

MODUL MATA KULIAH

REKAYASA PERANGKAT LUNAK 1

KP342 - 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR
JAKARTA**

TIM PENYUSUN

Noni Juliasari, M.Kom
Bima Cahya Putra, M.Kom
Basuki Hari Prasetyo, M.Kom

VERSION 1.0



PERTEMUAN 1

PENDAHULUAN

Capaian Pembelajaran	:	<ol style="list-style-type: none">1. Mahasiswa mampu memahami sistem perkuliahan dan 9 Nilai Kebudiluhuran2. Mahasiswa mengetahui perangkat lunak dan perkembangannya
Sub Pokok Bahasan	:	<ol style="list-style-type: none">1.1. Penjelasan ruang lingkup perkuliahan (RPS)1.2. Penjelasan kontrak perkuliahan dan tata tertib1.3. Penjelasan 9 Nilai Kebudiluhuran1.4. Pengantar Rekayasa Perangkat Lunak dan Cakupannya
Daftar Pustaka	:	<ol style="list-style-type: none">1. Kung, David C., 2014. Object Oriented Software Engineering: An Agile Methodology, McGraw-Hill2. Pressman, Roger S. 2010. Software Engineering : A Practitioner's Approach, 7th, McGraw-Hill3. Schach, Stephen R. 2010. Object Oriented and Classical Software Engineering, 8th, McGraw-Hill4. Sommerville, Ian. 2010. Software Engineering, 9th, Pearson Education

		5. Software Engineering Body of Knowledge (SWEBOK). 2004
--	--	--

PENDAHULUAN

1.1. Pengantar

Dalam dunia teknologi informasi, kadang kita mendengar istilah tentang *software* atau perangkat lunak komputer. Dikatakan bahwa, keberadaan *hardware* atau perangkat keras komputer tanpa ditunjang oleh perangkat lunak maka komputer tidak dapat digunakan, begitupun sebaliknya. Nama lain dari perangkat lunak komputer adalah *software*. Istilah "*Software*" pertama kali dicetuskan dalam versi cetak oleh John W. Tukey melalui tulisannya berjudul *The Teaching of Concrete Mathematics* di jurnal *The American Mathematical Monthly*, vol. 65, No. 1 bulan Januari 1958 hlm. 1-9. Selanjutnya istilah ini sering digunakan sebagai *software* aplikasi. Dimana perangkat lunak (*software*) ini kemudian menjelma menjadi bagian penting dalam sisi kehidupan manusia terutama dalam kaitan pemrosesan dengan bantuan mesin (sumber daya) komputer.

1.2. Sejarah Rekayasa Perangkat Lunak

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

1945 - 1965: Awal

Istilah *software engineering* digunakan pertama kali pada akhir 1950-an dan awal 1960-an. Saat itu, masih terdapat debat 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 perkembangan rekayasa perangkat lunak. Banyak yang menganggap bahwa dua konferensi inilah yang menandai awal resmi profesi rekayasa perangkat lunak.

1965 - 1985: krisis perangkat lunak

Pada tahun 1960-an hingga 1980-an, banyak masalah yang ditemukan para praktisi

pengembangan perangkat lunak. Banyak proyek yang gagal, hingga masa ini disebut sebagai krisis perangkat lunak. Kasus kegagalan pengembangan perangkat lunak terjadi mulai dari proyek yang melebihi anggaran, hingga kasus yang mengakibatkan kerusakan fisik dan kematian. Salah satu kasus yang terkenal antara lain meledaknya roket Ariane akibat kegagalan perangkat lunak.

1985 - kini: tidak ada senjata pamungkas

Selama bertahun-tahun, para peneliti memfokuskan usahanya untuk menemukan teknik jitu untuk memecahkan masalah krisis perangkat lunak. Berbagai teknik, metode, alat, proses diciptakan dan diklaim sebagai senjata pamungkas untuk memecahkan kasus ini. Mulai dari pemrograman terstruktur, pemrograman berorientasi object, perangkat pembantu pengembangan perangkat lunak (CASE tools), berbagai standar, UML hingga metode formal diagung-agungkan sebagai senjata pamungkas untuk menghasilkan *software* yang benar, sesuai anggaran dan tepat waktu.

Pada tahun 1987, Fred Brooks menulis artikel No Silver Bullet, yang berproposisi bahwa tidak ada satu teknologi atau praktek yang sanggup mencapai 10 kali lipat perbaikan dalam produktivitas pengembangan perangkat lunak dalam tempo 10 tahun. Sebagian berpendapat, no silver bullet berarti profesi rekayasa perangkat lunak dianggap telah gagal. Namun sebagian yang lain justru beranggapan, hal ini menandakan bahwa bidang profesi rekayasa perangkat lunak telah cukup matang, karena dalam bidang profesi lainnya pun, tidak ada teknik pamungkas yang dapat digunakan dalam berbagai kondisi

1.3. Definisi Rekayasa Perangkat Lunak

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.

Pada tahun 1960-an hingga 1980-an, banyak masalah yang ditemukan para praktisi pengembangan perangkat lunak. Banyak *project* yang gagal, hingga masa ini disebut sebagai krisis perangkat lunak. Kasus kegagalan pengembangan perangkat lunak terjadi mulai dari *project* yang melebihi anggaran, hingga kasus yang mengakibatkan kerusakan fisik dan kematian. Salah satu kasus yang terkenal antara lain meledaknya roket Ariane akibat kegagalan perangkat lunak. Selama bertahun-tahun, para peneliti memfokuskan usahanya untuk menemukan teknik jitu untuk memecahkan masalah krisis perangkat lunak.

Pada tahun 1969 Fritz Bauer memberikan definisi rekayasa perangkat lunak adalah sebagai berikut :

"The establishment and use of sound engineering principles in order to obtain economically software that is reliable and work efficiently on real machines."

Hampir setiap pembaca tergoda untuk menambah sendiri definisi tersebut, karena definisi tersebut hanya menyinggung sedikit saja tentang aspek teknis dan kualitas perangkat lunak, dan tidak secara langsung menyinggung kebutuhan dan kepuasan pelanggan, mengabaikan pencamtuman pentingnya pengukuran dan matriks, tidak menyinggung pentingnya sebuah proses. Apakah *sound engineering application* yang dapat diaplikasikan kepada pengembangan komputer? Bagaimana kita secara ekonomis membangun perangkat lunak sehingga menjadi dapat diandalkan dan reliable? Apakah yang dibutuhkan untuk menciptakan program komputer yang bekerja secara efisien pada lebih dari satu mesin riril yang berbeda? Pertanyaan-pertanyaan ini masih terus menjadi tantangan bagi pengembangan perangkat lunak.

Pada tahun 1985 Richard Fairly mendefinisikan rekayasa perangkat lunak sebagai berikut :

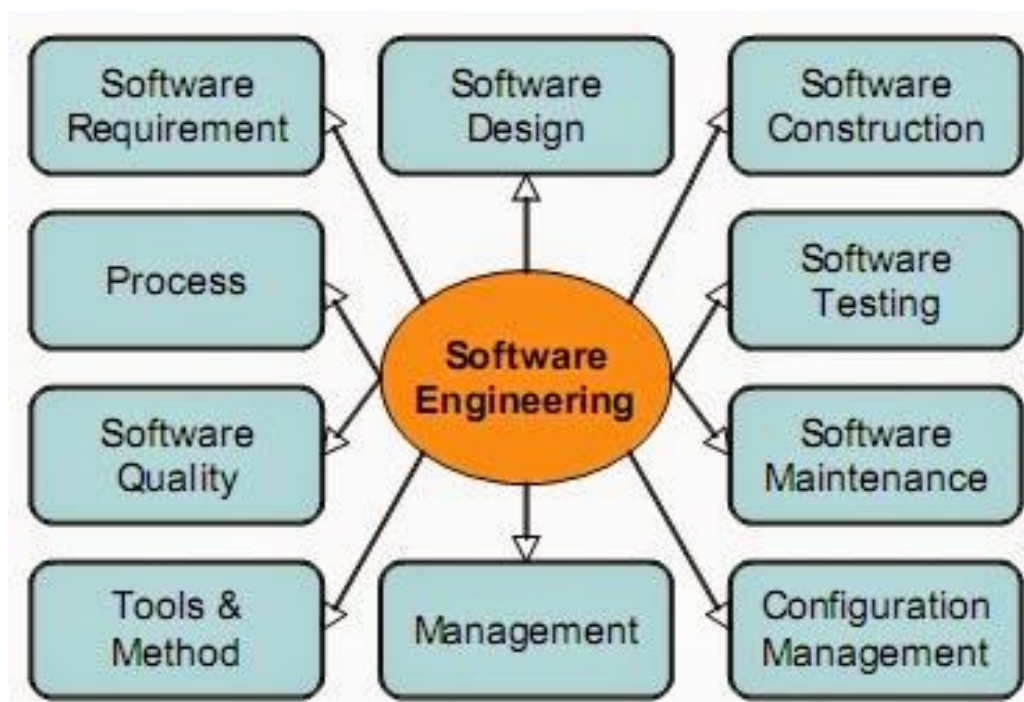
"The technological and managerial dicipline concernment with systematic production and maintenance of software products that are developed and modified on time and within cost estimates."

Definisi ini sudah menyinggung aspek teknis pengembangan perangkat lunak, pengelolaan tim yang terlibat dalam pengembangan tersebut, pemeliharaan perangkat lunak yang telah dikembangkan, serta waktu serta biaya pengembangan. Kemudian pada tahun 1993, IEEE mengembangkan definisi yang lebih komprehensif yaitu sebagai berikut : (1) Aplikasi dari sebuah pendekatan kuantifiabel, disiplin, dan sistematis terhadap pengembangan, operasi, dan pemeliharaan perangkat lunak; yaitu aplikasi dan rekayasa perangkat lunak.

Definisi lain dari perangkat lunak adalah suatu disiplin yang mengintegrasikan proses/prosedur, metode, dan perangkat tools untuk pembangunan perangkat lunak komputer (Pressman, 97).

1.4. Ruang Lingkup Rekayasa Perangkat Lunak

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

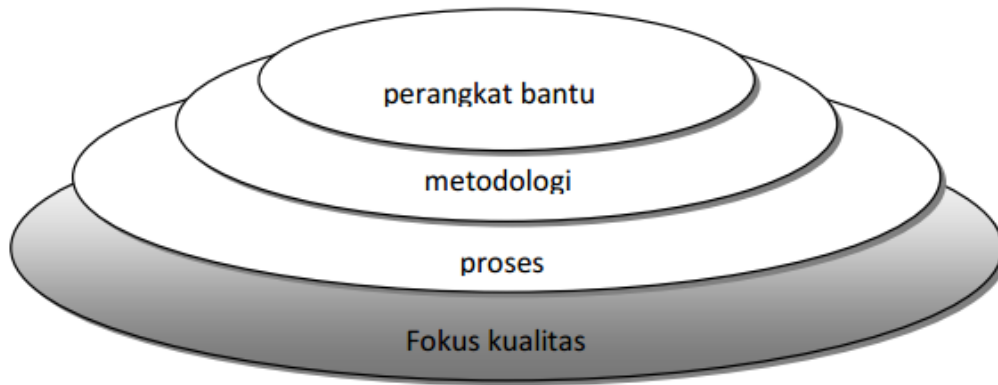


Gambar 1. Ruang lingkup Rekayasa Perangkat Lunak

- *Software Requirements* berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak

- *Software Design* 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 merupakan suatu kegiatan untuk menghasilkan suatu produk, sehingga harus berada pada satu komitmen dasar menuju kualitas, seperti diperlihatkan pada gambar 2. Untuk itu fokus kualitas menjadi batu landasannya. Lingkup kedua adalah proses. Proses-proses rekayasa perangkat lunak adalah perekat yang menjaga bentangan teknologi secara bersama-sama dan memungkinkan perkembangan perangkat lunak yang tepat waktu dan rasional. Proses-proses tersebut membatasi kerangka kerja untuk serangkaian area proses kunci yang harus dibangun demi keefektifan penyampaian teknologi pengembangan perangkat lunak. Area proses kunci ini membentuk dasar bagi kontrol manajemen proyek pengembangan perangkat lunak serta membangun konteks dimana metode teknis diaplikasikan sehingga sebuah produk yang berkualitas bisa dihasilkan.



Gambar 2. Ruang Lingkup Rekayasa Perangkat Lunak

Lingkup berikutnya adalah metodologi, yaitu sekumpulan metode untuk melaksanakan setiap tahap pengembangan perangkat lunak, yang meliputi : perencanaan dan estimasi proyek, analisa kebutuhan, prosedur algoritma dan arsitektur program, menulis program (*coding*), pengujian (*testing*), dan pemeliharaan (*maintenance*). Terakhir adalah perangkat bantu (*tools*). Perangkat bantu yang dimaksud adalah suatu perangkat, baik lunak atau keras, otomatis maupun semi-otomatis yang bisa digunakan untuk proses pengembangan perangkat lunak. Tools untuk rekayasa perangkat lunak disebut *computer-aided software engineering* (CASE). CASE ini terus dikembangkan untuk menciptakan lingkungan rekayasa perangkat lunak sehingga analog dengan CAD/CAE (*computer-aided design/engineering*) pada pengembangan perangkat keras.

1.5. Rekayasa Perangkat Lunak vs Rekayasa Sistem

Rekayasa sistem adalah kumpulan konsep, pendekatan dan metodologi, serta alat-alat bantu (*tools*) untuk merancang dan menginstalasi sebuah kompleks sistem. Kompleksitas sistem bisa diakibatkan karena 2 hal yaitu kompleksitas dinamis dan kompleksitas detail. Kompleksitas detail ketika komponen atau sub-sistem yang dirancang tidak hanya banyak tetapi ditambah pula dengan *multi-sourcing* (*multi supplier*), *multi standard*, *multi criteria* dan lainnya.

Dalam lingkup pengembangan perangkat lunak, rekayasa Sistem adalah kegiatan untuk menentukan spesifikasi, perancangan, pengimplementasian, penyebaran, dan

pemeliharaan sistem sebagai satu kesatuan. Sehingga, rekayasa sistem atau lebih tepatnya, rekayasa sistem berbasis komputer berhubungan dengan semua aspek pengembangan dan evolusi sistem kompleks dimana perangkat lunak memainkan peran utama. Rekayasa sistem merupakan disiplin yang lebih tua dibandingkan dengan rekayasa perangkat lunak. Orang telah melakukan spesifikasi dan perakitan sistem industri secara kompleks seperti kereta api dan pabrik kimia selama lebih dari 100 tahun. Akan tetapi, seiring dengan bertambahnya persentase perangkat lunak pada sistem, maka teknik rekayasa perangkat lunak seperti pemodelan use – case, manajemen konfigurasi, dan lain sebagainya sering dipergunakan dalam proses rekayasa sistem

Sommerville mendefinisikan sistem sebagai sekumpulan komponen yang saling berhubungan dan bekerja sama untuk mencapai tujuan. Definisi umum ini mencakup banyak jenis sistem. Sebagai contoh, sistem yang sederhana seperti sistem pencatatan skor mungkin hanya terdiri dari 2 atau 3 modul perangkat lunak. Sebaliknya, sistem kontrol lalu lintas dapat terdiri dari ratusan perangkat lunak dan keran, ditambah manusia sebagai pemakainya, yang membuat keputusan berdasarkan informasi dari sistem

Rekayasa sistem 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, infrastruktur, kontrol, aplikasi dan database pada sistem.

1.6. Rekayasa Perangkat Lunak vs Ilmu Komputer

Software engineering adalah bagaimana merancang dan membangun sebuah perangkat lunak serta perawatan dan pengembangannya sesuai kebutuhan manusia, *Computer Science* adalah bagaimana ilmu dan teknologi komputer itu dieksplorasi pemanfaatannya untuk kebutuhan manusia sedangkan *system engineering* bertujuan untuk menghasilkan sistem yang memenuhi kebutuhan pelanggan, meningkatkan probabilitas keberhasilan sistem, mengurangi risiko dan mengurangi total biaya siklus-hidup.

1.7. Kaitan nilai kebudiluhuran dalam mata kuliah

Nilai kebudiluhuran merupakan falsafah nilai-nilai etika dan moral yang dikembangkan oleh Universitas Budi Luhur dalam rangka mencapai visinya untuk mencetak generasi yang cerdas berbudi luhur. Dalam konteks transfer nilai-nilai kebudiluhuran ini ke dalam mata kuliah, terdapat 9 nilai yang harus diinternalisasikan ke dalam proses pembelajaran mata kuliah, antara lain :

1) Sabar Mensyukuri

Sabar Mensyukuri merupakan sikap menahan diri atau menerima, yang berarti tidak ada upaya untuk mengelak dengan perasaan senang saat menerima anugerah, baik berupa kesenangan atau kesedihan dengan dasar bahwa segala sesuatu yang baik atau buruk merupakan buah dari perbuatan yang sudah kita lakukan. Misal pada saat ada masalah tidak marah, tidak mudah putus asa. Syukuri apa saja yang kita terima dan terus berusaha. Berserah dan mendekatkan diri kepada Tuhan.

2) Cinta Kasih

Cinta Kasih merupakan perasaan suka yang mendalam yang diwujudkan secara nyata dalam bentuk kepedulian terhadap sesama manusia, hewan, tumbuhan, dan lingkungan atau alam sekitarnya.

3) Suka menolong

Suka Menolong merupakan tindakan untuk meringankan beban atau penderitaan orang lain, tidak terbatas pada pemberian materi atau benda saja, tetapi bisa juga sesuatu yang non materil. Suka menolong bukan berarti asal menolong, namun pertolongan tersebut harus menimbulkan rasa senang bagi orang yang ditolong ataupun yang menolong.

4) Jujur

Jujur merupakan sikap atau sifat seseorang yang menyatakan sesuatu dengan sesungguhnya dan apa adanya, tidak ditambahi atau dikurangi, sesuai dengan fakta dan obyektif sehingga dapat dipercaya semua ucapannya.

5) Tanggungjawab

Tanggung Jawab merupakan kesadaran manusia akan tingkah laku atau perbuatan baik yang disengaja atau tidak disengaja.

6) Rendah Hati

Rendah Hati merupakan sifat pribadi seseorang yang dapat memposisikan sama antara dirinya dengan orang lain, merasa tidak lebih pintar, baik, mahir, serta tidak merasa lebih tinggi atau mulia dan juga dapat menghargai orang lain.

7) Toleransi

Toleransi merupakan suatu sikap atau perilaku seseorang menerima pihak lain dan menghargai perbedaan atau tindakan orang lain. Toleransi dapat dilihat dari berbagai aspek, selama itu tidak melanggar unsur SARA sikap toleransi sangat dibutuhkan.

8) Kerjasama

Kerjasama merupakan kegiatan yang dilakukan dengan orang lain, dengan cara menyatukan kekuatan untuk mencapai satu tujuan yang diinginkan bersama.

9) Sopan santun

Kerjasama merupakan kegiatan yang dilakukan dengan orang lain, dengan cara menyatukan kekuatan untuk mencapai satu tujuan yang diinginkan bersama.

Internalisasi nilai-nilai kebudiluhuran ini jelas sangat dibutuhkan dalam proses pengembangan perangkat lunak. Karena memproduksi perangkat lunak yang sejatinya dimaksudkan untuk memberi solusi dari permasalahan individu atau organisasi, pasti akan selalu berhadapan dengan kondisi-kondisi yang boleh jadi tidak mudah. Pun pada saat kita bekerja dalam tim pengembangan perangkat lunak dibutuhkan sikap kepedulian untuk saling memotivasi keberhasilan dari capaian yang dituju. Dan selama proses tersebut nilai-nilai kebudiluhuran tersebut akan sangat bermanfaat untuk menunjang keberhasilan kinerja tim seperti kerjasama, toleransi, cinta kasih, saling menolong dan lain-lain tersebut.

Rangkuman

Dari pembahasan pada bab 1 ini kita sudah mengetahui bahwa keberadaan perangkat lunak saat ini sudah semakin menjamur dan eksistensinya sangat dibutuhkan sejalan dengan kemajuan alat komputasi serta tuntutan pemrosesan yang cepat dan akurat di segala sisi. Sejalan dengan hal tersebut, maka perkembangan industri perangkat lunak juga akan semakin melahirkan berbagai jenis produk untuk menjawab berbagai kebutuhan fungsi yang dibutuhkan manusia saat ini.

Latihan

Tuliskanlah dengan contoh penerapan setiap nilai-nilai kebudiluhuran dalam implementasi rekayasa perangkat lunak!



FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan

Jakarta Selatan, 12260

Telp: 021-5853753 Fax : 021-5853752

<http://fti.budiluhur.ac.id>