



**MODUL REKAYASA PERANGKAT LUNAK  
( CCR 210 )**

**MODUL 1  
PENGERTIAN REKAYASA PERANGKAT  
LUNAK**

**DISUSUN OLEH  
HANI DEWI ARIESSANTI, M.KOM**

**UNIVERSITAS ESA UNGGUL  
2019**

## **PENGERTIAN REKAYASA PERANGKAT LUNAK (SOFTWARE ENGINEERING)**

### **A. Pendahuluan**

Rekayasa perangkat lunak telah berkembang sejak pertama kali diciptakan pada tahun 1940-an hingga kini. Focus utama pengembangannya adalah untuk mengembangkan praktek dan teknologi untuk meningkatkan produktivitas para praktisi pengembang perangkat lunak dan kualitas aplikasi yang dapat digunakan oleh pemakai.

Istilah software engineering digunakan pertama kali pada akhir 1950-an dan awal 1960-an. Saat itu, masih terdapat perdebatan tajam mengenai aspek engineering dari pengembangan perangkat lunak. Pada tahun 1968 dan 1969, komite sains NATO mensponsori dua konferensi tentang rekayasa perangkat lunak, yang memberikan dampak kuat terhadap pengembangan rekayasa perangkat lunak. Banyak yang menganggap dua konferensi inilah yang menandai awal resmi profesi rekayasa perangkat lunak.

Modul Rekayasa Perangkat Lunak merupakan materi fundamental di Informatika. Interaksi Manusia dan Komputer merupakan mata kuliah yang mengajarkan mahasiswa tentang Definisi perangkat lunak, rekayasa perangkat lunak, jenis model proses dalam rekayasa perangkat lunak, rekayasa sistem, analisis dan disain dengan menggunakan pendekatan terstruktur beserta alat bantu pemodelannya (Data Flow Diagram, Data Dictionary, Process Specification, Structure Chart ), Dokumentasi hasil Analisis dan Disain, strategi dan teknik pengujian perangkat lunak, Jaminan Kualitas Perangkat lunak dan Manajemen Proyek Perangkat Lunak .

### **B. Kompetensi Dasar**

1. Mengetahui, memahami dan menguasai perakitan komputer dan algoritma dalam model proses Software Engineering

### **C. Kemampuan Akhir Yang Diharapkan**

- 1) Mahasiswa mampu menjeleaskan definisi perangkat lunak
- 2) Mahasiswa mampu menjeleaskan peran perangkat lunak
- 3) Mahasiswa mampu menjeleaskan Mitos dalam rekayasa perangkat lunak
- 4) Mahasiswa mampu menjelaskan What is Software

## D. Kegiatan Belajar

### Pengertian RPL

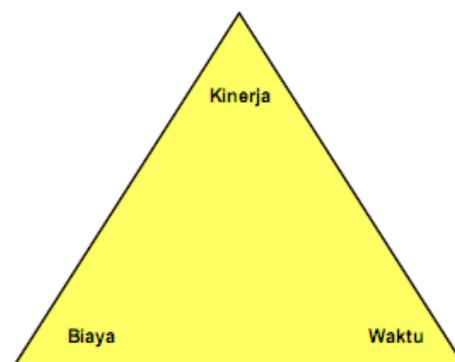
Istilah Reakayasa Perangkat Lunak (RPL) secara umum disepakati sebagai terjemahan dari istilah Software engineering. Istilah Software Engineering mulai dipopulerkan pada tahun 1968 pada software engineering Conference yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (software) dan program komputer.

Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (O'Brien, 1999).

Rekayasa Perangkat Lunak sendiri adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dari pengertian ini jelaslah bahwa RPL tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan "semua aspek produksi" pada pengertian di atas, mempunyai arti semua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari RPL.

### Tujuan Rekayasa Perangkat Lunak

Secara umum tujuan RPL tidak berbeda dengan bidang rekayasa yang lain. Hal ini dapat kita lihat pada Gambar di bawah ini.



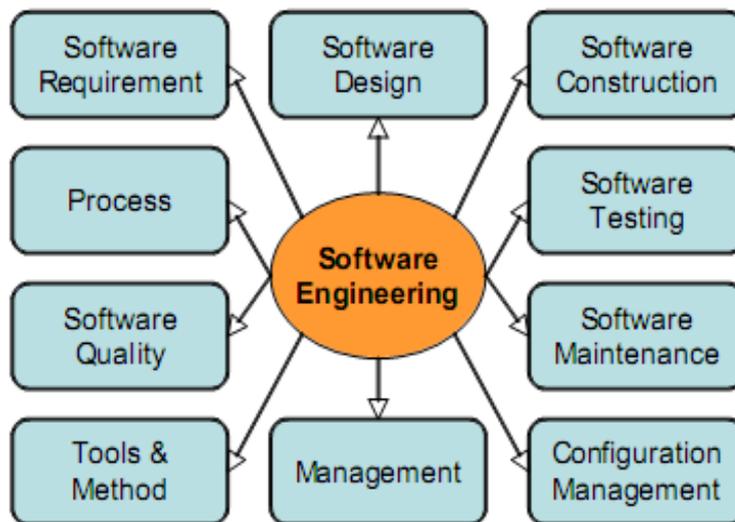
Gambar 1. Tujuan RPL

Dari Gambar di atas dapat diartikan bahwa bidang rekayasa akan selalu berusaha menghasilkan output yang kinerjanya tinggi, biaya rendah dan waktu penyelesaian yang tepat. Secara lebih khusus kita dapat menyatakan tujuan RPL adalah:

- a. Memperoleh biaya produksi perangkat lunak yang rendah
- b. Menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepat waktu
- c. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform
- d. Menghasilkan perangkat lunak yang biaya perawatannya rendah

### Ruang Lingkup

Sesuai dengan definisi yang telah disampaikan sebelumnya, maka ruang lingkup RPL dapat digambarkan sebagai berikut



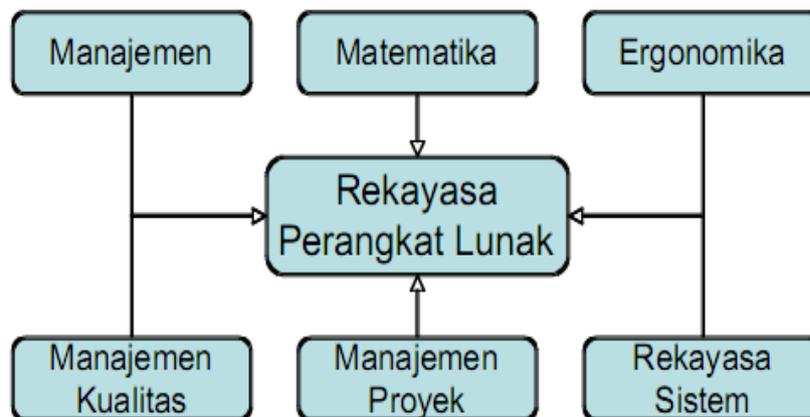
Gambar 2. Ruang lingkup RPL (Abran et.al., 2004).

- Software Requirements berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak
- Software Desain mencakup proses penampilan arsitektur, komponen, antar muka, dan karakteristik lain dari perangkat lunak
- Software Construction berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan
- Software Testing meliputi pengujian pada keseluruhan perilaku perangkat lunak
- Software Maintenance mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan

- Software Configuration Management berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu
- Software Engineering management berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak
- Software Engineering Tools And Methods mencakup kajian teoritis tentang alat bantu dan metode RPL
- Software Engineering Process berhubungan dengan definisi, implementasi pengukuran, pengelolaan, perubahan dan perbaikan proses RPL
- Software Quality menitik beratkan pada kualitas dan daur hidup perangkat lunak

### Rekayasa Perangkat Lunak Dan Disiplin Ilmu Lain

Cakupan ruang lingkup yang cukup luas, membuat RPL sangat terkait dengan disiplin dengan bidang ilmu lain. tidak saja sub bidang dalam disiplin ilmu komputer namun dengan beberapa disiplin ilmu lain diluar ilmu komputer. Hubungan keterkaitan RPL dengan ilmu lain dapat dilihat pada gambar dibawah ini



Gambar 3. Keterkaitan RPL dengan bidang ilmu lain.

- bidang ilmu manajemen meliputi akuntansi, finansial, pemasaran, manajemen operasi, ekonomi, analisis kuantitatif, manajemen sumber daya manusia, kebijakan, dan strategi bisnis
- bidang ilmu matematika meliputi aljabar linier, kalkulus, peluang, statistik, analisis numerik, dan matematika diskrit
- bidang ilmu manajemen proyek meliputi semua hal yang berkaitan dengan proyek, seperti ruang lingkup proyek, anggaran, tenaga kerja, kualitas,

manajemen resiko dan keandalan, perbaikan kualitas, dan metode-metode kuantitatif

- bidang ilmu ergonomika menyangkut hubungan (interaksi) antar manusia dengan komponen-komponen lain dalam sistem komputer
- bidang ilmu rekayasa sistem meliputi teori sistem, analisis biaya-keuntungan, pemodelan, simulasi, proses, dan operasi bisnis

### Perkembangan Rekayasa Perangkat Lunak

Meskipun baru dicetuskan pada tahun 1968, namun RPL telah memiliki sejarah yang cukup yang panjang. Dari sisi disiplin ilmu, RPL masih relatif muda dan akan terus berkembang. Arah perkembangan yang saat ini sedang dikembangkan antara lain meliputi :

Tahun	Kejadian
1940an	<ul style="list-style-type: none"> <li>• Komputer pertama yang membolehkan pengguna menulis kode program langsung</li> </ul>
1950an	<ul style="list-style-type: none"> <li>• Generasi awal interpreter dan bahasa macro</li> <li>• Generasi pertama compiler</li> </ul>
1960an	<ul style="list-style-type: none"> <li>• Generasi kedua compiler</li> <li>• Komputer mainframe mulai dikomersialkan</li> <li>• Pengembangan perangkat lunak pesanan</li> <li>• Konsep Software Engineering mulai digunakan</li> </ul>
1970an	<ul style="list-style-type: none"> <li>• Perangkat pengembang perangkat lunak</li> <li>• Perangkat minicomputer komersial</li> </ul>
1980an	<ul style="list-style-type: none"> <li>• Perangkat Komputer Personal (PC) komersial</li> <li>• Peningkatan permintaan perangkat lunak</li> </ul>
1990an	<ul style="list-style-type: none"> <li>• Pemrograman berorientasi obyek (OOP)</li> <li>• Agile Process (sekelompok aktifitas pembangunan perangkat lunak secara iteratif yang menekankan pada aktifitas konstruksi (desain dan koding))</li> <li>• Extreme Programming</li> <li>• Peningkatan drastis kapasitas memori</li> <li>• Peningkatan penggunaan internet</li> </ul>
2000an	<ul style="list-style-type: none"> <li>• Platform interpreter modern (Java, .Net, PHP, dll)</li> <li>• Outsourcing</li> </ul>

## Metode Rekayasa Perangkat Lunak

Pada rekayasa perangkat lunak, banyak model yang telah dikembangkan untuk membantu proses pengembangan perangkat lunak. Model-model ini pada umumnya mengacu pada model proses pengembangan sistem yang disebut **System Development Life Cycle (SDLC)** seperti terlihat pada Gambar berikut ini.



Gambar 4. System Development Life Cycle (SDLC).

1. Kebutuhan terhadap definisi masalah yang jelas.  
Input utama dari setiap model pengembangan perangkat lunak adalah pendefinisian masalah yang jelas. Semakin jelas akan semakin baik karena akan memudahkan dalam penyelesaian masalah..
2. Tahapan-tahapan pengembangan yang teratur.  
Meskipun model-model pengembangan perangkat lunak memiliki pola yang berbeda-beda, biasanya model-model tersebut mengikuti pola umum analysis – design – coding – testing – maintenance
3. Stakeholder berperan sangat penting dalam keseluruhan tahapan pengembangan.  
Stakeholder dalam rekayasa perangkat lunak dapat berupa pengguna, pemilik, pengembang, pemrogram dan orang-orang yang terlibat dalam rekayasa perangkat lunak tersebut.
4. Dokumentasi merupakan bagian penting dari pengembangan perangkat lunak.  
Masing-masing tahapan dalam model biasanya menghasilkan sejumlah tulisan, diagram, gambar atau bentuk-bentuk lain yang harus didokumentasi dan merupakan bagian tak terpisahkan dari perangkat lunak yang dihasilkan.

5. Keluaran dari proses pengembangan perangkat lunak harus bernilai ekonomis. Nilai dari sebuah perangkat lunak sebenarnya agak susah di-rupiah-kan. Namun efek dari penggunaan perangkat lunak yang telah dikembangkan haruslah memberi nilai tambah bagi organisasi. Hal ini dapat berupa penurunan biaya operasi, efisiensi penggunaan sumberdaya, peningkatan keuntungan organisasi, peningkatan “image” organisasi dan lain-lain.

### **Tahapan Rekayasa Perangkat Lunak**

Meskipun dalam pendekatan berbeda-beda, namun model-model pendekatan memiliki kesamaan, yaitu menggunakan pola tahapan analysis – design – coding construction) – testing – maintenance.

#### **1. Analisis sistem**

adalah sebuah teknik pemecahan masalah yang menguraikan sebuah sistem menjadi komponen-komponennya dengan tujuan mempelajari seberapa bagus komponen-komponen tersebut bekerja dan berinteraksi untuk meraih tujuan mereka.

Analisis mungkin adalah bagian terpenting dari proses rekayasa perangkat lunak. Karena semua proses lanjutan akan sangat bergantung pada baik tidaknya hasil analisis. Ada satu bagian penting yang biasanya dilakukan dalam tahapan analisis yaitu pemodelan proses bisnis.

#### **2. Model proses**

adalah model yang memfokuskan pada seluruh proses di dalam sistem yang mentransformasikan data menjadi informasi (Harris, 2003). Model proses juga menunjukkan aliran data yang masuk dan keluar pada suatu proses. Biasanya model ini digambarkan dalam bentuk Diagram Arus Data (Data Flow Diagram / DFD). DFD menyajikan gambaran apa yang manusia, proses dan prosedur lakukan untuk mentransformasi data menjadi informasi.

#### **3. Disain perangkat lunak**

adalah tugas, tahapan atau aktivitas yang difokuskan pada spesifikasi detil dari solusi berbasis computer (Whitten et al, 2004).

Disain perangkat lunak sering juga disebut sebagai physical design. Jika tahapan analisis sistem menekankan pada masalah bisnis (business rule), maka sebaliknya disain perangkat lunak fokus pada sisi teknis dan implementasi sebuah perangkat lunak (Whitten et al, 2004).

Output utama dari tahapan disain perangkat lunak adalah spesifikasi disain. Spesifikasi ini meliputi spesifikasi disain umum yang akan disampaikan kepada stakeholder sistem dan spesifikasi disain rinci yang akan digunakan pada tahap implementasi. Spesifikasi disain umum hanya berisi gambaran umum agar stakeholder sistem mengerti akan seperti apa perangkat lunak yang akan dibangun. Biasanya diagram USD tentang perangkat lunak yang baru merupakan point penting dibagian ini. Spesifikasi disain rinci atau kadang disebut disain arsitektur rinci perangkat lunak diperlukan untuk merancang sistem sehingga memiliki konstruksi yang baik, proses pengolahan data yang tepat dan akurat, bernilai, memiliki aspek user friendly dan memiliki dasar-dasar untuk pengembangan selanjutnya.

Desain arsitektur ini terdiri dari desain database, desain proses, desain user interface yang mencakup desain input, output form dan report, desain hardware, software dan jaringan. Desain proses merupakan kelanjutan dari pemodelan proses yang dilakukan pada tahapan analisis.

#### 4. **Konstruksi**

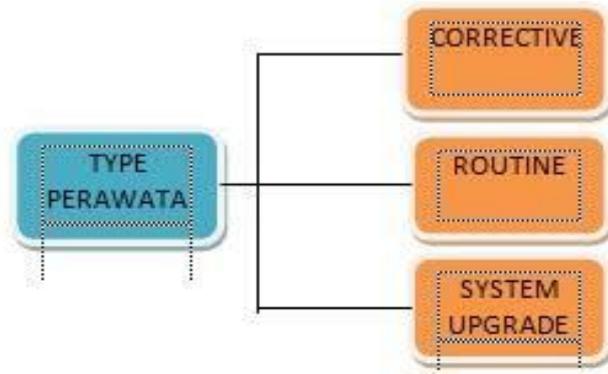
adalah tahapan menerjemahkan hasil disain logis dan fisik ke dalam kode-kode program komputer.

#### 5. **Pengujian sistem**

melibatkan semua kelompok pengguna yang telah direncanakan pada tahap sebelumnya. Pengujian tingkat penerimaan terhadap perangkat lunak akan berakhir ketika dirasa semua kelompok pengguna menyatakan bisa menerima perangkat lunak tersebut berdasarkan kriteria-kriteria yang telah ditetapkan.

#### 6. **Perawatan dan Konfigurasi.**

Ketika sebuah perangkat lunak telah dianggap layak untuk dijalankan, maka tahapan baru menjadi muncul yaitu perawatan perangkat lunak. Ada beberapa tipe perawatan yang biasa dikenal dalam dunia perangkat lunak seperti terlihat pada diagram di Gambar di bawah ini :



Gambar 5. Tipe-tipe perawatan.

- Perawatan corrective dilakukan jika terjadi kesalahan atau biasa dikenal sebagai bugs. Perawatan bisa dilakukan dengan memperbaiki kode program, menambah bagian yang dirasa perlu atau malah menghilangkan bagian-bagian tertentu.
- Perawatan routine biasa juga disebut preventive maintenance dilakukan secara rutin untuk melihat kinerja perangkat lunak ada atau tidak ada kesalahan.
- Tipe perawatan sistem upgrade dilakukan jika ada perubahan dari komponen-komponen yang terlibat dalam perangkat lunak tersebut. Sebagai contoh perubahan platform sistem operasi dari versi lama ke versi baru menyebabkan perangkat lunak harus diupgrade.

#### E. Latihan

- 1) Apakah pengertian dari rekayasa perangkat lunak?
- 2) Sebutkan contoh perspektif proses pada perangkat lunak?
- 3) Sebutkan model proses generic pada perangkat lunak? jelaskan?
- 4) Apakah arti dari suatu proses model perangkat lunak?
- 5) Sebutkan fase fase dan Masalah –masalah yang timbul pada Model air terjun?
- 6) Sebutkan stage stage pengujian perangkat lunak dan jelaskan artinya?
- 7) Sebutkan lima langkah pokok dalam menganalisis kebutuhan dalam merancang RPL?
- 8) Sebutkan produk kerja dari Perangkat lunak ?
- 9) Sebutkan karakteristik dari perangkat lunak?
- 10) Sebutkan tujuan dari RPL?

## F. Rangkuman

- 1) RPL adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu Analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.
- 2) RPL merupakan sub bidang ilmu komputer yang dalam penerapannya membutuhkan dukungan baik dari sub bidang ilmu komputer lainnya maupun bidang-bidang ilmu lain.. Tujuan RPL adalah menghasilkan perangkat lunak dengan kinerja tinggi, tepat waktu, berbiaya rendah, dan multiplatform.

## G. Tes Formatif

- 1) 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 disebut ....
    - a. Perangkat Lunak
    - b. Manajemen
    - c. Perangkat Keras
    - d. Sistem
    - e. Database
  - 2) Yang tidak termasuk ke dalam hal kemampuan dari Perangkat Lunak yang dibuat yaitu:
    - a. Tepat waktu
    - b. Mengoperasikan prosedur sistem dengan benar
    - c. Meningkatkan kinerja
    - d. Tepat guna
    - e. Tepat anggaran
  - 3) Perakayasa membuat suatu alat bekerja. Menerapkan teori, metode, dan alat bantu yang sesuai, selain itu mereka menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, disebut ...
    - a. Disiplin rekayasa
    - b. Mengoperasikan prosedur sistem dengan benar
    - c. Meningkatkan kinerja
    - d. Tepat anggaran
    - e. Tepat Waktu
  - 4) RPL tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak tetapi juga dengan kegiatan seperti Manajemen proyek PL dan pengembangan alat bantu, metode, dan teori untuk mendukung produksi Perangkat Lunak. Hal ini disebut
    - a. Tepat anggaran
    - b. Semua aspek produksi perangkat lunak
    - c. Disiplin Rekayasa
    - d. Mengoperasikan prosedur sistem dgn benar
    - e. Meningkatkan kinerja
-

- 5) Rekayasa yang berkaitan dengan semua aspek dalam pembangunan sistem berbasis komputer termasuk hardware, rekayasa Perangkat Lunak dan proses disebut ....
- a. Rekayasa Sistem
  - b. Rekayasa Komputer
  - c. Disiplin Rekayasa
  - d. Rekayasa perangkat Lunak
  - e. Rekayasa Database

#### **H. Daftar Pustaka**

Roger S. Pressman. Software Engineering, 6th edition. 2005

Ian Sommerville. Software Engineering, 6th edition. 2001

Software Engineering Body of Knowledge (SWEBOK). 2004

Computing and Information Science. Software Engineering Slides. Cornell University. 2009

<http://www.csbdu.in/econtent/Software%20Engineering/>

Edward Yourdon, Modern Structured Analysis, 1st edition, 1988

Kendall, System Analysis and Design, 8th edition, 2013