

REKAYASA NERSADA RANGKAT NAK

UKTUR dan BERORIENTASI OBJEK

Bahasan meliputi:

- Analisis dan Desain Sistem
 - SDLC =
 - Basis Data
 - Pemrograman Terstruktur
- Pemrograman Berorientasi Objek I
- Analisis dan Desain Berorientasi Objek I
 - Pemodelan dan UML
 - Studi Kasus UML
 - Perancangan Pola Berorientasi Objek
 - Manajemen Proyek Perangkat Lunak

Penerbit INFORMATIKA

Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek

Penyusun

: Rosa A:S.

M. Shalahuddin

Penerbit

: Informatika Bandung

Pemasaran

: BI-Obses

Pasar Buku Palasari No. 82

Bandung 40264 Telp.(022)7317812 Fax. (022)7317896

Cetakan Ketiga

: September 2015

ISBN

: 978-602-1514-05-4

Copyright © 2015 pada Penerbit INFORMATIKA Bandung

1.1 Perangkat Lunak

perangkat lunak (software) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (user manual). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak (software). Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak. 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 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 pemakai perangkat lunak adalah orang yang memiliki kepentingan untuk memakai atau menggunakan perangkat lunak untuk memudahkan pekerjaannya.

Karakter perangkat lunak adalah sebagai berikut:

- Perangkat lunak dibangun dengan rekayasa (software engineering) bukan diproduksi secara manufaktur atau pabrikan.
- Perangkat lunak tidak pernah usang ("wear out") karena kecacatan dalam perangkat lunak dapat diperbaiki.
- 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:

Perangkat lunak sistem (system software)
 adalah kumpulan program dalam hal ini program yang satu
 ditulis untuk memenuhi kebutuhan program lainnya.

Rekayasa Perangkat Lunak

- 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 respon yang diinginkan.
- Perangkat lunak bisnis (business software)
 merupakan perangkat lunak pengelola informasi bisnis
 (seperti akuntansi, penjualan, pembayaran, penyimpanan
 (inventory)).
- 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.
- 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.
- Perangkat lunak komputer personal (personal computer software)
 merupakan perangkat lunak untuk PC misalnya perangkat lunak pemroses teks, pemroses grafik dan lain sebagainya.
- Perangkat lunak berbasis web (*web based software*) merupakan perangkat lunak yang dapat diakses dengan menggunakan *browser*.

perangkat lunak berintelijensia buatan (artificial intelligence software)
merupakan perangkat lunak yang menggunakan algoritma tertentu untuk mengelola data sehingga seakan-akan memiliki intelijensia seiring bertambahnya data yang diproses.

Produk perangkat lunak yang dibuat oleh pengembang (developer) perangkat lunak terdiri dari dua jenis:

- 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.
- 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.

1.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (software engineering) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Perangkat lunak banyak dibuat dan pada akhirnya sering tidak digunakan karena tidak memenuhi kebutuhan pelanggan atau bahkan karena masalah non-teknis seperti keenggan pemakai perangkat lunak (user) untuk mengubah cara kerja dari manual ke otomatis, atau ketidakmampuan user menggunakan komputer. Oleh karena itu,

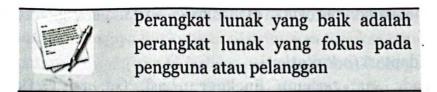
Rekayasa Perangkat Lunak

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). Adapun ilmu komputer lebih fokus pada teori dan konsep dasar perangkat komputer. Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria berikut:

- dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (maintainability)
- dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (dependability dan robust)
- efisien dari segi sumber daya dan penggunaan
- 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 (pemakai perangkat lunak) atau berorientasi pada pelanggan atau pemakai 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 perekayasa perangkat lunak (software engineer) berusaha untuk mendefinisikan bagaimana data distrukturkan dan bagaimana fungsi-fungsi yang dibutuhkan diimplementasikan didalam arsitektur perangkat lunak, bagaimana detail prosedural diimplementasikan, bagaimana karakter antarmuka tampilan, bagaimana desai 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 antara lain:

- Koreksi (correction)
 walaupun dengan jaminan kualitas yang terbaik, akan selalu
 ada kecacatan atau keinginan pelaggan (customer) yang tidak
 tertangani oleh perangkat lunak. Pemeliharaan dengan
 melakukan perbaikan terhadap kecacatan perangkat lunak.
- Adaptasi (adaptation)
 pada saat tertentu lingkungan asli (seperti CPU, sistem operasi, aturan bisnis, karakteristik produk luar) dimana perangkat lunak dikembangkan akan mengalami perubahan. Pemeliharaan adaptasi merupakan tahap untuk memodifikasi perangkat lunak guna mengakomodasi perubahan lingkungan luar dimana perangkat lunak dijalankan.
- Perbaikan (enhancement)
 sejalan dengan digunakannya perangkat lunak, maka
 pelanggan (customer) atau pemakainya (user) akan mengenali
 fungsi tambahan yang dapat mendatangkan manfaat.

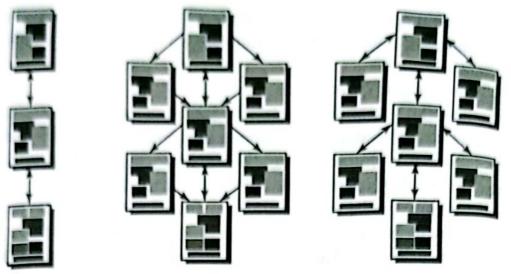
Pemeliharaan perfektif atau penyempurnaan melakukan ekstensi atau penambahan pada kebutuhan fungsional sebelumnya.

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 pemakainya (user). Untuk menanggulangi hal ini maka perangkat lunak harus dirancang dan dikondisikan untuk mengakomodasi perubahan kebutuhan yang diinginkan oleh pemakainya (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 sebagai berikut:

- 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 bahwa perangkat lunak dengan sekala besar dan kompleks sekalipun dapat sampai ke tangan pelanggan (customer) atau user dengan cepat dan kualitas tetap terjaga.

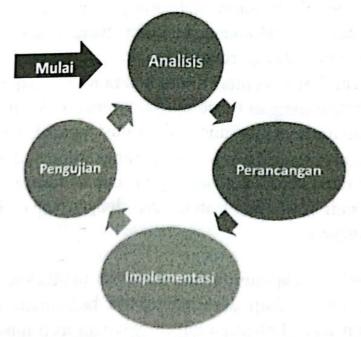
Ada suatu hal tambahan dalam rekayasa perangkat lunak yang berbasis web pada tahap desainnya yaitu bagaimana menentukan aliran halaman web. Beberapa aliran halaman web misalnya adalah sebagai berikut:



Gambar 1 Contoh aliran halaman web

1.3 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 2 Tahapan Umum Rekayasa Perangkat Lunak

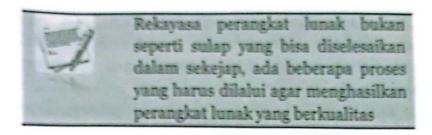
Proses-proses pada gambar di atas dapat dilakukan berulang kali sampai perangkat lunak memenuhi kebutuhan pelanggan atau user. Untuk membangun perangkat lunak yang benar-benar baik maka diperlukan tahapan-tahapan rekayasa perangkat lunak. Perangkat lunak yang dibangun tanpa melalui tahapan rekayasa perangkat 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 ang baik, maka tidak akan enak untuk digunakan.



Gambar 3 Celana tanpa rekayasa tidak akan menjadi celana yang berkualitas untuk dipakal, 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 tahapantahapan 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 aktifitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari:

- Pengumpulan Spesifikasi (Specification)
 mengetahui apa saja yang harus dapat dikerjakan sistem
 perangkat lunak dan batasan pengembangan perangkat lunak.
- Pengembangan (Development)
 pengembangan perangkat lunak untuk menghasilkan sistem
 perangkat lunak.
- Validasi (Validation)
 memeriksa apakah perangkat lunak sudah memenuhi
 keinginan pelanggan (customer).
- Evolusi (Evolution)
 mengubah perangkat lunak untuk memenuhi perubahan
 kebutuhan pelanggan (customer).

1.4 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 berkutat 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 pemakai perangkat lunak belum bisa mengoperasikan komputer, atau keengganan user untuk mengubah kebiasannya 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 apapun perangkat lunak jika tidak mampu digunakan di lingkungan pelanggan maka perangkat lunak itu hanya akan menjadi "artifak" yang disimpan di gudang atau bahkan dibuang.

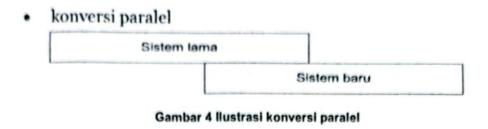
Hal-hal yang harus dilakukan sebelum mengembangkan perangkat lunak di lingkungan tertentu maka harus dicari tahu:

- pengetahuan lingkungan tentang teknologi informasi dan komputer
- "social knowledge" atau "local knowledge" (pengetahuan mengenai budaya lokal) di lingkungan yang akan dikembangkan perangkat lunak, apakah memungkinkan untuk dikembangkan perangkat lunak
- pengetahuan tentang apa saja yang bisa dibatasi dan yang tidak, sehingga saat pengembangan perangkat lunak dapat mendefinisikan aturan main dari perangkat lunak

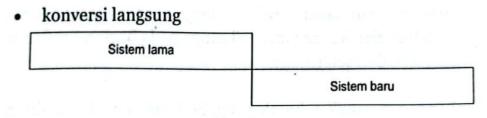
Setelah perangkat lunak dikembangkan tetap masih diperlukan adanya sosialisasi perangkat lunak dengan mengadakan pelatihan secara bertahap, karena mengubah kebiasaan sebuah lingkungan ke sesuatu yang baru tidaklah gampang dan yang akan menggunakan perangkat lunak memiliki karakter yang berbeda-beda. 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 ekstrim akan menghabiskan lebih banyak sumber daya, dana, dan waktu. Beberapa cara konversi adalah sebagai berikut:

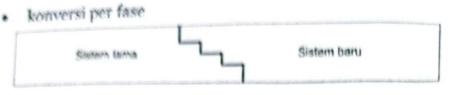


konversi paralel dilakukan dengan melakukan beberapa waktu transisi dimana ada waktu dimana kedua sistem (sistem lama dan sistem baru) berjalan bersama untuk keperluan transisi sampai sistem baru dapat berjalan mandiri. Sumber daya yang dibutuhkan pada konversi paralel akan banyak terkuras pada waktu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



Gambar 5 Ilustrasi konversi langsung

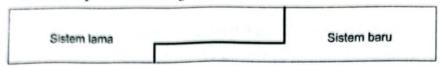
konversi langsung dilakukan karena sistem lama secara ekstrim langsung diganti dengan sistem yang baru. Konversi ini akan mengalami waktu yang sangat sulit di awal berjalannya sistem baru.



Gambar 6 flustrasi 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).

konversi pilot atau single location



Gambar 7 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 nonteknis seperti permasalahan sosial dan mencermati isu-isu yang sedang berkembang di masyarakat.