

MSIM4303
Edisi 1

MODUL 01

Pendahuluan Rekayasa Perangkat Lunak

Rosa Ariani Sukamto, S.T., M.T.

Daftar Isi

Modul 01	1.1
Pendahuluan Rekayasa Perangkat Lunak	
Kegiatan Belajar 1	1.5
Perangkat Lunak	
Latihan	1.8
Rangkuman	1.9
Tes Formatif 1	1.10
Kegiatan Belajar 2	1.13
Rekayasa Perangkat Lunak	
Latihan	1.17
Rangkuman	1.18
Tes Formatif 2	1.19
Kegiatan Belajar 3	1.22
Proses Rekayasa Perangkat Lunak	
Latihan	1.27
Rangkuman	1.27
Tes Formatif 3	1.28
Kunci Jawaban Tes Formatif	1.32
Glosarium	1.33
Daftar Pustaka	1.34



Pendahuluan

Rekayasa perangkat lunak merupakan bagian yang penting untuk dipelajari dalam bidang yang terkait dengan pengembangan perangkat lunak. Pengembangan perangkat lunak memiliki banyak tahapan yang perlu dilalui agar dapat menghasilkan perangkat lunak yang dapat digunakan sesuai dengan tujuan dikembangkannya perangkat lunak.

Modul ini merupakan pendahuluan sebelum menjelaskan tahapan-tahapan inti dari rekayasa perangkat lunak itu sendiri agar dapat dipahami beberapa hal penting yang sering disinggung dalam rekayasa perangkat lunak. Modul ini akan berisi terkait:

1. pengertian perangkat lunak atau *software* dari sudut pandang rekayasa perangkat lunak;
2. pengertian rekayasa perangkat lunak atau *software engineering*;
3. pengenalan proses rekayasa perangkat lunak;
4. pengenalan tentang faktor sosial yang berkaitan dengan teknologi informasi.

Pada Modul 1 ini akan membagi materi belajar menjadi tiga kegiatan belajar sebagai berikut.

1. Perangkat Lunak.
2. Rekayasa Perangkat Lunak.
3. Proses Rekayasa Perangkat Lunak.

Setelah mempelajari modul ini mahasiswa diharapkan secara lebih rinci mampu menjelaskan:

1. pengertian perangkat lunak atau *software* dari sudut pandang rekayasa perangkat lunak;
2. pengertian rekayasa perangkat lunak atau *software engineering*;
3. pengenalan proses rekayasa perangkat lunak;
4. pengenalan tentang faktor sosial yang berkaitan dengan teknologi informasi.

Untuk dapat memahami Modul 1 ini, Anda dapat melakukan beberapa strategi di antaranya:

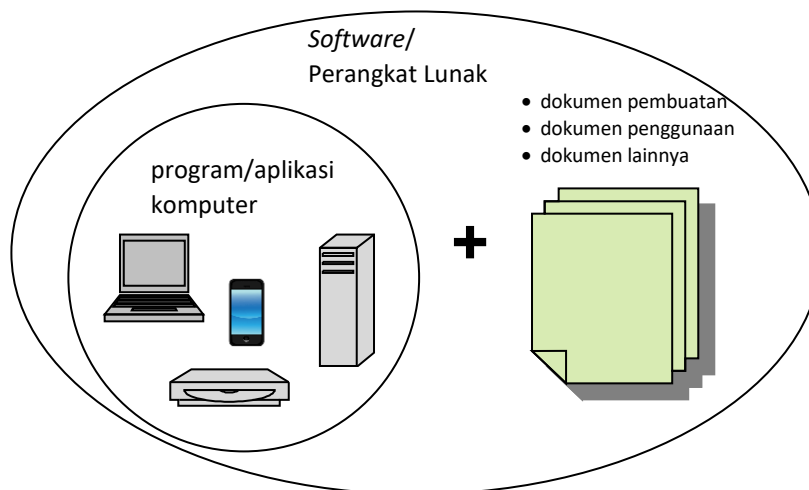
1. silakan baca setiap Kegiatan Belajar pada Modul 1 ini sampai tuntas;
2. baca setiap paragraf dalam modul ini kemudian pahami sampai Anda benar-benar paham maksud dari paragraf tersebut. Jika masih belum paham maka baca ulang paragraf tersebut;
3. untuk meyakinkan bahwa Anda sudah benar-benar memahami materinya, kerjakan latihan-latihan di modul ini. Latihan-latihan yang belum bisa Anda jawab, dapat dicari kembali jawabannya dengan membaca materi yang bersangkutan;

4. jawablah tes formatif yang telah disediakan untuk dapat mengukur tingkat pemahaman Anda. Jika jawaban Anda kurang dari 80%, maka Anda harus membaca ulang bagian yang belum dipahami;
5. kuasai arti istilah-istilah yang ada dengan baik. Pastikan Anda memahami istilah-istilah penting tersebut;
6. pastikan tujuan dari materi modul ini tercapai oleh Anda. Jika tujuan dari modul ini belum tercapai maka baca kembali setiap kegiatan pembelajaran pada modul ini.

Selamat belajar!

Perangkat Lunak

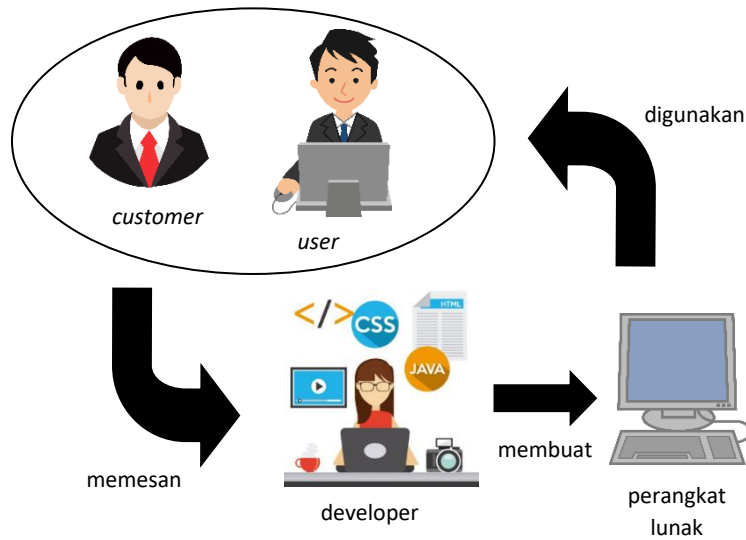
Perangkat lunak (*software*) dari sudut pandang rekayasa perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, cara penggunaan (*user manual*), dokumen teknis, maupun dokumen lainnya yang dapat mendukung program komputer untuk terus digunakan maupun dikembangkan. Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak (*software*) dari sudut pandang rekayasa perangkat lunak. Ilustrasi perangkat lunak dapat dilihat pada Gambar 1.1.



Gambar 1.1
Ilustrasi Perangkat Lunak

Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak karena dapat terdiri dari banyak modul-modul maupun bagian-bagian dari program komputer yang berbeda di berbagai lingkungan pengembangan. Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai. Sistem perangkat lunak berarti sebuah sistem yang memiliki komponen berupa perangkat lunak yang memiliki hubungan satu sama lain untuk memenuhi kebutuhan pelanggan (*customer*). Pelanggan (*customer*) adalah orang atau organisasi yang memesan atau membeli perangkat lunak (*software*) dari pengembang (*developer*).

perangkat lunak atau bisa dianggap bahwa pelanggan (*customer*) adalah orang atau organisasi yang dengan sukarela mengeluarkan uang untuk memesan atau membeli perangkat lunak. *User* atau pengguna perangkat lunak adalah orang yang memiliki kepentingan untuk memakai atau menggunakan perangkat lunak untuk memudahkan pekerjaannya.



Gambar 1.2
Ilustrasi Hubungan Perangkat Lunak yang Dipesan *Customer*



Gambar 1.3
Ilustrasi Hubungan Perangkat Lunak yang Dipasarkan kepada *User*

Ilustrasi hubungan antara perangkat lunak dengan pihak-pihak yang terkait (*stakeholder*) dapat dilihat pada Gambar 1.2 dan 1.3. Ilustrasi pada Gambar 1.2 adalah untuk perangkat lunak yang dipesan oleh *customer* (*customized software*), sedangkan ilustrasi pada Gambar 1.3 adalah perangkat lunak yang dibuat oleh *developer* untuk dipasarkan kepada *user* secara langsung (*generic products*).

Karakter perangkat lunak adalah:

1. perangkat lunak dibangun dengan rekayasa (*software engineering*) bukan diproduksi secara manufaktur atau pabrikan;
2. perangkat lunak tidak pernah usang (“*wear out*”) karena kecacatan dalam perangkat lunak dapat diperbaiki;
3. barang produksi pabrikan biasanya komponen barunya akan terus diproduksi, sedangkan perangkat lunak biasanya terus diperbaiki seiring bertambahnya kebutuhan.

Aplikasi dari perangkat lunak adalah sebagai berikut.

1. Perangkat lunak sistem (*system software*)
Adalah kumpulan program di mana program yang satu ditulis untuk memenuhi kebutuhan program lainnya.
2. Perangkat lunak waktu nyata (*real-time software*)
Merupakan perangkat lunak yang memonitor, menganalisis, mengontrol sesuatu secara waktu nyata (*real-time*). Reaksi yang dibutuhkan pada perangkat lunak harus langsung menghasilkan respons yang diinginkan.
3. Perangkat lunak bisnis (*business software*)
Merupakan perangkat lunak pengelola informasi bisnis (seperti akuntansi, penjualan, pembayaran, penyimpanan (*inventory*)).
4. Perangkat lunak untuk keperluan rekayasa dan keilmuan (*engineering and scientific software*)
Merupakan perangkat lunak yang mengimplementasikan algoritma yang terkait dengan keilmuan ataupun perangkat lunak yang membantu keilmuan, misalkan perangkat lunak di bidang astronomi, di bidang matematika, dan lain sebagainya.
5. Perangkat lunak tambahan untuk membantu mengerjakan suatu fungsi dari perangkat lunak yang lainnya (*embedded software*)
Misalnya perangkat lunak untuk mencetak dokumen ditambahkan agar perangkat lunak yang memerlukan dapat mencetak laporan, maka perangkat lunak untuk mencetak dokumen ini disebut *embedded software*.
6. Perangkat lunak komputer personal (*personal computer software*)
Merupakan perangkat lunak untuk PC misalnya perangkat lunak pemroses teks, pemroses grafik, dan lain sebagainya.
7. Perangkat lunak berbasis web (*web based software*)
Merupakan perangkat lunak yang dapat diakses dengan menggunakan *browser*.

8. Perangkat lunak berkecerdasan buatan (*artificial intelligence software*)
Merupakan perangkat lunak yang menggunakan algoritma tertentu untuk mengelola data sehingga seakan-akan memiliki kecerdasan seiring bertambahnya data yang diproses.

Produk perangkat lunak yang dibuat oleh pengembang (*developer*) perangkat lunak terdiri atas dua jenis.

1. Produk generik
Produk perangkat lunak yang dibuat oleh pengembang perangkat lunak untuk dijual atau dipopulerkan (*open source*) tanpa ada yang memesan terlebih dahulu, perangkat lunak yang termasuk dalam produk generik misalnya perangkat lunak sistem operasi, perangkat lunak pendukung perkantoran untuk membuat dokumen, *slide* presentasi, atau perhitungan dalam bentuk *papersheet* dan lain sebagainya.
2. Produk pemesanan
Produk perangkat lunak yang dibuat karena ada pelanggan yang melakukan pemesanan, misalnya sebuah instansi memerlukan perangkat lunak untuk memenuhi proses bisnis yang terjadi di instansinya, maka instansi itu akan bekerja sama dengan pengembang untuk membuat perangkat lunak yang diinginkan.



Latihan

Untuk memperdalam pemahaman Anda mengenai materi di atas, kerjakanlah latihan berikut!

- 1) Sebutkan definisi atau pengertian dari istilah-istilah sebagai berikut.
 - a. Perangkat lunak (*software*).
 - b. Sistem (*system*).
 - c. Rekayasa perangkat lunak (*software engineering*).
 - d. *Software developer*.
 - e. *Software customer*.
 - f. *Software user*.
 - g. *System software*.
 - h. *Real-time software*.
 - i. *Business software*.
 - j. *Engineering and Scientific software*.
 - k. *Embedded software*.
 - l. *Personal Computer software*.
 - m. *Web-based software*.

- n. *Artificial Intelligence software*.
 - o. Produk perangkat lunak generik.
 - p. Produk perangkat lunak pemesanan.
- 2) Apakah proses produksi perangkat lunak identik atau serupa dengan proses produksi pada pabrik/manufaktur pembuatan perangkat komputer? Jelaskan alasannya!

Petunjuk Jawaban Latihan

- 1) Pelajari tentang istilah-istilah pada Modul 1 Kegiatan Belajar 1.
- 2) Pelajari tentang karakter perangkat lunak pada Modul 1 Kegiatan Belajar 1.



Rangkuman

Perangkat lunak (*software*) dari sudut pandang rekayasa perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, cara penggunaan (*user manual*), dokumen teknis, maupun dokumen lainnya yang dapat mendukung program komputer untuk terus digunakan maupun dikembangkan.

Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak karena dapat terdiri dari banyak modul-modul maupun bagian-bagian dari program komputer yang berbeda di berbagai lingkungan pengembangan.

Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai. Sistem perangkat lunak berarti sebuah sistem yang memiliki komponen berupa perangkat lunak yang memiliki hubungan satu sama lain untuk memenuhi kebutuhan pelanggan (*customer*). Pelanggan (*customer*) adalah orang atau organisasi yang memesan atau membeli perangkat lunak (*software*) dari pengembang (*developer*) perangkat lunak atau bisa dianggap bahwa pelanggan (*customer*) adalah orang atau organisasi yang dengan sukarela mengeluarkan uang untuk memesan atau membeli perangkat lunak.

Aplikasi dari perangkat lunak yaitu sebagai berikut.

1. *System software*;
2. *Real-time software*;
3. *Business software*;
4. *Engineering and Scientific software*;
5. *Embedded software*;
6. *Personal Computer software*;
7. *Web-based software*;
8. *Artificial Intelligence software*.

Jenis produk perangkat lunak yaitu:

1. produk perangkat lunak generik;
2. produk perangkat lunak pemesanan.



Tes Formatif 1

Pilihlah satu jawaban yang paling tepat!

- 1) Orang atau organisasi yang memesan perangkat lunak disebut dengan
 - A. *user*
 - B. *customer*
 - C. *developer*
 - D. *software*

- 2) Orang atau organisasi yang mengembangkan/membangun perangkat lunak disebut dengan
 - A. *user*
 - B. *customer*
 - C. *developer*
 - D. *software*

- 3) Orang atau organisasi yang menggunakan perangkat lunak disebut dengan
 - A. *user*
 - B. *customer*
 - C. *developer*
 - D. *software*

- 4) Di bawah ini yang bukan karakter perangkat lunak adalah
 - A. perangkat lunak dibangun dengan proses rekayasa bukan manufaktur
 - B. perangkat lunak tidak pernah *wear out*
 - C. perangkat lunak dapat diproduksi secara manufaktur masal
 - D. perangkat lunak terus diperbaiki seiring bertambahnya kebutuhan

- 5) *Real-time software* adalah
 - A. perangkat lunak yang memonitor, menganalisis, mengontrol sesuatu secara waktu nyata
 - B. kumpulan program dimana saling memenuhi kebutuhan program satu sama lain
 - C. perangkat lunak yang mengimplementasikan algoritma terkait keilmuan atau membantu keilmuan
 - D. perangkat lunak tambahan untuk membantu mengerjakan fungsi dari perangkat lunak yang lain

- 6) *Embedded software* adalah
- A. perangkat lunak yang memonitor, menganalisis, mengontrol sesuatu secara waktu nyata
 - B. kumpulan program dimana saling memenuhi kebutuhan program satu sama lain
 - C. perangkat lunak yang mengimplementasikan algoritma terkait keilmuan atau membantu keilmuan
 - D. perangkat lunak tambahan untuk membantu mengerjakan fungsi dari perangkat lunak yang lain
- 7) *Engineering and Scientific software* adalah
- A. perangkat lunak yang memonitor, menganalisis, mengontrol sesuatu secara waktu nyata
 - B. kumpulan program di mana saling memenuhi kebutuhan program satu sama lain
 - C. perangkat lunak yang mengimplementasikan algoritma terkait keilmuan atau membantu keilmuan
 - D. perangkat lunak tambahan untuk membantu mengerjakan fungsi dari perangkat lunak yang lain
- 8) *System software* adalah
- A. perangkat lunak yang memonitor, menganalisis, mengontrol sesuatu secara waktu nyata
 - B. kumpulan program di mana saling memenuhi kebutuhan program satu sama lain
 - C. perangkat lunak yang mengimplementasikan algoritma terkait keilmuan atau membantu keilmuan
 - D. perangkat lunak tambahan untuk membantu mengerjakan fungsi dari perangkat lunak yang lain
- 9) Jenis produk perangkat lunak generik adalah produk perangkat lunak yang dibuat untuk
- A. memenuhi pesanan *customer*
 - B. dijual maupun dipopulerkan
 - C. memenuhi tugas ujian
 - D. memenuhi kelulusan sertifikasi

- 10) Jenis produk perangkat lunak pemesanan adalah produk perangkat lunak yang dibuat untuk
- A. memenuhi pesanan *customer*
 - B. dijual maupun dipopulerkan
 - C. memenuhi tugas ujian
 - D. memenuhi kelulusan sertifikasi

Cocokkanlah jawaban Anda dengan Kunci Jawaban Tes Formatif 1 yang terdapat di bagian akhir modul ini. Hitunglah jawaban yang benar. Kemudian, gunakan rumus berikut untuk mengetahui tingkat penguasaan Anda terhadap materi Kegiatan Belajar 1.

$$\text{Tingkat Penguasaan} = \frac{\text{Jumlah Jawaban yang Benar}}{\text{Jumlah Soal}} \times 100$$

Arti tingkat penguasaan



Apabila mencapai tingkat penguasaan 80% atau lebih, Anda dapat meneruskan dengan Kegiatan Belajar 2. **Bagus!** Jika masih di bawah 80%, Anda harus mengulangi materi Kegiatan Belajar 1, terutama bagian yang belum dikuasai.

Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (*software engineering*) pertama kali mulai diperkenalkan pada tahun 1968 pada sebuah konferensi yang mendiskusikan krisis perangkat lunak. Hal ini terkait kurang berhasilnya beberapa pendekatan pengembangan program komputer, terutama untuk skala besar terkait keandalan, biaya, dan waktu. Antara tahun 1970-an sampai 1980-an, teknik dan metode rekayasa perangkat lunak banyak dikembangkan, seperti terkait pemrograman terstruktur, pemrograman berorientasi objek, dan lain sebagainya. Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi maupun manfaat yang dipercaya dan bekerja secara efisien menggunakan mesin komputer. Perangkat lunak banyak dibuat dan pada akhirnya sering tidak digunakan karena tidak memenuhi kebutuhan pelanggan (*customer*) atau bahkan karena masalah non-teknis seperti keengganan pengguna perangkat lunak (*user*) untuk mengubah cara kerja dari manual ke otomatis, atau ketidakmampuan *user* menggunakan komputer. Oleh karena itu, rekayasa perangkat lunak dibutuhkan agar perangkat lunak yang dibuat tidak hanya menjadi perangkat lunak yang tidak terpakai.

Rekayasa perangkat lunak lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan (*customer*) dan dapat digunakan oleh pengguna (*user*) dengan baik. Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria:

1. dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (*maintainability*);
2. dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*dependability* dan *robust*);
3. efisien dari segi sumber daya dan penggunaan;
4. kemampuan untuk dipakai sesuai dengan kebutuhan (*usability*).

Dari kriteria di atas maka perangkat lunak yang baik adalah perangkat lunak yang dapat memenuhi kebutuhan pelanggan (*customer*) atau *user* (pengguna perangkat lunak) atau berorientasi pada pelanggan atau pengguna perangkat lunak, bukan berorientasi pada pembuat atau pengembang perangkat lunak.

Pekerjaan yang terkait dengan rekayasa perangkat dapat dikategorikan menjadi tiga buah kategori umum tanpa melihat area dari aplikasi, ukuran proyek perangkat lunak, atau kompleksitas perangkat lunak yang akan dibuat. Setiap fase dialamatkan pada satu atau lebih pertanyaan yang diajukan sebelumnya.

Fase pendefinisian fokus pada “*what*” yang artinya harus mencari tahu atau mengidentifikasi informasi apa yang harus diproses, seperti apa fungsi dan performansi yang diinginkan, seperti apa perilaku sistem yang diinginkan, apa kriteria validasi yang dibutuhkan untuk mendefinisikan sistem.

Fase pengembangan yang fokus dengan “*how*” yang artinya selama tahap pengembangan perangkat lunak seorang perancang perangkat lunak (*software engineer*) berusaha untuk mendefinisikan bagaimana data distrukturkan dan bagaimana fungsi-fungsi yang dibutuhkan diimplementasikan di dalam arsitektur perangkat lunak, bagaimana detail prosedural diimplementasikan, bagaimana karakter antarmuka tampilan, bagaimana desain ditranslasikan ke bahasa pemrograman, dan bagaimana pengujian akan dijalankan.

Fase pendukung (*support phase*) fokus pada perubahan yang terasosiasi pada perbaikan kesalahan (*error*), adaptasi yang dibutuhkan pada lingkungan perangkat lunak yang terlibat, dan perbaikan yang terjadi akibat perubahan kebutuhan pelanggan (*customer*). Fase pendukung terdiri dari empat tipe perubahan.

1. Koreksi (*Correction*)

Walaupun dengan jaminan kualitas yang terbaik, akan selalu ada kecacatan atau keinginan pelanggan (*customer*) yang tidak tertangani oleh perangkat lunak. Pemeliharaan dengan melakukan perbaikan terhadap kecacatan perangkat lunak.

2. Adaptasi (*Adaptation*)

Pada saat tertentu lingkungan asli (seperti CPU, sistem operasi, aturan bisnis, karakteristik produk luar) di mana perangkat lunak dikembangkan akan mengalami perubahan. Pemeliharaan adaptasi merupakan tahap untuk memodifikasi perangkat lunak guna mengakomodasi perubahan lingkungan luar di mana perangkat lunak dijalankan.

3. Perbaikan (*Enhancement*)

Sejalan dengan digunakannya perangkat lunak, maka pelanggan (*customer*) atau penggunanya (*user*) akan mengenali fungsi tambahan yang dapat mendatangkan manfaat. Pemeliharaan perfektif atau penyempurnaan melakukan ekstensi atau penambahan pada kebutuhan fungsional sebelumnya.

4. Pencegahan (*Prevention*)

Keadaan perangkat lunak komputer sangat dimungkinkan untuk perubahan. Oleh karena itu pemeliharaan pencegahan (*preventive*) atau sering disebut juga dengan rekayasa ulang sistem (*software reengineering*) harus dikondisikan untuk mampu

melayani kebutuhan penggunanya (*user*). Untuk menanggulangi hal ini maka perangkat lunak harus dirancang dan dikondisikan untuk mengakomodasi perubahan kebutuhan yang diinginkan oleh penggunanya (*user*). Di lain sisi biasanya setelah perangkat lunak dikirimkan ke *user* maka masih dibutuhkan asistensi dan *help desk* dari pengembang perangkat lunak.

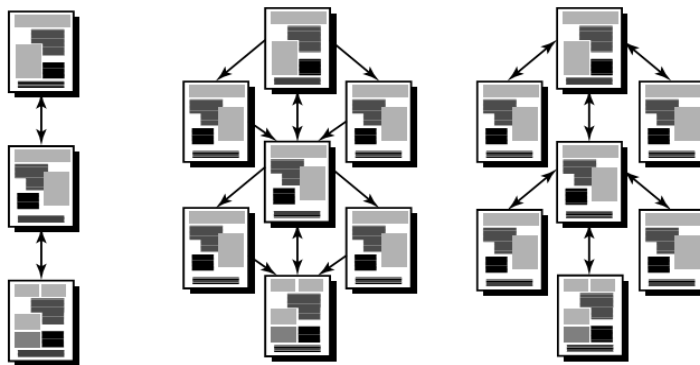
Tantangan yang dihadapi dari proses rekayasa perangkat lunak adalah:

1. tantangan warisan di mana perangkat lunak dikembangkan selama bertahun-tahun oleh orang-orang yang berbeda, hal ini dapat menyebabkan ketidakpahaman atau perubahan tujuan pembuatan perangkat lunak;
2. tantangan heterogenitas di mana perangkat lunak harus dapat beradaptasi dengan teknologi yang terus berkembang dengan semakin luasnya lingkungan distribusi perangkat lunak;
3. tantangan pengiriman di mana perangkat lunak dengan skala besar dan kompleks sekalipun dapat sampai ke tangan pelanggan (*customer*) atau *user* dengan cepat dan kualitas tetap terjaga.

Ada suatu hal tambahan dalam rekayasa perangkat lunak yang berbasis *web* pada tahap desainnya yaitu bagaimana menentukan aliran halaman *web*. Beberapa aliran halaman *web* misalnya adalah sebagai berikut.

a. *Struktur linier*

Pada struktur linier interaksi pengguna pada aplikasi web dapat diramalkan urutannya. Contoh pada urutan pemesanan produk yang di lakukan secara tahap demi tahap (*step by step*). Semakin kompleks aplikasi web memungkinkan urutan tidak berjalan murni linier.

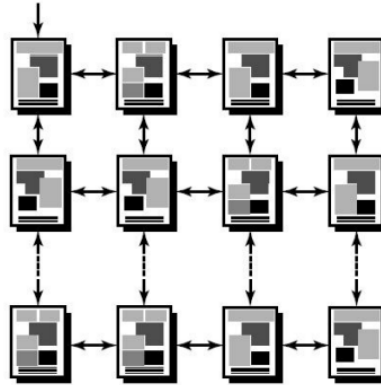


Sumber: Pressman (2001)

Gambar 1.4
Contoh Aliran Halaman Web Struktur Linier

b. *Struktur grid*

Struktur ini diterapkan dalam 2 dimensi yaitu vertikal dan horizontal. Contoh penggunaan pada aplikasi *e-commerce* yang menjual laptop merek tertentu, dimensi horizontal menggambarkan jenis seri laptopnya, sedangkan dimensi vertikal memuat penawaran/harga barang.

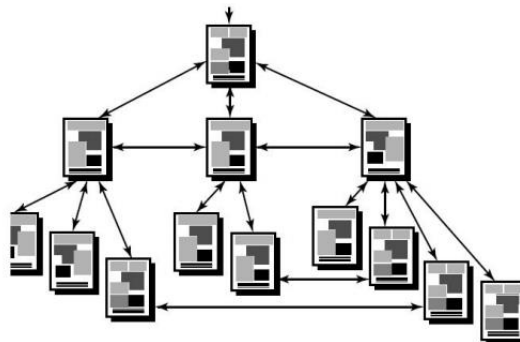


Sumber: Pressman (2001)

Gambar 1.5
Contoh Aliran Halaman Web Struktur *Grid*

c. *Struktur hierarki*

Struktur ini merupakan arsitektur yang umum dijumpai pada aplikasi web. Hierarki sebelah kiri bisa digunakan sebagai tempat untuk menyimpan *link-link* yang menuju ke hierarki bagian tengah maupun kanan. Namun hierarki ini mempunyai kelemahan apabila terlalu banyak *link* yang ditampilkan bisa menimbulkan kebingungan untuk penggunaanya.

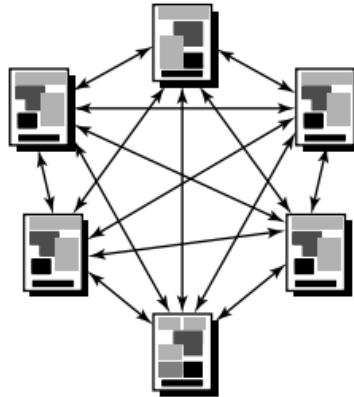


Sumber: Pressman (2001)

Gambar 1.6
Contoh Aliran Halaman Web Struktur Hierarki

c. *Struktur jaringan*

Struktur jaringan ini bisa disebut juga dengan struktur web murni, selain itu struktur ini mempunyai penelusuran yang lebih fleksibel, namun memiliki kelemahan apabila memuat *link* yang terlalu banyak dapat menimbulkan kebingungan untuk penggunanya.



Sumber: Pressman (2001)

Gambar 1.7
Contoh Aliran Halaman Web Struktur Jaringan

Struktur isi pada web yang dibuat bisa saja campuran dari beberapa struktur tersebut. Misalnya pada bagian awal pada *e-commerce* menggunakan hierarki jaringan, sedangkan pada bagian pemesanan barang menggunakan hierarki linier, sedangkan pada hierarki penawaran barang dapat menggunakan hierarki *grid*.



Latihan

Untuk memperdalam pemahaman Anda mengenai materi di atas, kerjakanlah latihan berikut!

- 1) Bidang rekayasa perangkat lunak apakah sebagai bagian dari seni atau bagian dari teknik? Jelaskan alasannya!
- 2) Mengapa ada proses-proses atau tahapan-tahapan yang harus dilakukan dalam rekayasa perangkat lunak? Jelaskan proses-proses utamanya!
- 3) Mengapa rekayasa perangkat lunak sebaiknya fokus pada pelanggan atau pengguna?
- 4) Jelaskan tantangan yang dihadapi dari proses rekayasa perangkat lunak!

Petunjuk Jawaban Latihan

- 1) Pelajari tentang rekayasa perangkat lunak pada Modul 1 Kegiatan Belajar 2.
- 2) Pelajari tentang proses rekayasa perangkat lunak pada Modul 1 Kegiatan Belajar 2.
- 3) Pelajari tentang rekayasa perangkat lunak pada Modul 1 Kegiatan Belajar 2.
- 4) Pelajari tentang tantangan rekayasa perangkat lunak pada Modul 1 Kegiatan Belajar 2.

**Rangkuman**

Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi maupun manfaat yang dipercaya dan bekerja secara efisien menggunakan mesin komputer.

Rekayasa perangkat lunak lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan (*customer*) dan dapat digunakan oleh pengguna (*user*) dengan baik.

Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria:

- a. dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (*maintainability*);
- b. dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*dependability* dan *robust*);
- c. efisien dari segi sumber daya dan penggunaan;
- d. kemampuan untuk dipakai sesuai dengan kebutuhan (*usability*).

Fase utama rekayasa perangkat lunak sebagai berikut.

- a. Fase pendefinisian fokus pada “*what*” yang artinya harus mencari tahu atau mengidentifikasi informasi apa yang harus diproses, seperti apa fungsi dan performansi yang diinginkan, seperti apa perilaku sistem yang diinginkan, apa kriteria validasi yang dibutuhkan untuk mendefinisikan sistem.
- b. Fase pengembangan yang fokus dengan “*how*” yang artinya selama tahap pengembangan perangkat lunak seorang perekayasa perangkat lunak (*software engineer*) berusaha untuk mendefinisikan bagaimana data distrukturkan dan bagaimana fungsi-fungsi yang dibutuhkan diimplementasikan di dalam arsitektur perangkat lunak, bagaimana detail prosedural diimplementasikan, bagaimana karakter antarmuka tampilan, bagaimana desain ditranslasikan ke bahasa pemrograman, dan bagaimana pengujian akan dijalankan.
- c. Fase pendukung (*support phase*) fokus pada perubahan yang terasosiasi pada perbaikan kesalahan (*error*), adaptasi yang dibutuhkan pada lingkungan perangkat lunak yang terlibat, dan perbaikan yang terjadi akibat perubahan kebutuhan pelanggan (*customer*).

- Tantangan yang dihadapi dari proses rekayasa perangkat lunak adalah:
- tantangan warisan dimana perangkat lunak dikembangkan selama bertahun-tahun oleh orang-orang yang berbeda, hal ini dapat menyebabkan ketidakpahaman atau perubahan tujuan pembuatan perangkat lunak;
 - tantangan heterogenitas dimana perangkat lunak harus dapat beradaptasi dengan teknologi yang terus berkembang dengan semakin luasnya lingkungan distribusi perangkat lunak;
 - tantangan pengiriman (*delivery*) di mana perangkat lunak dengan skala besar dan kompleks sekalipun dapat sampai ke tangan pelanggan (*customer*) atau pengguna (*user*) dengan cepat dan kualitas tetap terjaga.



Tes Formatif 2

Pilihlah satu jawaban yang paling tepat!

- Berikut yang bukan merupakan kriteria perangkat lunak yang dibuat melalui rekayasa perangkat lunak adalah
 - maintainability*
 - dependability*
 - inefficient*
 - robust*
- Yang dimaksud dengan *maintainability* adalah
 - dapat diandalkan terkait dengan proses bisnis yang dijalankan dan perubahan yang terjadi
 - efisien dari segi sumber daya dan penggunaan
 - kemampuan untuk digunakan sesuai dengan kebutuhan
 - dapat terus dipelihara seiring berkembangnya teknologi dan lingkungan
- Yang dimaksud dengan *usability* adalah
 - dapat diandalkan terkait dengan proses bisnis yang dijalankan dan perubahan yang terjadi
 - efisien dari segi sumber daya dan penggunaan
 - kemampuan untuk digunakan sesuai dengan kebutuhan
 - dapat terus dipelihara seiring berkembangnya teknologi dan lingkungan
- Yang dimaksud dengan *dependability and robust* adalah
 - dapat diandalkan terkait dengan proses bisnis yang dijalankan dan perubahan yang terjadi
 - efisien dari segi sumber daya dan penggunaan

- C. kemampuan untuk digunakan sesuai dengan kebutuhan
 - D. dapat terus dipelihara seiring berkembangnya teknologi dan lingkungan
- 5) Fase pendefinisian adalah seperti di bawah ini, *kecuali*
- A. fokus pada "*what*"
 - B. mencari tahu atau mengidentifikasi informasi yang perlu diproses
 - C. fokus pada "*how*"
 - D. mengidentifikasi performansi dan fungsi seperti apa yang diinginkan
- 6) Fase pengembangan adalah seperti di bawah ini, *kecuali*
- A. fokus pada "*how*"
 - B. *software engineer* berusaha untuk mendefinisikan data dan fungsi-fungsi diimplementasikan
 - C. mengidentifikasi performansi dan fungsi seperti apa yang diinginkan
 - D. perekayasa perangkat lunak berusaha mentranslasikan desain ke bahasa pemrograman.
- 7) Apa yang dimaksud dengan tantangan warisan dalam proses rekayasa perangkat lunak?
- A. Tantangan dimana perangkat lunak dikembangkan selama bertahun-tahun oleh orang-orang yang berbeda-beda.
 - B. Tantangan dimana perangkat lunak dapat beradaptasi terhadap perkembangan teknologi dan lingkungan.
 - C. Tantangan dimana perangkat lunak dengan skala besar tetap dapat sampai ke tangan pelanggan atau pengguna (*user*).
 - D. Tantangan dimana perangkat lunak tidak dapat digunakan oleh *customer* ataupun *user*.
- 8) Apa yang dimaksud dengan tantangan heterogenitas dalam proses rekayasa perangkat lunak?
- A. Tantangan dimana perangkat lunak dikembangkan selama bertahun-tahun oleh orang-orang yang berbeda-beda.
 - B. Tantangan dimana perangkat lunak dapat beradaptasi terhadap perkembangan teknologi dan lingkungan.
 - C. Tantangan dimana perangkat lunak dengan skala besar tetap dapat sampai ke tangan pelanggan atau pengguna (*user*).
 - D. Tantangan dimana perangkat lunak tidak dapat digunakan oleh *customer* ataupun *user*.

- 9) Apa yang dimaksud dengan tantangan pengiriman (*delivery*) dalam proses rekayasa perangkat lunak?
- A. Tantangan dimana perangkat lunak dikembangkan selama bertahun-tahun oleh orang-orang yang berbeda-beda.
 - B. Tantangan dimana perangkat lunak dapat beradaptasi terhadap perkembangan teknologi dan lingkungan.
 - C. Tantangan dimana perangkat lunak dengan skala besar tetap dapat sampai ke tangan pelanggan atau pengguna (*user*).
 - D. Tantangan dimana perangkat lunak tidak dapat digunakan oleh *customer* ataupun *user*.
- 10) Perangkat lunak harus dapat memiliki hal-hal berikut, *kecuali*
- A. memiliki nilai ekonomis
 - B. memiliki manfaat dan dapat digunakan oleh *user*
 - C. memenuhi spesifikasi *customer*
 - D. memenuhi spesifikasi *developer* saja

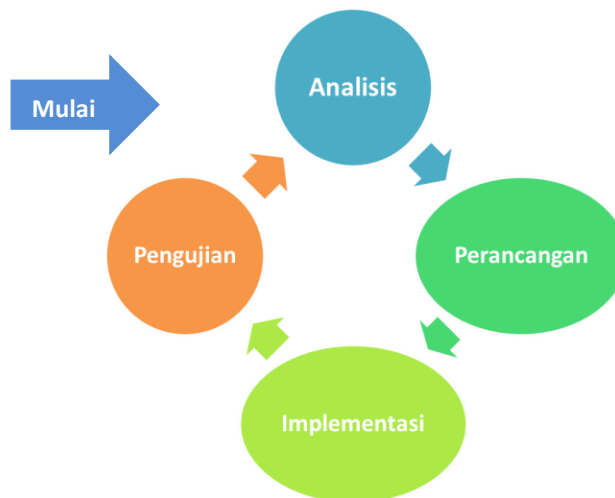
Cocokkanlah jawaban Anda dengan Kunci Jawaban Tes Formatif 2 yang terdapat di bagian akhir modul ini. Hitunglah jawaban yang benar. Kemudian, gunakan rumus berikut untuk mengetahui tingkat penguasaan Anda terhadap materi Kegiatan Belajar 2.



Apabila mencapai tingkat penguasaan 80% atau lebih, Anda dapat meneruskan dengan Kegiatan Belajar 3. **Bagus!** Jika masih di bawah 80%, Anda harus mengulangi materi Kegiatan Belajar 2, terutama bagian yang belum dikuasai.

Proses Rekayasa Perangkat Lunak

Proses rekayasa perangkat lunak dilakukan selama pembangunan perangkat lunak. Proses-proses yang dilakukan dalam rekayasa perangkat lunak secara garis besar adalah sebagai berikut.



Gambar 1.8
Tahapan Umum Rekayasa Perangkat Lunak

Proses-proses pada Gambar 1.8 di atas dapat dilakukan berulang kali sampai perangkat lunak memenuhi kebutuhan pelanggan atau *user*. Untuk membangun perangkat lunak yang benar-benar baik maka diperlukan tahapan-tahapan rekayasa perangkat lunak. Perangkat lunak yang dibangun tanpa melalui tahapan rekayasa perangkat lunak yang baik maka akan seperti membuat baju tanpa pola dan rencana model baju yang baik. Dalam membuat sebuah baju, maka dibutuhkan adanya perencanaan model baju dan pembuatan pola baju yang baik. Setelah pola baju terbentuk maka kain mulai dipotong sesuai pola, baru kemudian dijahit. Jika baju dibuat tanpa pola, maka yang terjadi adalah baju hasil tambal sulam. Seperti halnya pembuatan perangkat lunak tanpa melalui proses rekayasa perangkat lunak, maka hanya akan menghasilkan perangkat lunak "tambal sulam", begitu kurang maka akan ditambal. Tentu saja hasil perangkat lunak "tambal sulam" tidak akan bagus. Contoh lain misalnya

sebuah bangunan tanpa rencana yang matang, maka yang terjadi adalah bangunan "tambal sulam" yang akhirnya dirobohkan atau bahkan roboh karena pembuatannya tanpa perencanaan yang matang. Misalkan awalnya dibuat tanpa jendela, lalu butuh jendela, maka ditambahkan jendela. Jika butuh tambahan lantai ke atas, maka ditambahkan lantai, tanpa melihat kekuatan lantai di bawahnya. Maka yang terjadi adalah bangunan yang ambruk jika tidak mampu menahan kebutuhan tanpa perencanaan yang baik. Berikut adalah ilustrasi celana yang dibuat tanpa perencanaan yang baik. Apakah Anda mau memakainya? Dari segi bentuk memang masih celana, tapi tidak enak untuk dipakai. Seperti halnya perangkat lunak, bentuknya akan tetap perangkat lunak, tapi tanpa rekayasa yang baik, maka tidak akan enak untuk digunakan.



Gambar 1.9

Celana Tanpa Rekayasa Tidak Akan Menjadi Celana yang Berkualitas untuk Dipakai, Maukah Anda Memakainya?

Perkembangan teknologi informasi saat ini sangat mempermudah dan mempercepat proses rekayasa perangkat lunak dengan adanya bermacam-macam *tools* desain, *tools* pengembangan seperti IDE (*Integrated Development Environment*), *application framework*, dan lain-lain. Hal tersebut sering disalahartikan oleh pengembang perangkat lunak pemula dan pelanggan/pengguna perangkat lunak bahwa pengembangan perangkat lunak bisa diselesaikan dalam satu malam.

Anggapan tersebut sangat tidak tepat. Rekayasa perangkat lunak terdiri dari beberapa kegiatan yang harus dilakukan. Jika tahapan-tahapan tersebut tidak dilalui dengan baik, hampir bisa dipastikan perangkat lunak yang dihasilkan tidak akan mempunyai kualitas yang baik. Jadi tidak ada perangkat lunak yang bisa dihasilkan dengan baik hanya dalam satu malam seperti mitos pembuatan candi Prambanan atau mitos pembuatan Gunung Tangkuban Perahu.

Proses perangkat lunak (*software process*) adalah sekumpulan aktivitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri atas berikut ini.

1. Pengumpulan Spesifikasi (*Specification*)

Mengetahui apa saja yang harus dapat dikerjakan sistem perangkat lunak dan batasan pengembangan perangkat lunak.

2. Pengembangan (*Development*)

Pengembangan perangkat lunak untuk menghasilkan perangkat lunak yang sesuai dengan spesifikasi yang telah didefinisikan sebelumnya.

3. Validasi (*Validation*)

Memeriksa apakah perangkat lunak sudah memenuhi keinginan pelanggan (*customer*) dan kebutuhan pengguna (*user*).

4. Evolusi (*Evolution*)

Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*customer*) ataupun pengguna (*user*).

■ **TEKNOLOGI INFORMASI SOSIAL**

Teknologi informasi sosial atau *social information technology (social IT)* adalah faktor atau aspek sosial yang berkaitan dengan implementasi suatu teknologi informasi. Sebuah produk perangkat lunak tidak hanya berfokus di masalah teknologi dan teknis. Sering sebuah produk perangkat lunak hanya dibuat berdasarkan sudut pandang pengembang perangkat lunak yang memiliki kecenderungan berpola pikir teknis.

Sebuah perangkat lunak dianggap berkualitas jika memenuhi kebutuhan pelanggan (*customer*) dan sesuai keinginan pelanggan (*customer*). Sering kendala dari pengembangan perangkat lunak bukan berada pada masalah teknis (teknologi perangkat lunak dan perangkat keras) tapi pada kondisi lingkungan pelanggan, misalnya ternyata yang akan menjadi *user* atau pengguna perangkat lunak belum bisa mengoperasikan komputer, atau keengganan *user* untuk mengubah kebiasaannya bekerja menggunakan perangkat lunak yang baru atau dari manual menjadi menggunakan perangkat lunak, lingkungan yang tidak mendukung disiplin (misalnya untuk aplikasi keuangan) maka dari itu perlu adanya analisis terlebih dahulu di lingkungan pelanggan apakah dapat dijalankan perangkat lunak yang akan dikembangkan, karena sebaik apa pun perangkat lunak jika tidak mampu digunakan di lingkungan pelanggan maka perangkat lunak itu hanya akan menjadi “artefak” yang disimpan di gudang atau bahkan dibuang.

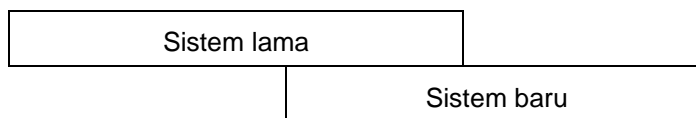
Hal-hal yang harus dilakukan sebelum mengembangkan perangkat lunak di lingkungan tertentu maka harus dicari tahu:

1. pengetahuan lingkungan tentang teknologi informasi dan komputer;
2. “*social knowledge*” atau “*local knowledge*” (pengetahuan mengenai budaya lokal) di lingkungan yang akan dikembangkan perangkat lunak, apakah memungkinkan untuk dikembangkan perangkat lunak;
3. pengetahuan tentang apa saja yang bisa dibatasi dan yang tidak, sehingga saat pengembangan perangkat lunak dapat mendefinisikan aturan main dari perangkat lunak.

Setelah perangkat lunak dikembangkan tetap masih diperlukan adanya sosialisasi perangkat lunak dengan mengadakan pelatihan secara bertahap, karena mengubah kebiasaan sebuah lingkungan ke sesuatu yang baru tidaklah gampang dan yang akan menggunakan perangkat lunak memiliki karakter yang berbeda-beda. Memang tidak harus menuruti semua karakter, tapi setidaknya dapat dicari titik tengah yang dapat diterima semua pihak. Komunikasi yang baik antara pengembang dan pelanggan atau *user* sangat dibutuhkan agar terjalin kerja sama yang baik dan saling menguntungkan.

Melakukan konversi dari cara kerja yang lama ke cara kerja yang baru menggunakan perangkat lunak yang dikembangkan perlu dilakukan secara bertahap, karena perubahan yang ekstrem akan menghabiskan lebih banyak sumber daya, dana, dan waktu. Beberapa cara konversi adalah sebagai berikut.

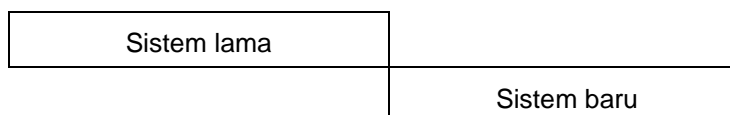
1. Konversi Paralel



Gambar 1.10
Ilustrasi Konversi Paralel

Konversi paralel dilakukan dengan melakukan beberapa waktu transisi dimana ada waktu dimana kedua sistem (sistem lama dan sistem baru) berjalan bersama untuk keperluan transisi sampai sistem baru dapat berjalan mandiri. Sumber daya yang dibutuhkan pada konversi paralel akan banyak terkuras pada waktu transisi.

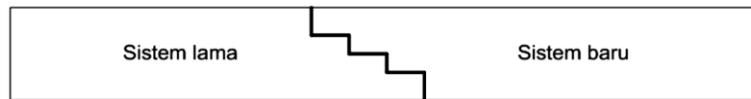
2. Konversi Langsung



Gambar 1.11
Ilustrasi Konversi Langsung

Konversi langsung dilakukan dimana sistem lama secara ekstrem langsung diganti dengan sistem yang baru. Konversi ini akan mengalami waktu yang sangat sulit di awal berjalannya sistem baru.

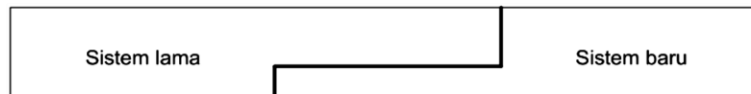
3. Konversi Per Fase



Gambar 1.12
Ilustrasi Konversi Per Fase

Konversi per fase dilakukan dengan berpindah per fase dari sistem lama ke sistem baru misalkan pada awal konversi hanya pada pekerjaan memasukkan data-data saja, pada tahap berikutnya mulai menggunakan proses perhitungan, lalu fase berikutnya mulai menggunakan proses pelaporan sistem baru, dan seterusnya (lebih fokus pada per fungsi sistem).

4. Konversi Pilot atau *Single Location*



Gambar 1.13
Ilustrasi Konversi Pilot

Konversi pilot dilakukan dengan melakukan konversi per unit kerja atau per lokasi di dalam sebuah lingkungan kerja. Misalnya pada tahap awal unit kerja yang sistemnya berubah adalah bagian keuangan, berikutnya pada bagian sumber daya manusia, dan seterusnya.

Pengembangan perangkat lunak yang berhasil dan dapat diterima dengan baik tidak hanya memperhatikan masalah teknis, tapi memperhatikan masalah non teknis seperti permasalahan sosial dan mencermati isu-isu yang sedang berkembang di masyarakat.



Latihan

Untuk memperdalam pemahaman Anda mengenai materi di atas, kerjakanlah latihan berikut!

- 1) Sebutkan definisi atau pengertian dari istilah-istilah sebagai berikut.
 - a. *Software Process*.
 - b. *Social IT*.
 - c. *Social Knowledge*.
- 2) Mengapa faktor sosial dari teknologi informasi sering sekali diabaikan oleh pengembang aplikasi?
- 3) Mengapa faktor sosial dari teknologi informasi perlu untuk diperhatikan?
- 4) Sebutkan kelebihan dan kekurangan masing-masing metode konversi sistem!

Petunjuk Jawaban Latihan

- 1) Pelajari tentang istilah-istilah rekayasa perangkat lunak pada Modul 1 Kegiatan Belajar 3.
- 2) Pelajari tentang faktor sosial dari teknologi informasi pada Modul 1 Kegiatan Belajar 3.
- 3) Pelajari tentang faktor sosial dari teknologi informasi pada Modul 1 Kegiatan Belajar 3.
- 4) Pelajari tentang metode konversi sistem pada Modul 1 Kegiatan Belajar 3.



Rangkuman

Untuk membangun perangkat lunak yang benar-benar baik maka diperlukan tahapan-tahapan rekayasa perangkat lunak. Perangkat lunak yang dibangun tanpa melalui tahapan rekayasa perangkat lunak yang baik maka akan seperti membuat baju tanpa pola dan rencana model baju yang baik.

Proses perangkat lunak (*software process*) adalah sekumpulan aktivitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri atas berikut ini.

1. Pengumpulan Spesifikasi (*Specification*).
2. Pengembangan (*Development*).
3. Validasi (*Validation*).
4. Evolusi (*Evolution*).

Teknologi informasi sosial atau *social information technology (social IT)* adalah faktor atau aspek sosial yang berkaitan dengan implementasi suatu teknologi informasi.

Melakukan konversi dari cara kerja yang lama ke cara kerja yang baru menggunakan perangkat lunak yang dikembangkan perlu dilakukan secara bertahap, karena perubahan yang ekstrem akan menghabiskan lebih banyak sumber daya, dana, dan waktu. Beberapa cara konversi adalah sebagai berikut.

1. Konversi paralel;
2. Konversi langsung;
3. Konversi per fase;
4. Konversi pilot atau *single location*.



Tes Formatif 3

Pilihlah satu jawaban yang paling tepat!

- 1) Berikut ini yang bukan merupakan proses umum rekayasa perangkat lunak adalah
 - A. pengumpulan spesifikasi (*specification*)
 - B. validasi (*validation*)
 - C. perekayasa kembali produk perangkat lunak (*reengineering*)
 - D. pengembangan (*development*)
- 2) Proses pengumpulan spesifikasi (*specification*) pada proses rekayasa perangkat lunak adalah
 - A. pengembangan perangkat lunak untuk menghasilkan perangkat lunak yang sesuai dengan spesifikasi
 - B. proses untuk memeriksa apakah perangkat lunak sudah memenuhi keinginan *customer* dan kebutuhan *user*
 - C. mengetahui spesifikasi apa saja yang harus dapat dikerjakan perangkat lunak dan batasan pengembangan perangkat lunak
 - D. proses perubahan perangkat lunak untuk memenuhi perubahan yang kebutuhan pelanggan (*customer*) ataupun pengguna (*user*)
- 3) Proses pengumpulan pengembangan (*development*) pada proses rekayasa perangkat lunak adalah
 - A. pengembangan perangkat lunak untuk menghasilkan perangkat lunak yang sesuai dengan spesifikasi
 - B. proses untuk memeriksa apakah perangkat lunak sudah memenuhi keinginan *customer* dan kebutuhan *user*
 - C. mengetahui spesifikasi apa saja yang harus dapat dikerjakan perangkat lunak dan batasan pengembangan perangkat lunak
 - D. proses perubahan perangkat lunak untuk memenuhi perubahan yang kebutuhan pelanggan (*customer*) ataupun pengguna (*user*)

- 4) Proses pengumpulan evolusi (*evolution*) pada proses rekayasa perangkat lunak adalah
- A. pengembangan perangkat lunak untuk menghasilkan perangkat lunak yang sesuai dengan spesifikasi
 - B. proses untuk memeriksa apakah perangkat lunak sudah memenuhi keinginan *customer* dan kebutuhan *user*
 - C. mengetahui spesifikasi apa saja yang harus dapat dikerjakan perangkat lunak dan batasan pengembangan perangkat lunak
 - D. proses perubahan perangkat lunak untuk memenuhi perubahan yang kebutuhan pelanggan (*customer*) ataupun pengguna (*user*)
- 5) Proses pengumpulan validasi (*validation*) pada proses rekayasa perangkat lunak adalah
- A. pengembangan perangkat lunak untuk menghasilkan perangkat lunak yang sesuai dengan spesifikasi
 - B. proses untuk memeriksa apakah perangkat lunak sudah memenuhi keinginan *customer* dan kebutuhan *user*
 - C. mengetahui spesifikasi apa saja yang harus dapat dikerjakan perangkat lunak dan batasan pengembangan perangkat lunak
 - D. proses perubahan perangkat lunak untuk memenuhi perubahan yang kebutuhan pelanggan (*customer*) ataupun pengguna (*user*)
- 6) Proses konversi paralel adalah proses
- A. konversi di mana ada waktu sistem lama dan sistem baru berjalan bersamaan
 - B. konversi yang dilakukan per fase untuk berpindah dari sistem lama ke sistem baru
 - C. konversi per unit kerja atau per lokasi
 - D. secara langsung berganti antara sistem lama ke sistem baru
- 7) Proses konversi per fase adalah proses
- A. konversi di mana ada waktu sistem lama dan sistem baru berjalan bersamaan
 - B. konversi yang dilakukan per fase untuk berpindah dari sistem lama ke sistem baru
 - C. konversi per unit kerja atau per lokasi
 - D. secara langsung berganti antara sistem lama ke sistem baru

- 8) Proses konversi langsung adalah proses
- A. konversi di mana ada waktu sistem lama dan sistem baru berjalan bersamaan
 - B. konversi yang dilakukan per fase untuk berpindah dari sistem lama ke sistem baru
 - C. konversi per unit kerja atau per lokasi
 - D. secara langsung berganti antara sistem lama ke sistem baru
- 9) Proses konversi pilot adalah proses
- A. konversi di mana ada waktu sistem lama dan sistem baru berjalan bersamaan
 - B. konversi yang dilakukan per fase untuk berpindah dari sistem lama ke sistem baru
 - C. konversi per unit kerja atau per lokasi
 - D. secara langsung berganti antara sistem lama ke sistem baru
- 10) Hal-hal yang perlu diketahui sebelum mengembangkan perangkat lunak adalah sebagai berikut, *kecuali* pengetahuan
- A. akan lingkungan teknologi informasi dan komputer
 - B. mengenai *social knowledge*
 - C. mengenai batasan perangkat lunak yang akan dikembangkan
 - D. mengenai pribadi *customer* maupun *user*

Cocokkanlah jawaban Anda dengan Kunci Jawaban Tes Formatif 3 yang terdapat di bagian akhir modul ini. Hitunglah jawaban yang benar. Kemudian, gunakan rumus berikut untuk mengetahui tingkat penguasaan Anda terhadap materi Kegiatan Belajar 3.

Tingkat Penguasaan =

$$\frac{\text{Jumlah Jawaban yang Benar}}{\text{Jumlah Soal}} \times 100$$

Arti tingkat penguasaan



Apabila mencapai tingkat penguasaan 80% atau lebih, Anda dapat meneruskan ke modul selanjutnya. **Bagus!** Jika masih di bawah 80%, Anda harus mengulangi materi Kegiatan Belajar 3, terutama bagian yang belum dikuasai.

Kunci Jawaban Tes Formatif

Tes Formatif 1

- 1) B
- 2) C
- 3) A
- 4) C
- 5) A
- 6) D
- 7) C
- 8) B
- 9) B
- 10) A

Tes Formatif 2

- 1) C
- 2) D
- 3) C
- 4) A
- 5) C
- 6) C
- 7) A
- 8) B
- 9) C
- 10) D

Tes Formatif 3

- 1) C
- 2) C
- 3) A
- 4) D
- 5) B
- 6) A
- 7) B
- 8) D
- 9) C
- 10) D

Glosarium

- Algoritma** : Langkah logis yang digunakan untuk menyelesaikan suatu masalah.
- Browser** : Perangkat lunak yang digunakan mengambil, menyajikan, dan melintasi berbagai sumber informasi dari internet.
- E-commerce** : Suatu proses berbisnis dengan menggunakan teknologi elektronik yang menghubungkan antara perusahaan, konsumen, dan masyarakat dalam bentuk transaksi elektronik atau internet.
- Link** : Sambungan yang merujuk dari satu halaman ke halaman lainnya untuk mempermudah akses.
- Open source** : Perangkat lunak membuka atau membebaskan *source code*-nya.
- PC** : Singkatan dari *personal computer* merupakan sebuah perangkat komputer yang digunakan oleh perorangan.
- Stakeholder** : Segenap pihak yang terkait dalam pengembangan perangkat lunak.
- Tools** : Alat yang digunakan untuk dalam membantu pengembangan perangkat lunak, alat tersebut bisa berupa *software* maupun *hardware*.
- Web** : Halaman dokumen yang ditulis dalam format HTML yang berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi.

Daftar Pustaka

Pressman, R.S. (2001). *Software engineering: A practitioner's approach*. fifth edition. New York: Mc Graw Hill.

Sommerville, I. (2016). *Software engineering: 10th Edition*. New York: McGraw-Hill.

Sukamto, R.A. (2018). *Rekayasa perangkat lunak terstruktur dan berorientasi objek*. Bandung: Penerbit Informatika.