

BAB 1 Rekayasa Perangkat Lunak

Kecepatan proses dalam bisnis yang ada sekarang ini disebabkan karena kecanggihan teknologi perangkat lunak. Perangkat lunak semakin berkembang menyesuaikan kebutuhan manusia. Para pengembang berlomba – lomba menyajikan perangkat lunak yang memiliki efektifitas dan efisiensi dalam mendukung proses yang ada. Tentunya untuk mengembangkan perangkat lunak , kita perlu memahami terlebih dahulu karakteristik , proses dan praktek dari rekayasa perangkat lunak.

I.1. Karakteristik Perangkat Lunak

Pada saat ini , kita banyak sekali menemukan perangkat lunak yang memiliki banyak fungsi , contohnya saja kita lihat perangkat lunak pada telepon seluler , computer , laptop dan lain – lain. Perangkat lunak berperan dalam menyajikan informasi dan menjadi gerbang menuju jaringan informasi global.

Peningkatan perkembangan dari peran perangkat lunak ini didukung oleh teknologi perangkat keras serta perubahan – perubahan dari arsitektur komputer , memori yang mendukung kecepatan proses , penyimpanan yang besar , dan bermacam – macam alat input output yang dapat mendukung, sehingga semua sistem yang berbasis komputer dapat saling terhubung sehingga dapat saling berkomunikasi dan berbagi informasi.

Kecanggihan sistem tersebut , tentunya tidak lepas dari peran para pengembang perangkat lunak yang menjadi faktor dominan.

I.I.I. Definisi Perangkat Lunak

Banyak para pengguna perangkat lunak memahami fungsi dari perangkat lunak dan para ahli telah menjabarkan tentang deskripsi dari perangkat lunak sebagai berikut :

Menurut Ian Sommerville perangkat lunak tidak hanya mencakup program komputer saja, akan tetapi juga termasuk semua dokumentasi dan konfigurasi data yang berhubungan yang diperlukan untuk membuat program beroperasi dengan benar. (Sommerville, 2003)

Sedangkan menurut Pressman mengemukakan bahwa perangkat lunak adalah instruksi – instruksi dalam hal ini adalah program komputer yang ketika dijalankan menyediakan fitur – fitur , fungsi – fungsi dan kinerja – kinerja yang dikehendaki pengguna, dalam perangkat lunak terdapat struktur data yang memungkinkan program – program dapat memanipulasi informasi, dan informasi tersebut tercetak dalam bentuk maya yang menggambarkan pengoperasian dan penggunaan program – program tersebut. (Roger S. Pressman, 2012)

Dari definisi di atas dapat di simpulkan bahwa perangkat lunak adalah suatu program yang dapat beroperasi untuk memanipulasi informasi , yang di dalamnya terdapat instruksi – instruksi berupa fitur atau fungsi tertentu sehingga informasi yang telah di peroleh dapat tercetak dalam bentuk maya.

Namun untuk lebih jelasnya memahami tentang perangkat lunak , kita dapat mengidentifikasi dari karakteristiknya . Berikut adalah karakteristik – karakteristik dari perangkat lunak (Roger S. Pressman, 2012) :

1. Perangkat lunak dikembangkan atau direkayasa, tidak diproduksi dalam konteks manufaktur.
2. Perangkat lunak tidak mengalami kelelahan, namun juga kinerja dapat memburuk, untuk itu di perlukan perancangan perangkat lunak dengan cara yang lebih baik. Metode – metode rekayasa perangkat lunak pada dasarnya bertujuan untuk mengurangi simpangan angka kegagalan rekayasa perangkat lunak.
3. Meskipun industri terus beralih ke konstruksi berbasis komponen, sebagian besar perangkat lunak masih tetap dibuat berdasarkan spesifikasi yang di minta pengguna

Dengan melihat karateristik tersebut , ini memperlihatkan bahwa karakteristik dari perangkat lunak sangat berbeda dengan karakteristik perangkat keras.

I.1.2. Ranah Aplikasi Perangkat Lunak

Secara umum aplikasi perangkat lunak dibagi kedalam 7 (tujuh) kategori yaitu (Roger S. Pressman, 2012):

1. **Perangkat lunak sistem** , sekumpulan program yang ditulis untuk melayani program – program lain , yang melakukan pemrosesan struktur –

struktur informasi yang kompleks namun umumnya bersifat terbatas. Contoh dari perangkat lunak sistem ini adalah sistem operasi , driver, perangkat lunak jaringan , dll).

2. **Perangkat lunak aplikasi** , program – program mandiri yang menjawab kebutuhan bisnis secara terperinci dengan melakukan pemrosesan data bisnis atau data teknis yang mendukung berjalannya operasi – operasi bisnis.
3. **Perangkat lunak rekayasa** , aplikasi yang telah memiliki algoritma yang penuh dengan kalkulasi data numerik. Contohnya adalah aplikasi untuk kebutuhan vulkanologi , astronomi , dan lain – lain.
4. **Perangkat lunak yang tertanam**, perangkat lunak yang berada pada suatu produk atau sistem dan digunakan untuk menjalankan dan mengendalikan fitur – fitur atau fungsi – fungsi untuk user. Contohnya tampilan indikator pada peralatan elektronik.
5. **Perangkat lunak lini produk**, perangkat lunak yang dirancang untuk menyediakan kemampuan khusus untuk digunakan oleh pelanggan yang bersifat terbatas dan berkonsentrasi pada pasar tertentu . contohnya adalah pengolah kata, *spreadsheet*, dan lain – lain.
6. **Aplikasi web**, perangkat lunak yang berpusat pada jaringan komputer , berisi sekumpulan file *hypertext* yang saling terhubung untuk menunjukkan informasi – informasi tertentu,

berada lingkungan komputasi dan terintegrasi dengan sistem basis data.

7. **Perangkat lunak kecerdasan buatan** , perangkat lunak yang menggunakan algoritma non-numerik untuk memecahkan permasalahan – permasalahan rumit yang tidak dapat di selesaikan dengan komputasi atau analisis permasalahan langsung. Contohnya adalah sistem pakar , robotik, *game*, dll.

Para rekayasawan perangkat lunak bekerja keras untuk membuat sistem baru, mengoreksi aplikasi, mengadaptasi dan meningkatkan kemampuannya. Seiring dengan kerja keras tersebut karena adanya perkembangan teknologi maka muncul tantangan – tantangan baru yang harus di hadapi oleh para rekayasawan diantaranya adalah :

1. Pertumbuhannya jaringan nirkabel yang mengembangkan sistem dan perangkat lunak aplikasi untuk dapat berkomunikasi dengan jaringan yang sangat luas.
2. Kemudahan dalam pencarian sumber informasi melalui internet menantang para rekayasawan untuk membuat sebuah aplikasi yang sederhana yang menyediakan manfaat bagi pengguna akhir di seluruh dunia.
3. Berkembangnya *open source* menantang para rekayasawan untuk mengembangkan kode - kode dengan teknik – teknik yang

memungkinkan pelanggan dan pengembang tahu perubahan – perubahan yang telah dibuat.

Selain kategori perangkat lunak diatas , dikenal juga adanya perangkat lunak warisan yaitu perangkat lunak yang telah terlebih dahulu dikembangkan . Perangkat lunak warisan ini dicirikan dengan panjangnya usia dan kekritisannya bisnis. Perangkat lunak warisan ini bisa saja berkualitas buruk , perancangan yang tidak memungkinkan untuk diperluas lagi, kode – kode rumit, buruknya dokumentasi , pengujian dan hasilnya tidak pernah diarsipkan, namun perangkat lunak warisan ini mendukung fungsi – fungsi bisnis inti. Jika perangkat lunak warisan ini memenuhi kebutuhan penggunaannya dan beroperasi dengan baik tentunya tidak perlu diperbaiki. Namun, perangkat lunak warisan perlu diubah atau dikembangkan karena beberapa alasan (Roger S. Pressman, 2012) :

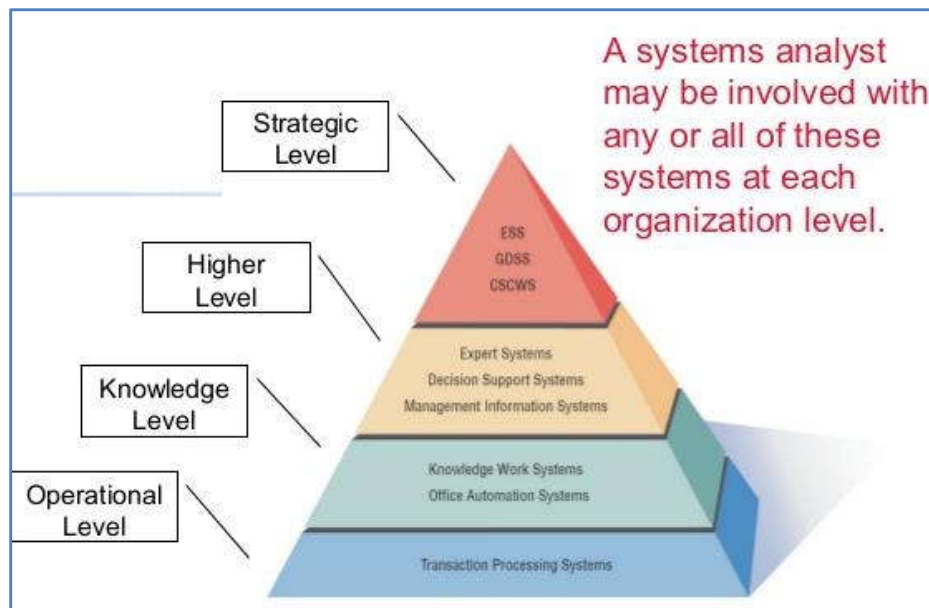
1. Perlu diadaptasikan sehingga memenuhi kebutuhan lingkungan atau teknologi komputasi yang baru.
2. Perlunya ditingkatkan kinerjanya supaya menjalankan kebutuhan – kebutuhan bisnis baru.
3. Perlu di perluas agar dapat saling berkomunikasi dengan perangkat lunak modern atau basis data modern.
4. Perlu dirancang ulang agar hidup dalam lingkungan jaringan komputer.

Dalam ranah perangkat lunak yang telah di jelaskan diatas , yang paling populer adalah aplikasi – aplikasi web. Menurut Powel aplikasi web pada dasarnya mempunyai sifat – sifat berikut (Roger S. Pressman, 2012) :

1. Kepadatan Jaringan , aplikasi web biasanya berada pada suatu jaringan komputer dan harus melayani client yang beragam.
2. Keserempakan, sejumlah besar pengguna mungkin akan mengakses aplikasi web secara serempak.
3. Jumlah pengguna akhir tidak dapat diprediksi.
4. Kinerja, jika aplikasi web yang dibuka tidak memiliki *performace* yang tinggi , maka pengguna akan meninggalkannya.
5. Ketersediaan, para pengguna menuntut akses 24 jam setiap harinya.
6. Digerakkan oleh data , menyajikan data baik itu teks, grafis, audio dan video.
7. Peka terhadap isi, kualitas dan karakter isi merupakan faktor penentu yang penting dari kualitas aplikasi.
8. Kesegaran, aplikasi web menunjukkan kedisiplinan waktu untuk merilis produk ke pasar dalam waktu beberapa hari saja.
9. Keamanan , karena aplikasi web tersedia melalui akses jaringan komputer , maka sulit membatasi jumlah pengguna akhir .
10. Estetika , daya tarik aplikasi web adalah tampilan dan nuansanya.

1.2. Tipe dari sistem atau perangkat lunak berdasarkan tujuan pengembangannya

Berdasarkan tujuan pengembangan sistem dengan mempertimbangkan kebutuhan manusia dan bisnis, sistem terbagi menjadi (Kenneth E. Kendall, 2010):



Gambar I.1 Tipe dari sistem berdasarkan tujuan pengembangannya

Sumber : (Kenneth E. Kendall, 2010)

1. *Transaction processing system* (TPS)
Merupakan sistem informasi terkomputerisasi yang dikembangkan dengan tujuan untuk memproses sejumlah data yang besar untuk transaksi bisnis yang rutin. Biasanya sistem ini digunakan untuk berinteraksi dengan lingkungan eksternal.
2. *Office Automation System* (OAS)
Merupakan sistem yang mendukung pekerjaan data, yang biasanya digunakan untuk menciptakan suatu

pengetahuan baru dengan menganalisis informasi , dalam analisis tersebut dapat dilakukan manipulasi data dengan cara – cara tertentu sebelum di bagikan secara formal atau di sebarluaskan ke seluruh organisasi.

3. *Knowledge work system*(KWS)
Merupakan sistem yang mendukung para pekerja profesional seperti ilmuwan , dokter dan lain – lain dalam menciptakan pengetahuan baru dan memungkinkan mereka berkontribusi pada organisasi atau masyarakat luas.
4. *Management Information system* (MIS)
Merupakan sistem informasi terkomputerisasi yang berfungsi karena interaksi manusia dan komputer. Dengan mengharuskan manusia, perangkat lunak dan perangkat keras berkerjasama untuk mendukung organisasi dalam melakukan pemrosesan transaksi , termasuk dalam analisis keputusan dan pengambilan keputusan.
5. *Decision support system* (DSS)
Merupakan sistem informasi terkomputerisasi level yang tinggi karena bergantung pada basis data yang merupakan sumber data , fungsi dari sistem ini adalah mendukung proses pengambilan keputusan sehingga sistem ini berfokus pada bisnis *intelligence*.
6. *Artificial Intelligence*(AI) dan *Expert System* (ES)
Expert system merupakan sistem berbasis pengetahuan yang secara efektif menangkap dan menggunakan pengetahuan para ahli untuk memecahkan masalah tertentu yang dialami suatu organisasi. *Expert system* menggunakan penalaran *Artificial Intelligence*.
7. *Group decision support system* (GDSS) dan *Computer-Supported collaborative work system* (CSCWS)

Merupakan sistem informasi yang digunakan secara bersama – sama untuk suatu kelompok , digunakan untuk mendukung proses pengambilan keputusan dalam suatu organisasi.

8. *Executive Support System*

Merupakan sistem informasi yang mendukung eksekutif mengatur interaksi mereka dengan lingkungan eksternal. Sistem ini menyediakan dukungan grafis dan komunikasi di tempat yang dapat diakses dari kantor atau ruang pribadi. Sistem ini mengandalkan informasi dari TPS dan MIS , sistem ini memperluas dan mendukung kemampuan para eksekutif sehingga memungkinkan mereka untuk memahami lingkungan mereka.

I.3. Rekayasa Perangkat Lunak

Untuk membangun perangkat lunak , sebagai rekayasawan harus memahami bahwa perangkat lunak kini telah menyatu dalam kehidupan banyak orang, maka seorang rekayasawan harus memahami masalah terlebih dahulu sebelum memberikan solusi terhadap perangkat lunak yang akan dikembangkan. Perancangan menjadi sesuatu yang sangat penting agar menghasilkan perangkat lunak yang berkualitas tinggi . Perangkat lunak berkualitas tinggi adalah perangkat lunak yang memiliki nilai bisnis bagi pengguna , berusia panjang dan tentunya perangkat lunak tersebut harus bersifat dapat dipelihara atau dirawat. Dengan melihat kenyataan tersebut maka perangkat lunak dalam segala bentuk kategorinya membutuhkan untuk di rekayasa.

Menurut Rosa rekayasa perangkat lunak merupakan pembangunan dengan menggunakan prinsip atau konsep

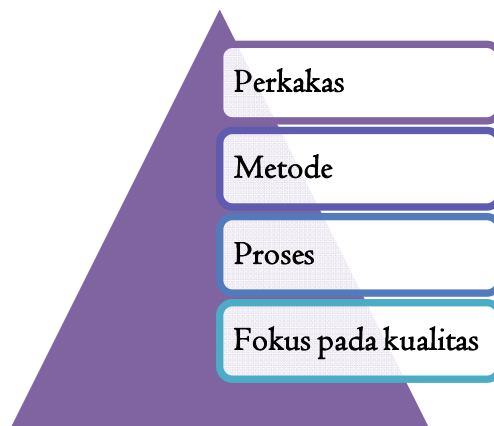
rekayasa dengan tujuan untuk menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. (Rosa A. S, 2018).

IEEE mengembangkan definisi rekayasa perangkat lunak yaitu suatu aplikasi dari pendekatan yang sistematis, disiplin dan dapat diukur pada pengembangan , operasi dan perawatan lunak. (Roger S. Pressman, 2012).

Sedangkan menurut Ian mengemukakan bahwa rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak , mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan. (Sommerville, 2003).

Jadi dapat disimpulkan bahwa rekayasa perangkat lunak adalah proses produksi perangkat lunak yang menggunakan pendekatan sistematis , disiplin dan dapat diukur yang pada akhirnya menghasilkan perangkat lunak yang bekerja secara efisien .

Pressman menyatakan bahwa segala pendekatan rekayasa perangkat lunak harus bersandar pada komitmen organisasi pada peningkatan kualitas. Untuk lebih jelas berikut adalah gambaran lapisan – lapisan rekayasa perangkat lunak (Roger S. Pressman, 2012);



Gambar 1.2 Lapisan – lapisan rekayasa perangkat lunak

Pondasi dari tindakan yang berkaitan dengan rekayasa perangkat lunak adalah lapisan proses. Proses perangkat lunak mendefinisikan kerangka kerja yang harus dibangun, serta membentuk kendali manajemen proyek dan membangun konteks. Metode rekayasa perangkat lunak berisi tentang prosedur untuk mengembangkan perangkat lunak. Dalam prosedur tersebut terdapat *task* yang menyertakan komunikasi, analisis kebutuhan, pemodelan rancangan, pembuatan program, pengujian dan dukungan – dukungan untuk *user*. Perangkat rekayasa perangkat lunak menyajikan dukungan semiotomatis maupun otomatis bagi proses dan metode – metode.

I.4. Proses Perangkat Lunak

Proses merupakan sekumpulan aktivitas, aksi dan tugas yang dijalankan ketika suatu produk harus dibuat. Aktivitas pada dasarnya berupaya mencapai tujuan umum, aksi mencakup didalamnya sederetan tugas – tugas yang menghasilkan suatu produk kerja utama, tugas pada dasarnya berkonsentrasi pada sebuah tujuan yang kecil namun terdefinisi dengan baik.

Kerangka kerja proses membangun dasar bagi proses rekayasa perangkat lunak yang lengkap dengan cara mengidentifikasi sejumlah kecil aktivitas kerangka kerja yang cocok bagi semua proyek rekayasa perangkat lunak. Kerangka kerja proses pada rekayasa perangkat lunak terdiri atas lima aktivitas berikut (Roger S. Pressman, 2012) :

1. **Komunikasi**, bertujuan untuk memahami tujuan – tujuan stakeholder atas proyek perangkat lunak yang sedang dikembangkan dan mengumpulkan kebutuhan – kebutuhan yang akan membantu mendefinisikan fitur – fitur perangkat lunak berikut dengan fungsi – fungsinya.
2. **Perencanaan** , kegiatan perencanaan menciptakan suatu peta yang dapat membantu membimbing tim perangkat lunak. Rencana proyek perangkat lunak menggambarkan risiko – risiko yang mungkin muncul, sumber daya yang akan dibutuhkan, produk – produk kerja yang harus dihasilkan dan *schedule* kerja.
3. **Pemodelan**, dilakukan bertujuan untuk membuat sketsa sehingga tim perangkat lunak dapat memahami gambaran besar produk yang akan di buat.
4. **Konstruksi**, kegiatan yang menggabungkan pengkodean dan pengujian .
5. **Penyerahan perangkat lunak kepada user** , penyajian perangkat lunak kepada *user* untuk di evaluasi.

Lima aktivitas diatas dapat juga berulang , pengulangan tersebut menghasilkan sebuah peningkatan perangkat lunak yang memberikan *user* suatu fitur dan fungsionalitas perangkat lunak. Aktivitas – aktivitas kerangka kerja perangkat lunak

diasas disempurnakan dengan aktivitas – aktivitas yang bertindak sebagai penyangga, yaitu :

1. **Penelusuran dan kendali proyek perangkat lunak**, dilakukannya penilaian terhadap kemajuan proyek perangkat lunak yang berjalan sehingga dapat diambil tindakan – tindakan untuk membuat tepat waktu.
2. **Manajemen risiko**, menilai risiko – risiko yang dapat berpengaruh terhadap hasil akhir dan kualitas perangkat lunak
3. **Penjaminan kualitas perangkat lunak**, melakukan kegiatan – kegiatan yang memastikan kualitas perangkat lunak.
4. **Ulasan** , mengevaluasi produk untuk menemukan dan menyingkirkan *defect*.
5. **Pengukuran**, menjelaskan dan mengukur proses, proyek dan produk perangkat lunak yang akan membantu tim perangkat lunak dalam memenuhi kebutuhan stakeholder.
6. **Manajemen konfigurasi perangkat lunak**
7. **Manajemen penggunaan ulang**
8. **Persiapan produk kerja dan produksi**

I.5. Praktik Rekayasa Perangkat Lunak

Kegiatan kerangka kerja dari perangkat lunak membentuk kerangka arsitektur bagi pekerjaan perangkat lunak, namun bagaimana praktek implementasi dari rekayasa perangkat lunak yang baik. Untuk itu perlu di pahami konsep dan prinsip yang merujuk pada kegiatan kerangka kerja tersebut.

I.5.1. Esensi Praktik

Menurut George Poyla dalam (Roger S. Pressman, 2012) menjelaskan esensi dari praktik rekayasa perangkat lunak yaitu :

1. Pahami masalahnya
2. Rancang solusinya
3. Laksanakan rancangan
4. Periksa ketepatan hasilnya

I.5.2. Prinsip – prinsip Umum dari Praktek Rekayasa Perangkat Lunak.

Davis Hooker dalam (Roger S. Pressman, 2012) menyatakan ada tujuh prinsip pada praktik rekayasa perangkat lunak :

1. Alasan keberadaan perangkat lunak – memberikan nilai tertentu bagi para penggunanya.
2. Tetap sederhana , sehingga mendukung terbentuknya sistem yang lebih mudah dipahami dan di rawat.
3. Pertahankan visi, tanpa visi yang jelas, proyek perangkat lunak hampir pasti akan berakhir menjadi dua atau lebih pemikiran.
4. Apa yang anda buat, akan digunakan oleh konsumen, dengan mempermudah pekerjaan konsumen , maka akan menambahkan nilai tertentu pada perangkat lunak yang dibangun.
5. Membuka diri terhadap masa depan, perangkat lunak harus dapat beradaptasi dengan perubahan – perubahan sehingga dapat berumur panjang dan nilai lebih.

6. Rancanglah selangkah ke depan sehingga dapat digunakan kembali, perencanaan selangkah kedepan untuk penggunaan uang akan dapat mengurangi biaya dan akan meningkatkan nilai dari perangkat lunak yang dibangun.
7. Pikirlah, efek samping dari berfikir adalah belajar untuk mengenal ketika anda tidak mengetahui sesuatu yang disaat yang sama anda dapat meneliti jawabannya.

I.6. Penggunaan CASE TOOLS

Dalam proses pengembangan perangkat lunak , agar mendapatkan produktivitas yang tinggi , analis dapat menggunakan alat produktivitas yang disebut dengan Computer Aided Software Engineering (CASE), yang dibuat secara eksplisit untuk meningkatkan produktivitas, komunikasi yang lebih efektif dengan pengguna dan mengintegrasikan pekerjaan mereka dari awal sistem hingga akhir siklus sistem (Kenneth E. Kendall, 2010).

Salah satu contoh dari CASE tools adalah *Visible Analyst* yang digunakan untuk perancangan grafis , analisis dan desain untuk membangun aplikasi , database dan lain – lain. Contoh CASE tools adalah Microsoft office visio , star UML , Blasamiq Mockups , dan lain – lain