

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

VERSI 1.0
N



PERTEMUAN 4

KARAKTERISTIK DAN MITOS PERANGKAT LUNAK

Capaian Pembelajaran	:	<ol style="list-style-type: none">1. Mahasiswa mengetahui mitos seputar pengembangan perangkat lunak2. Mahasiswa memahami peranan perangkat lunak serta karakteristik dari produk perangkat lunak yang baik
Sub Pokok Bahasan	:	<ol style="list-style-type: none">1.1. Mitos seputar pengembangan perangkat lunak1.2. Peranan ganda perangkat lunak1.3. Karakteristik perangkat lunak yang baik
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 Education5. Software Engineering Body of Knowledge (SWEBOK). 2004

1.1. Karakteristik Perangkat Lunak

1.1.1. Peranan ganda perangkat lunak

Perangkat lunak memiliki peranan ganda, yaitu :

- perangkat lunak adalah sebuah produk
 - menyajikan potensi komputasi
 - menghasilkan, mengola, memperoleh, modifikasi, merapikan / mengirimkan informasi
- perangkat lunak adalah alat untuk menyajikan sebuah produk
 - mendukung / menyediakan langsung fungsi sistem
 - mengendalikan program lain (misalnya sistem operasi)
 - mempengaruhi komunikasi (misalnya perangkat lunak jaringan)
 - membantu membangun perangkat lain (misalnya tool perangkat lunak)

1.1.2. Karakteristik Perangkat Lunak

Berikut ini adalah karakteristik dari perangkat lunak (Roger S.Pressman, 2001)

- Perangkat lunak dibangun dan dikembangkan, tidak dibuat dalam bentuk yang klasik (pabrikasi). Biaya untuk perangkat lunak dikonsentrasikan kepada pengembangan. Hal ini berarti proyek perangkat lunak tidak dapat diatur seperti pengaturan proyek-proyek pemanufakturan.
- Perangkat lunak tidak pernah usang. Perangkat lunak tidak rentan terhadap pengaruh lingkungan yang merusak yang menyebabkan perangkat keras menjadi usang. Selama hidupnya, perangkat lunak mengalami perubahan (pemeliharaan). Aspek lain dari keusangan menggambarkan perbedaan antara perangkat keras dan perangkat lunak. Bila komponen suatu perangkat keras telah usang, komponen dapat diganti dengan suku cadangnya. Namun tidak ada suku cadang bagi perangkat lunak. Setiap kegagalan perangkat lunak menggambarkan kesalahan dalam perancangan atau proses di mana rancangan diterjemahkan ke dalam kode mesin yang dapat dieksekusi.
- Sebagian besar perangkat lunak dibuat secara *custom-built*, serta tidak dapat dirakit dari komponen yang sudah ada. Perhatikan bagaimana perangkat keras

untuk produksi berbasis mikroprosesor dirancang dan dibuat. Setelah masing-masing komponen diseleksi, perangkat keras dapat dipesan secara terpisah. Sementara pada perangkat lunak, tidak katalog komponen perangkat lunak. Memang memungkinkan untuk memesan perangkat lunak secara terpisah, tetapi tetap merupakan satu kesatuan yang lengkap, bukan sebagai komponen yang dapat dipasangkan ke dalam program-program yang baru.

- Zero Tolerance

Maknanya perangkat lunak itu akan terus dapat digunakan/bermanfaat walaupun proses pengembangannya dinyatakan telah selesai atau sudah tidak melakukan peningkatan versi dan update. Contoh OS windows XP sudah tutup versi di bulan april 2014, namun masih akan tetap ada saja pengguna yang menggunakan versi OS tersebut meski telah tutup versi.

1.1.3. Perbedaan perangkat keras dan perangkat lunak

- Manufacture vs development

Perangkat keras dihasilkan dari serangkaian proses manufaktur, sekali produk perangkat keras diproduksi maka akan sulit atau bahkan tidak mungkin untuk dimodifikasi di kemudian hari. Bila pada produksi manufaktur sang produk apabila sudah di pasarkan maka pihak produsen tidak lagi perlu memelihara produknya. Sementara produk perangkat lunak dibuat dari serangkaian proses pengembangan yang dimungkinkan secara rutin mengalami proses perubahan dan peningkatan.

Dalam pembuatan perangkat keras, jika banyak orang yang terlibat maka akan lebih cepat membantu terselesaikannya pekerjaan, tetapi hal yang berbeda terjadi pada proses rekayasa perangkat lunak. Proses pembuatan perangkat lunak hanya lebih banyak terkonsentrasi pada proses disain daripada produksinya.

- Failure

Ketika komponen perangkat keras aus, dapat diganti dengan suku cadang. Namun tidak ada suku cadang bagi perangkat lunak. Setiap kegagalan perangkat lunak menunjukkan kesalahan dalam desain atau dalam proses melalui mana desain diterjemahkan ke dalam kode yang dapat dieksekusi mesin. Karena itulah,

pemeliharaan perangkat lunak lebih banyak kerumitannya.

- Component based vs custom built

Produk perangkat keras biasanya menggunakan banyak komponen desain standar. Sementara sebagian besar perangkat lunak terus dibuat khusus. Industri perangkat lunak tampaknya bergerak (perlahan) ke arah konstruksi berbasis komponen.

- Visibility of Product

Produk software tidak terlihat dengan kasat mata, termasuk bila ada cacat (*defect*) dari produk. Produk hardware terlihat dengan kasat mata, termasuk bila ada cacat (*defect*) dari produk.

1.2. Karakteristik Perangkat Lunak yang baik

Berikut ini adalah karakteristik dari produk perangkat lunak yang baik :

- Maintainability

Merupakan kemampuan yang mengacu pada kemudahan sisi perawatan dari produk termasuk dalam kebutuhan modifikasi di kemudian hari. Modifikasi yang dimaksud dapat berupa perluasan fungsinya, perbaikan kinerjanya, atau perbaikan kesalahannya. Karakteristik sangat ditunjang oleh faktor dokumentasi yang lengkap serta disain yang baik pada saat pengembangannya. Dokumentasi yang lengkap akan memberi tuntutan bagaimana seluk beluk dari perangkat lunak tersebut sehingga penelusuran kebutuhan modifikasi dapat dengan mudah dilakukan. Sementara disain yang baik akan mendukung stabilitas dari produk sekalipun jika suatu hari nanti dimodifikasi.

- Dependability

Merupakan karakteristik yang mengacu pada kehandalan dari produk terkait dengan :

- Functionality: tingkat kinerja perangkat lunak terhadap tujuan yang telah ditentukan.
- Reliability: kemampuan perangkat lunak untuk menyediakan fungsionalitas yang diinginkan dalam kondisi tertentu.
- Security: kemampuan dari perangkat lunak dalam hal akses data yang

tidak sah dan juga data yang bisa dicegah

- Safety : kemampuan dari perangkat lunak untuk memberi kenyamanan kepada user saat digunakan, seperti tersedianya menu help dan juga menghadirkan fungsionalitas yang tidak banyak kerumitan yang mengejutkan.

- Efficiency

Merupakan karakteristik yang mengacu pada kemampuan dari perangkat lunak dalam menggunakan/mengaplikasikan sumber daya sistem dengan cara yang efisien serta cara yang efektif untuk dipakai.

- Usability

Merupakan karakteristik yang mengacu pada kemampuan dari perangkat lunak sampai sejauh mana bisa untuk digunakan dengan baik dan mudah. Yang menyangkut karakteristik ini antara lain adalah dengan menyajikan tampilan antar muka yang user friendly serta melengkapi produk dengan dokumentasi yang akan membantu user makin mudah dalam mengoperasikan perangkat lunak.

1.3. Mitos Pengembangan Perangkat Lunak

Mitos Software adalah penafsiran-penafsiran pada rekayasa piranti lunak ini secara sekilas terdengar masuk akal, bahkan terkadang memang mengandung unsur kebenaran di dalamnya. Beberapa Mitos Software:

1.3.1. Mitos di sisi manajemen

- Mitos : Kita tidak perlu mengubah pendekatan terhadap pengembangan software, karena jenis pemrograman yang kita lakukan sekarang ini sudah kita lakukan 10 tahun yang lalu.

Realitasnya : Walau hasil program sama, produktivitas dan kualitas software harus ditingkatkan dengan menggunakan pendekatan software developments

- Mitos : Kita sudah mempunyai buku yang berisi standarisasi dan prosedur untuk pembentukan software.

Realitasnya : Memang buku tersebut ada, tetapi apakah buku tersebut sudah dibaca atau buku tersebut sudah ketinggalan jaman (out of date).

- Mitos : Jika kita tertinggal dari jadwal yang ditetapkan, kita menambah beberapa programmer saja. Konsep ini sering disebut Mongolian harde concept.

Realitasnya : dengan menambah tenaga baru ditengah proyek yang sudah berjalan justru akan membuat proyek semakin terlambat.

- Mitos: Jika sebuah perusahaan memutuskan untuk melakukan outsourcing suatu proyek software, perusahaan tidak perlu melakukan apapun dan cukup membiarkan perusahaan outsource untuk melakukan keseluruhan proyek.

Realitas: Apabila sebuah perusahaan tidak mengerti bagaimana cara mengatur dan mengontrol sebuah proyek software secara internal, usaha akan selalu dibutuhkan meskipun outsourcing dilakukan.

1.3.2. Mitos di sisi pelanggan (Customer)

- Mitos : Pernyataan tujuan umum sudah cukup untuk memulai penulisan program. Penjelasan yang lebih rinci akan menyusul kemudian.

Realitas: Definisi awal yang buruk adalah penyebab utama kegagalan terhadap usaha-usaha pem-bentukan software. Penjelasan yang formal dan terinci tentang informasi fungsi performance interface, hambatan desain dan kriteria validasi adalah penting. Karakteristik di atas dapat ditentukan hanya setelah adanya komunikasi antara customer dan developer.

- Mitos : Kebutuhan proyek yang terus menerus berubah dapat dengan mudah diatasi karena software itu bersifat fleksibel.

Realitas: Kenyataannya memang benar bahwa kebutuhan software berubah, tetapi dampak dari perubahan berbeda dari waktu ke waktu. Jika perubahan mendekati akhir penyelesaian, maka biaya akan lebih besar.

- Mitos: Satu-satunya produk yang dapat disampaikan dari proyek yang berhasil adalah program yang dapat dijalankan.

Realitas: program hanya merupakan salah satu bagian dari konfigurasi software

yang terdiri dari banyak elemen lainya (contoh: model, dokumentasi, perencanaan, dan lain-lain).

1.3.3. Mitos Praktisi

- Mitos : Tidak ada metode untuk analisis disain dan testing terhadap suatu pekerjaan, cukup menuju ke depan terminal dan mulai coding.
Realitas: Metode untuk analisis desain dan testing diperlukan dalam pengembangan software.
- Mitos : ketika program ditulis dan dapat dijalankan, pekerjaan kita selesai.
Realitas: pada kenyataannya 60%-80% dari keseluruhan pekerjaan dilakukan setelah software diberikan kepada user untuk pertama kalinya.
- Mitos : Segera setelah software digunakan, pemeliharaan dapat diminimalisasikan dan diatasi dengan cara "CATCH AS CATCH CAN".
Realitas: Diperlukan budget yang besar dalam maintenance software. Pemeliharaan software harus diorganisir, direncanakan dan dikontrol seolah-olah sebagai suatu proyek besar dalam sebuah or-ganisasi.

Rangkuman

Produk perangkat lunak memiliki karakteristik yang berbeda jauh dengan produk perangkat keras terutama dalam hal produksinya. Sehingga disinilah dibutuhkan konsep rekayasa dalam produksi perangkat lunak agar terhindar dari kegagalan proyek pengembangannya. Di samping itu keberhasilan dalam proses pengembangan perangkat lunak juga haruslah memperhatikan pemahaman mengenai software myth sehingga tim pengembang tidak terjebak pada kesalahan dalam pola pikir serta pengambilan keputusan yang mengarah pada kegagalan.

Latihan

Jawablah soal berikut dengan memilih satu jawaban yang tepat menurut Anda

1. Produk perangkat lunak dapat dikelompokkan menjadi dua jenis, apa saja kelompok tersebut?
 - a. Produk generik dan produk spesifik
 - b. Produk asli dan produk umum
 - c. Produk generik dan produk asli
 - d. Produk umum dan produk spesifik
2. Apa yang dimaksud dengan produk generik?
 - a. Perangkat lunak untuk keperluan dunia farmasi
 - b. Perangkat lunak untuk keperluan sebuah perusahaan
 - c. Perangkat lunak yang dibuatkan untuk bidang kedokteran
 - d. Perangkat lunak yang dijual untuk umum
3. Yang tidak termasuk sebagai karakteristik software adalah :
 - a. software bersifat logic
 - b. software tidak rusak
 - c. software tergantung pada hardware
 - d. software pada umumnya dikembangkan
4. Ada empat ciri PL yang sudah direkayasa dengan baik, berikut ini mana yang termasuk bagian dari ciri tersebut?
 - a. Mudah disesuaikan dengan perubahan kebutuhan user
 - b. Mudah diperjualbelikan
 - c. Mempunyai dokumentasi yang baik
 - d. Jawaban a dan c benar
5. Apa diantaranya faktor yang harus dimiliki oleh sebuah perangkat lunak untuk dapat dikatakan mempunyai kualitas yang baik?
 - a. Program berukuran kecil
 - b. Jumlah antarmuka(interface) pemakai sedikit

- c. Jumlah laporan sedikit
 - d. Aman dan tahan gangguan
6. Ada empat ciri PL yang sudah direkayasa dengan baik, berikut ini mana yang merupakan keempat ciri tersebut?
- a. Mudah dirawat, dapat diandalkan, efisien, mudah dijual
 - b. Mudah dirawat, dapat diandalkan, efisien, mudah dipakai
 - c. Mudah dijual, dapat diandalkan, efisien, mudah dipakai
 - d. Mudah dirawat, disukai pembeli, efisien, mudah dipakai
7. Apa yang dimaksud dengan rancangan yang mudah dirawat (maintainable design)?
- a. rancangan yang dirancang dengan baik
 - b. rancangan yang dianalisa dengan baik
 - c. rancangan yang siap (mudah) diadaptasi
 - d. rancangan yang dapat diprogram dengan baik
8. Apa yang dimaksud ciri PL yang efisien?
- a. Membutuhkan memory (tempat penyimpanan eksternal dan internal) yang sedikit
 - b. Membutuhkan siklus proses yang sedikit
 - c. Harus dapat disesuaikan dengan perkembangan/perubahan kebutuhan user
 - d. Jawaban a dan b benar
9. Ukuran kemampuan PL untuk di-up grade tanpa mengganggu proses penggunaan PL yang sedang berlangsung adalah salah satu pengertian dari.....
- a. Flexibility
 - b. Testability
 - c. Configurability
 - d. Maintainability

10. Resources yang dibutuhkan untuk menjalankan program, diukur dari :
kecepatan, waktu tanggap dan optimisasi sumber adalah pengertian dari
- a. Expandability
 - b. Efficiency
 - c. Interoperability
 - d. Reliability
11. Suatu Software dapat direcovery jika terjadi kerusakan dan mendapatkan resources secara cepat dan optimal. Hal ini menunjukkan bahwa software tersebut telah sesuai dengan karakteristik
- a. Reliability & Efficiency
 - b. Usability & Efficiency
 - c. Efficiency & robustness
 - d. Reliability & Maintainability
12. Kegiatan yang termasuk dalam tahap perawatan (maintenance) adalah koreksi-adaptasi-peningkatan dan pencegahan. Apa yang dimaksud dengan koreksi?
- a. Memperbaiki fungsi agar sesuai dengan lingkungan baru
 - b. Memperbaiki kesalahan-kesalahan yang ditemukan
 - c. Menambahkan fungsi-fungsi baru
 - d. Penyesuaian dengan lingkungan baru
13. Kegiatan yang termasuk dalam tahap perawatan (maintenance) adalah koreksi-adaptasi-peningkatan dan pencegahan. Apa yang dimaksud dengan adaptasi?
- a. Memperbaiki lingkungan yang ada
 - b. Memperbaiki kesalahan yang ditemukan
 - c. Menambahkan fungsi-fungsi baru
 - d. Penyesuaian dengan lingkungan baru

14. Kegiatan yang termasuk dalam tahap perawatan (maintenance) adalah koreksi-adaptasi-peningkatan dan pencegahan. Apa yang dimaksud dengan peningkatan?
- a. Menambahkan kondisi lingkungan baru
 - b. Memperbaiki kesalahan-kesalahan yang ditemukan
 - c. Menambahkan fungsi-fungsi baru
 - d. Penyesuaian dengan lingkungan baru
15. Kegiatan yang termasuk dalam tahap perawatan (maintenance) adalah koreksi-adaptasi-peningkatan dan pencegahan. Apa yang dimaksud dengan pencegahan?
- a. Menjaga sistem dalam keadaan baik
 - b. Memperbaiki kesalahan-kesalahan yang ditemukan
 - c. Menambahkan fungsi-fungsi baru
 - d. Penyesuaian dengan lingkungan baru



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>