknik dan metode software engineering)

Software engineering pertama kali diusulkan pada konferensi tahun 1968 dimana kondisi pada waktu itu disebut masa “kritis software” (Naur dan Randell 1969). Pada tahun 1970 dan 1980 muncul berbagai teknik dan metode software engineering.

Software adalah program komputer dan dokumentasi yang terasosiasi. Produk software dapat dikembangkan untuk konsumen tertentu atau untuk pasar global.

Atribut dari software yang baik adalah software yang memenuhi kebutuhan fungsional dan tampilan user yang harus maintable, dapat digunakan, dapat dipertanggung jawabkan.

Software engineering adalah disiplin ilmu yang mengacu pada seluruh aspek produksi software.

Kegiatan terpenting bagi software engineer adalah spesifikasi software, pengembangan software, evolusi software, validasi software.

Perbedaan computer sains dan software engineering:

- Computer sains focus ke teori dan pondasi

- Software engineering focus ke praktikalitas mengembangkan dan mengirim software berguna.

Perbedaan system engineering dan software engineering:

- System engineering mempelajari seluruh basis computer seperti hardware, software, proses engineering
- Software engineering focus ke prosesnya saja.

Biaya untuk software engineering 60%-nya untuk pengembangan software dan 40% untuk testing. Untuk kostum software biaya pengembangan lebih sering.

Ada 2 jenis produk software, yakni:

1. Produk general: dibuat oleh suatu organisasi dan dijual untuk pasar umum
2. Produk kostumisasi: dibuat karena permintaan konsumen.

Perbedaan penting antara produk general dan produk kostumisasi adalah:

- Produk general mengembangkan software dan software spesifikasi sendiri
- Produk kostumisasi pengembangan dan kontrol tergantung pada pihak yang memesan software tersebut

### **1.1.1 Software engineering**

Karakteristik produk ada 4, yakni:

1. Maintainabilitas : software dapat memenuhi keinginan konsumen apabila konsumen menginginkan perubahan pada software.
2. Dependabilitas dan keamanan: Jika terjadi gagal system maka tak mengakibatkan kerugian ekonomi maupun fisikal. Pengguna yang berbahaya tak merusak system dan mengaksesnya
3. Effisien: Menggunakan memory lebih kecil
4. Acceptabilitas: dapat dipahami, digunakan dan compatible dengan berbagai system yang konsumer gunakan.

Alasan mengapa software engineering begitu penting adalah karena 2 hal, yakni:

1. Lagi dan lagi, individu dan masyarakat membutuhkan system software yang lebih maju.
2. Biasanya murah

Evolusi software adalah penyesuaian software terhadap permintaan pasar dan konsumen

Validasi software adalah mengecek apa software yang dibuat sesuai dengan permintaan konsumen

Pengembangan software adalah dimana software didesain dan deprogram

Spesifikasi software adalah dimana kostumer dan pembuat bertemu untuk menentukan software seperti apa yang ia buat.

3 isu umum yang memengaruhi banyaknya perbedaan tipe software:

1. Heterogenitas : dapat dijalankan di berbagai macam system baik itu mobile maupun computer
2. Perubahan bisnis dan social: berkembangnya berbagai software sebagai bentuk beradaptasi terhadap permintaan pasar
3. Keamanan dan kepercayaan: aman dari berbagai ancaman dan kerusakan

### **1.1.2 Software engineering diversity**

Tipe – tipe aplikasi:

1. Aplikasi stand-alone : aplikasi yang berjalan pada pc local, tak perlu internet.
2. Aplikasi berbasis transaksi interactive: aplikasi yang dijalankan dan dapat diakses oleh berbagai kostumer dari pc mereka atau terminal. Contohnya seperti e-commerce
3. Sistem embedded kontrol: aplikasi yang mengatur aktifitas hardware
4. Sistem batch processing: system bisnis yang didesain untuk proses data dalam batch-batch yang besar. Contohnya seperti billing warnet, tagihan telepon.
5. Sistem Entertainment: Sistem ini digunakan untuk pribadi dan menghibur pengguna. Biasanya berupa game
6. Sistem untuk modeling dan simulasi: system ini dibuat oleh saintis dan engineer untuk memodelkan suatu situasi dan proses, yang melibatkan banyak objek
7. Sistem koleksi data: sistem yang mengoleksi data dari lingkungan sekitar dengan menggunakan sensor.
8. Sistemnya sistem: sistem yang terbuat dari beberapa system software

Berikut pondasi software engineering yang diaplikasikan di berbagai sistem software:

1. Harus dikembangkan menggunakan proses pengembangan yang teratur dan dipahami
2. Dependabilitas dan penampilan adalah hal penting untuk semua sistem
3. Memahami dan mengatur spesifikasi software dan persyaratan (apa yang harus software lakukan) adalah hal penting
4. Kamu harus membuat penggunaan seefektif mungkin dengan sumber daya yang ada

### **1.1.3 Software engineering and web**

produk-produk sistem ini, pengembangan browser web yang bisa

menjalankan program kecil dan melakukan beberapa pemrosesan lokal menyebabkan evolusi dalam bisnis dan

perangkat lunak organisasi. Alih-alih menulis perangkat lunak dan menerapkannya di PC pengguna,

perangkat lunak itu digunakan pada server web. Ini membuatnya lebih murah untuk berubah dan meningkatkan peranti lunak, karena tidak perlu menginstal peranti lunak pada setiap PC. Saya t

juga mengurangi biaya, karena pengembangan antarmuka pengguna sangat mahal.

Akibatnya, di mana pun dimungkinkan untuk melakukannya, banyak bisnis telah pindah interaksi berbasis web dengan sistem perangkat lunak perusahaan.

Tahap berikutnya dalam pengembangan sistem berbasis web adalah gagasan tentang web

jasa. Layanan web adalah komponen perangkat lunak yang memberikan fungsionalitas tertentu yang bermanfaat

dan yang diakses melalui Web. Aplikasi dibangun dengan mengintegrasikan

layanan web ini, yang mungkin disediakan oleh perusahaan yang berbeda. Pada prinsipnya, ini penautan dapat bersifat dinamis sehingga aplikasi dapat menggunakan layanan web yang berbeda setiap kali

bahwa itu dieksekusi.

Perubahan radikal dari organisasi software, merintis sistem berbasis web menjadi terengineer. Contohnya:

1. Penggunaan kembali software menjadi dominan untuk sistem berbasis web.
2. Sekarang umumnya diakui bahwa itu tak bisa dipraktikan untuk menentukan semua persyaratan untuk sistem lanjutan
3. User interfaces terikat dengan kapabilitas dari web browser

## 1.2 Software engineering ethics

Sifat/sikap yang dimiliki oleh seorang software engineer adalah sebagai berikut:

1. **Konfidensialitas:** kamu harus menghormati konfidensialitas pegawai atau klien.
2. **Kompeten**
3. **Hak properti intellectual:** Kamu harus berhati-hati atas hak property intellectual yang dilindungi oleh hukum sebagai hak paten.
4. **Jangan menyalahgunakan computer**

Rekayasa Perangkat Lunak Kode Etik dan Praktik Profesional

ACM / IEEE-CS Joint Task Force pada Etika Rekayasa Perangkat Lunak dan Praktik Profesional  
PEMBUKAAN

Versi singkat dari kode ini merangkum aspirasi pada tingkat abstraksi yang tinggi; klausula-klausula itu

termasuk dalam versi lengkap memberikan contoh dan rincian tentang bagaimana aspirasi ini mengubah cara kita bertindak

profesional rekayasa perangkat lunak. Tanpa aspirasi, detailnya bisa menjadi legalistik dan membosankan;

tanpa perincian, aspirasi bisa menjadi tinggi tetapi kosong; bersama, aspirasi dan perincian membentuk kode kohesif.

Insinyur perangkat lunak harus berkomitmen untuk membuat analisis, spesifikasi, desain, pengembangan,

pengujian dan pemeliharaan perangkat lunak profesi yang bermanfaat dan dihormati. Sesuai dengan mereka

komitmen terhadap kesehatan, keselamatan dan kesejahteraan masyarakat, insinyur perangkat lunak harus mematuhi hal-hal berikut

Delapan Prinsip:

1. **PUBLIK** - Insinyur perangkat lunak harus bertindak konsisten dengan kepentingan publik.
2. **KLIEN DAN EMPLOYER** - Insinyur perangkat lunak harus bertindak dengan cara yang ada di dalam kepentingan terbaik dari klien dan majikan mereka yang konsisten dengan kepentingan publik.
3. **PRODUK** - Insinyur perangkat lunak harus memastikan bahwa produk mereka dan terkait modifikasi memenuhi standar profesional setinggi mungkin.
4. **JUDGMENT** - Insinyur perangkat lunak harus menjaga integritas dan kemandirian di dalam mereka penilaian profesional.
5. **MANAJEMEN** - Manajer dan pemimpin rekayasa perangkat lunak harus berlangganan dan mempromosikan pendekatan etika untuk manajemen pengembangan perangkat lunak dan pemeliharaan.
6. **PROFESI** - Insinyur perangkat lunak harus meningkatkan integritas dan reputasi

Profesi konsisten dengan kepentingan publik.

7. KOLEGA - Insinyur perangkat lunak harus adil dan mendukung mereka rekan kerja.

8. DIRI - Insinyur perangkat lunak harus berpartisipasi dalam pembelajaran sepanjang hayat terkait dengan mempraktekkan profesi mereka dan akan mempromosikan pendekatan etika terhadap mempraktekkan profesi.

Etika lain dari software engineering adalah menjauhi segala macam pengembangan software yang sangat berbahaya seperti membantu membuat senjata nuklir (sistemnya) atau membuat senjata lainnya.

## **Chapter 2**

Proses perangkat lunak adalah serangkaian aktivitas yang mengarah pada produksi perangkat lunak. Aktivitas ini bisa merupakan pengembangan perangkat lunak dari awal atau pengembangan perangkat lunak yang sudah ada yaitu dengan memperluas dan memodifikasi system yang sudah ada.

Empat kegiatan dasar yang penting untuk perangkat lunak :

1. Spesifikasi mendefinisikan apa yang harus dilakukan sistem.
2. Desain dan implementasi - mendefinisikan organisasi sistem dan mengimplementasikan sistem.
3. Validasi - memeriksa apakah itu melakukan apa yang diinginkan pelanggan.
4. Evolusi - mengubah sistem sebagai tanggapan terhadap perubahan kebutuhan pelanggan.

Hal-hal yang termasuk ke deskripsi proses :

1. Produk merupakan hasil dari sebuah proses
2. Peran merupakan tanggung jawab orang-orang yang terlibat dalam proses tersebut.
3. Pra-dan pasca-kondisi, yang merupakan pernyataan yang benar sebelum dan sesudah aktivitas proses telah diberlakukan atau produk yang dihasilkan.

### **2.1 Model Proses Perangkat Lunak**

Model perangkat lunak merupakan representasi yang disederhanakan dari proses perangkat lunak.

Ada beberapa model proses perangkat lunak, yaitu sebagai berikut :

1. The Waterfall Model

Proses perangkat lunak pada model ini adalah mengidentifikasi kebutuhan, merancang , implementasi, menguji dan melakukan perawatan perangkat lunak. Pada model ini

dokumentasi dibuat di setiap fase, hal ini membuat proses pemantauan terhadap kemajuan pengembangan lebih mudah. Tetapi kekurangan model ini adalah tidak fleksibel karena sulit untuk menangani perubahan persyaratan dari pelanggan. Kesimpulan dari model Waterfall adalah hanya boleh digunakan ketika persyaratan dari pelanggan dipahami baik oleh pengembang dan persyaratan tersebut tidak berubah selama pengembangan system.

## 2. Incremental development

Proses perangkat lunak pada model ini adalah identifikasi kebutuhan, pengembangan dan validasi. Pada model ini lebih mengembangkan system yaitu menambah fungsi baru terhadap versi sebelumnya, pada model ini konsumen dapat mengevaluasi sistem pada tahap yang relatif awal dalam pengembangan untuk memberikan apa yang dibutuhkan. Model ini mempunyai kelebihan dibanding dengan model Waterfall, yaitu jumlah analisis dan dokumentasi yang harus dikerjakan ulang jauh lebih sedikit dibanding model Waterfall, lebih mudah mendapatkan feedback dari konsumen tentang pengembangan yang telah dilakukan, konsumen dapat menggunakan perangkat lunak lebih awal dibanding model Waterfall. Model ini juga mempunyai kekurangan, yaitu prosesnya tidak terlihat dan struktur system cenderung menurun karena terjadi penambahan.

## 3. Reuse-oriented software engineering

Pada model ini pengembangan perangkat lunak dengan cara menggunakan kembali system yang sudah ada yang desainnya mirip dengan apa yang dibutuhkan, pengembang mencari system yang bisa digunakan kembali dan memodifikasinya sesuai kebutuhan system yang dibuat. Proses pengembangan sistem berfokus pada mengintegrasikan komponen-komponen ini ke dalam sistem daripada mengembangkannya mereka dari awal. Reuse-oriented software engineering memiliki keuntungan nyata mengurangi jumlah perangkat lunak yang akan dikembangkan sehingga mengurangi biaya dan risiko.

## 2.2 Aktivitas Proses

### 2.2.1 Software Specification

Software Specification adalah proses pemahaman dan menentukan layanan apa yang diperlukan dari system dan mengidentifikasi kendala operasi dan pengembangan system.

Ada empat kegiatan utama dalam proses Software Specification, yaitu :

1. Studi Kelayakan, studi ini mempertimbangkan apakah system yang diusulkan akan hemat biaya dari sudut pandang bisnis dan jika dikembangkan dalam batasan anggaran yang ada.
2. Persyaratan elisitasi dan analisis Ini adalah proses menurunkan sistem persyaratan melalui pengamatan sistem yang ada. Ini mungkin melibatkan pengembangan satu atau lebih model sistem dan prototipe. Ini membantu Anda memahami sistem yang akan ditentukan.
3. Spesifikasi kebutuhan Spesifikasi kebutuhan adalah aktivitas menerjemahkan informasi yang dikumpulkan selama kegiatan analisis menjadi dokumen.
4. Persyaratan validasi adalah kegiatan memeriksa persyaratan untuk realisme, konsistensi, dan kelengkapannya. Jika terdapat kesalahan maka akan diperbaiki.

### **2.2.2 Software Design and Implementation**

Ada empat kegiatan yang menjadi bagian proses desain untuk system informasi :

1. Desain arsitektur, mengidentifikasi struktur keseluruhan sistem, yang komponen utama, hubungan mereka, dan bagaimana mereka didistribusikan.
2. Desain antarmuka, menentukan antarmuka antar komponen sistem.
3. Desain komponen, yaitu mengambil setiap komponen system dan merancang bagaimana komponen tersebut dapat beroperasi.
4. Perancangan database, yaitu merancang struktur data system.

### **2.2.3 Software validation**

1. Development testing adalah system diuji oleh orang yang mengembangkan system.
2. System testing System components adalah menemukan kesalahan yang dihasilkan antara komponen dan masalah antarmuka komponen.
3. Acceptance testing adalah Sistem diuji dengan data yang disediakan oleh pelanggan sistem daripada dengan data uji simulasi.

### **2.2.4 Software Evolution**

Fleksibilitas sistem perangkat lunak adalah salah satu alasan utama mengapa semakin banyak perangkat lunak sedang dimasukkan dalam sistem yang besar dan kompleks. Orang-orang berpikir pengembangan perangkat lunak sebagai aktivitas kreatif di mana sistem perangkat lunak dikembangkan dari konsep awal hingga sistem kerja. Namun, mereka terkadang pikirkan pemeliharaan perangkat lunak sebagai membosankan dan tidak menarik.



## **2.3 Mengatasi perubahan**

Perubahan merupakan hal yang tak dapat dihindari dalam suatu proyek pembuatan perangkat lunak, dan tentunya mempengaruhi data, struktur, desain, dan aspek-aspek program lainnya sehingga diperlukan nya pengerjaan ulang yang tentunya memerlukan biaya tambahan, sehingga dapat mengakomodasi suatu perubahan pada perangkat lunak merupakan aspek yang sangat penting. Ada 2 pendekatan penting yang dapat digunakan untuk mengurangi biaya pengerjaan ulang:

1. Menghindari Perubahan
2. Mentoleransi Perubahan

terdapat juga 2 cara untuk mengatasi perubahan dan penggantian kebutuhan :

1. prototype sistem, ketika sebuah versi sistem atau bagian sistem telah selesai dibuat, segeralah mengkonfirmasi kebutuhan pelanggan
2. pengiriman peningkatan, mengirimkan peningkatan sistem ke pelanggan untuk diuji coba dan dimintai pendapat

### **2.3.1 Prototipe**

Prototipe adalah versi awal pada suatu sistem perangkat lunak yang digunakan untuk mendemonstrasikan konsep, dengan pembuatan prototipe yang jelas, dapat membantu mengatasi perubahan karena pelanggan dapat membayangkan secara lebih jelas sistem dari perangkat lunak yang akan dibuat ,prototipe dapat digunakan pada sistem, desain, dan testing, masalah umum pada prototipe, ialah penggunaan nya belum tentu dapat sesuai dengan hasil akhir

### **2.3.2 Pengiriman tambahan**

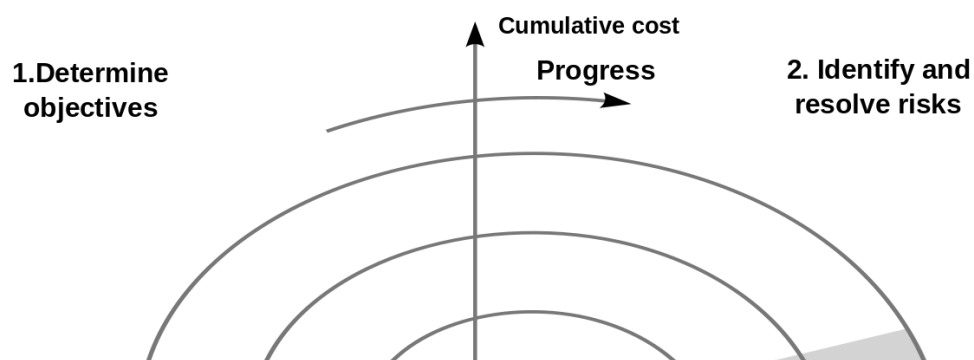
Kebutuhan user sangat diprioritaskan, dan kebutuhan prioritas tertinggi akan dimasukkan pada penambahan awal, ketika pembuatan penambahan dimulai , kebutuhan dihentikan sementara sekalipun kebutuhan untuk penambahan kemudian dapat lanjut dikembangkan

pembuatan sistem tambahan dilakukan dengan cara mengevaluasi satu persatu tambahan sebelum melanjutkan ke pengembangan tambahan selanjutnya. Pendekatan normal dapat menggunakan metode agile, pengiriman tambahan diuji cobakan ke pengguna khusus,

terdapat pula pantangan dalam pengiriman tambahan yaitu prioritas tertinggi pada sistem layanan cenderung sering menerima pengujian, beresiko rendah terjadinya kegagalan proyek

### **2.3.3 Model Spiral Boehm's**

Model ini diusulkan oleh Boehm's pada tahun 1988 , model ini proses perangkat lunak direpresentasikan dalam bentuk spiral





Setiap putaran mewakili fase dari proses perangkat lunak, dari yang terdalam seperti Requirement analisis, kemudian desain, perancangan atau coding, dan seterusnya, setiap putaran dibagi menjadi empat sektor :

1. Objective setting
2. Risk assesment
3. Development and validation
4. Planning

### **2.3.4 Rational Unified Process (RUP)**

RatIoan Unified Process atau bisa disebut proses kesatuan rasional merupakan contoh dari proses model secara modern yang diturunkan dari lembaga asosiasi Unified Software Development Process, RUP biasanya digambarkan dengan tiga perspektif :

1. Perspektif Dinamis
2. Perspektif Statis
3. Perspektif Praktik

Fase – fase RUP :

1. Inception
2. Elaboration
3. Construction
4. Transition

RUP menjelaskan praktek rekayasa perangkat lunak yang baik yang direkomendasikan untuk digunakan dalam pengembangan sistem, berikut enam fundamental yang baik untuk dipraktikan :

1. Kembangkan perangkat lunak secara interaktif
2. Kelola kebutuhan secara eksplisit
3. Gunakan komponen berbasis arsitektur
4. Gunakan model grafis UML untur perangkat lunak model visual
5. Verifikasi kualitas perangkat lunak
6. Kontrol perubahan pada perangkat lunak