TUGAS KELOMPOK MANAJEMEN PROYEK

"SOFTWARE ENGINEERING"

Disusun oleh :

Djarot S.P.	I1A003041
Agus Sampurno	I1A005026
Eko Sapti B.B	I1A005060
M.Arief Affandy	I1A005066
Mufti Al Aziz	H1C007031
Besi Bestari Abror	H1C007035
Arly Fadli M.	H1C007052
Ika Rizka Annisa	H1C007058
David Setiawan	H1C007071
Yusran Adhitya K.	H1C007074

UNIVERSITAS JENDERAL SOEDIRMAN

FAKULTAS SAINS DAN TEKNIK

JURUSAN TEKNIK

PROGRAM STUDI TEKNIK ELEKTRO

2009

DAFTAR ISI

BAB I
PENDAHULUAN

Pada saat ini, perkembangan teknologi berubah dengan cepat, terutama dalam bidang elektronika, teknologi tersebut dapat berupa Software maupun Hardware. Saat ini perkembangan software sangat cepat sekali, Seiring dengan berkembangnya internet, Terjadi banyak perubahan dalam sistem pendistribusian perangkat lunak. Perangkat lunak yang dahulu biasa distribusikan melalui toko-toko penjual perangkat lunak, sekarang dapat diperoleh dengan men*download* file instalasinya secara langsung dari situs pembuat perangkat lunak, atau melalui situs-situs yang menyediakan berbagai perangkat lunak yang dapat di-*download*.

Implementasi suatu teknologi perangkat lunak pada suatu proses pengembangan perangkat lunak dan pada suatu produk perangkat lunak, melibatkan tahapan-tahapan kompleks yang memerlukan pembelajaran yang berkelanjutan, yang akhirnya dapat memberikan pengetahuan mengenai status dari suatu proses pembuatan perangkat lunak atau suatu produk dari perangkat lunak.

Dengan adanya evaluasi pada atribut-atribut yang ada dalam perangkat lunak, dapat diperoleh status dari suatu perangkat lunak tersebut. Dalam hal ini, situasi yang ada dapat diidentifikasi dan diklasifikasikan, sehingga dapat digunakan untuk membantu dalam mencari peluang-peluang baru yang bisa digunakan dalam pengembangan dan perbaikan perangkat lunak. Evaluasi seperti ini pada akhirnya dapat digunakan untuk membuat perencanaan dalam perubahan-perubahan yang mungkin perlu diimplementasikan di masa yang akan datang. Atribut-atribut yang diidentifikasi ini juga dapat digunakan sebagai referensi dan bahan pertimbangan bagi proses pengembangan perangkat lunak lainnya.

BAB II
PEMBAHASAN

2.1 Perangkat Lunak

Software engineering biasa disebut sebagai perangkat lunak, yaitu sebuah produk yang menyajikan potensi komputasi serta menghasilkan dan mengelola informasi. Selain itu, fungsi dari perangkat lunak adalah sebagai alat untuk menyajikan sebuah produk, diantaranya sebagai berikut :

1. Mengendalikan program lain misalkan system operasi.
2. Mempengaruhi komunikasi, seperti perangkat lunak jaringan.
3. Membantu membangun perangkat lunak yang lain misalnya tool perangkat lunak.

Perangkat lunak terdiri dari program dokumen dan data yang saling berhubungan antara satu dengan yang lainnya. Berdasarkan fungsinya ternyata perangkat lunak dengan hardware memiliki perbedaan, yaitu perangkat lunak dirakit, tidak kadaluarsa dan lebih kompleks dari pada hardware.

Aplikasi perangkat lunak dapat diklasifikasikan sebagai system software, application software, engineering software, embedded software, product line software, Web Applications, dan AI software.

Perangkat lunak di Kategorikan Baru sebagi berikut :

1. Obiquitous computing (jaringan nirkabel).
2. Netsourcing (Web sebagai mesin komputasi).
3. Open source (kode program dibuka gratis kepada komunitas komputasi).
4. Data mining, Grid computing, Cognitive machines, dan Software for nanotechnologies juga termasuk di dalamnya.

Perangkat lunak memiliki perubahan-perubahan di dalam kebutuhan maupun pembuatannya, perubahan tersebut harus di sesuaikan berdasarkan cara-cara berikut ini :

1. PL harus ber-adaptasi untuk memenuhi kebutuhan lingkungan komputasi atau teknologi baru.
2. PL harus di-perbaiki untuk mengimplementasi kebutuhan bisnis baru.
3. PL harus diperluas untuk membuatnya dapat berinteroperasi dengan sistem dan database modern yang lain.
4. PL harus di-arsitek ulang untuk membuatnya 'hidup' di tengah-tengah lingkungan jaringan**.**

Pada kenyataannya perangkat lunak memiliki perkembangan-perkembangan sesuai dengan kebutuhannya, oleh karena itu Perangkat lunak memiliki evolusi yang juga perlu di perhatikan, diantaranya :

1. The Law of Continuing Change (1974): Hukum Perubahan Berkelanjutan. Sistem Elektronik harus secara berkelanjutan beradaptasi jika tidak akan mengurangi tingkat kenyamanan.
2. The Law of Increasing Complexity (1974): Hukum Peningkatan Kompleksitas : sejalan dengan pertumbuhan PL, kompleksitasnya juga akan meningkat, kecuali ada sesuatu yang dilakukan untuk mengelola atau menguranginya.
3. The Law of Self Regulation (1974): Hukum Regulasi Mandiri, Proses evolusi sistem elektronik adalah regulasi mandiri dengan distribusi pengukuran produk dan proses yang dekat dengan normal.
4. The Law of Conservation of Organizational Stability (1980): Hukum Konservasi Stabilitas Organisasi. Rata-rata kecepatan aktivitas global efektif dalam sistem elektronik tidak jauh berbeda dengan kecepatan produksi.
5. The Law of Conservation of Familiarity (1980): Hukum Konservasi Kebiasaan Sistem elektronik turut mengembangkan segala sesuatu yang berkaitan dengannya, developer, sales, user, sebagai contoh, harus tetap menguasai isi dan perilaku untuk dapat menggapai evolusi yang nyaman.
6. The Law of Continuing Growth (1980): Hukum Keberlanjutan Pertumbuhan. Muatan fungsional sistem elektronik harus secara terus menerus meningkat untuk merawat kenyamanan pengguna.

7. The Law of Declining Quality (1996): Hukum Penurunan Kualitas, Kualitas sistem elektronik akan selalu menurun kecuali ada usaha perawatan dan beradaptasi pada perubahan lingkungan operasional.
8. The Feedback System Law (1996): Evolusi proses tipe elektronik membentuk sistem umpan balik multi-level, multi-loop, multi-agen, dan harus diperlakukan sebagaimana usaha-usaha serius lain untuk membuat peningkatan yang signifikan.
9. Mitos PL
10. Mempengaruhi manajer, pelanggan (dan stakeholder non teknis lainnya) serta para praktisi
11. Dipercaya karena mereka sering mempunyai bagian-bagian kebenaran.,

2.2 Praktek Rekayasa Perangkat Lunak

Praktek adalah sejumlah konsep, prinsip, metode dan tools yang harus dimiliki ketika software direncanakan dan dikembangkan. Praktek juga menunjukkan detail yang mempertimbangkan teknis dan praktis, yang berada di dalam proses perangkat lunak, sesuatu yang dibutuhkan untuk membangun perangkat lunak komputer berkualitas tinggi. Praktek memiliki beberapa esensi, Esensi Praktek tersebut tertuang dalam buku George Polya yang ditulis di tahun 1945 menggambarkan esensi dari praktek RPL, yang bertujuan untuk *Memahami permasalahan* (komunikasi dan analisis), *Merencanakan solusi* (pemodelan dan desain PL), *Melaksanakan rencana* (pembuatan kode), dan *Memeriksa akurasi hasil* (menguji dan QA).

Pada prinsipnya praktek yang baik adalah pemecahan masalah yang umum, secara inti RPL tersebut menyediakan nilai pada konsumen dan pengguna secara KIS (keep it simple!), berfungsi juga untuk Mengelola produk dan visi project. Pada intinya Apa yang dihasilkan, dapat dimanfaatkan oleh pihak lain dan Terbukalah pada masa depan.

Praktek-Praktek RPL digunakan untuk memahami bingkai kerja proses umum yang antara lain adalah komunikasi, perencanaan, pemodelan, konstruksi, dan deployment. Di sini kita akan mengidentifikasi prinsip-prinsip bagaimana memulai praktek dan mengerjakan sekelompok tugas yang bisa diperpendek.

Pada praktek komunikasi kita akan menggunakan beberapa prinsip mendasar yaitu mendengarkan dan mempersiapkan fasilitas komunikasi sebelum komunikasi. Disamping itu tatap muka merupakan prinsip yang terbaik, membuat keputusan dan catatan tertulis, kolaborasi dengan konsumen, tetap fokus membuat gambar ketika ada sesuatu yang tidak jelas, terus bergerak dan negosiasi dikatakan sukses ketika dua belah pihak menang.

Selanjutnya, pada praktek perencanaan kita juga memiliki beberapa hal-hal yang harus di lakukan antara lain memahami ruang lingkup proyek, melibatkan konsumen (dan stakeholder yang lain), mengenali bahwa perencanaan adalah iterative, melakukan estimasi berdasar apa yang anda ketahui. Selain hal-hal tersebut diatas, kita juag harus menyadari resiko yang akan terjadi, bertindak realistis, menyesuaikan hal-hal kecil yang tidak tertata ketika anda merencanakan, menentukan bagaimana kualitas dapat digapai, menentukan bagaimana anda dapat mengakomodasi perubahan, serta melacak apa yang telah anda rencanakan

Model dibuat untuk mendapatkan pemahaman yang lebih baik terhadap entitias aktual yang akan dibangun. Model Analisis menampilkan kebutuhan konsumen dengan melukiskan PL dalam tiga domain yang berbeda : domain informasi, domain fungsi, dan domain perilaku. Model Desain menampilkan karakteristik PL yang membatu praktisi untuk mengkonstruksinya secara efektif : arsitektur, antarmuka dan detail level komponen. Adapun, prinsip-prinsip yang digunakan pada pemodelan analisis antara lain mampu menampilkan domain informasi , menampilkan fungsi PL, menampikan perilaku PL, memiliki partisi dari tiga representasi ini, dan bergerak dari esensi menuju implementasi. Disamping itu, pada praktek pemodelan desain harus memenuhi prinsip: desain harus dapat dilacak dari model analisis, senantiasa memahami arsitektur, fokus pada desain data, antarmuka (pengguna maupun internal) harus didesain,

komponen harus menunjukkan independensi fungsional, komponen-komponen harus “loosely coupled”, representasi desain harus mudah dipahami, model desain harus dikembangkan secara iterative.

Praktek Konstruksi meliputi beberapa prinsip yaitu:

1. prinsip persiapan, yaitu mempersiapkan segala sesuatu sebelum menulis kode pemrograman.
2. prinsip coding, yaitu membuat kode program yang sesuai dengan kebutuhan pengguna program.
3. prinsip validasi, yaitu melakukan pengecekan kevalidan program.
4. prinsip pengujian, yaitu menguji program tersebut apakah telah sesuai dengan kebutuhan atau belum.

Praktek Deployment untuk RPL harus meliputi beberapa hal, antara lain adalah mengelola harapan pengguna pada setiap tahap, paket penyajian lengkap harus disusun terpadu dan teruji, harus menyediakan tim pendukung, materi pelatihan harus disediakan pada pengguna akhir, memperbaiki PL yang buggy, setelah itu baru disajikan.

2.3 Model COCOMO

COCOMO adalah sebuah model yang didesain oleh Barry Boehm untuk memperoleh perkiraan dari jumlah orang-bulan yang diperlukan untuk mengembangkan suatu produk perangkat lunak. Satu hasil observasi yang paling penting dalam model ini adalah bahwa motivasi dari tiap orang yang terlibat ditempatkan sebagai titik berat. Hal ini menunjukkan bahwa kepemimpinan dan kerja sama tim merupakan sesuatu yang penting, namun demikian poin pada bagian ini sering diabaikan.

Model COCOMO dapat diaplikasikan dalam tiga tingkatan kelas:

1. Proyek organik, adalah proyek dengan ukuran relatif kecil, dengan anggota tim yang sudah berpengalaman, dan mampu bekerja pada permintaan yang relatif fleksibel.
2. Proyek sedang (semi-terpisah), adalah proyek yang memiliki ukuran dan tingkat kerumitan yang sedang, dan tiap anggota tim memiliki tingkat keahlian yang berbeda.
3. Proyek terintegrasi, adalah proyek yang dibangun dengan spesifikasi dan operasi yang ketat.

Persamaan dasar model COCOMO adalah:

E = ab (KLOC)b

b

D = cb (E)d

b

P = E / D

Dimana **E** adalah usaha dalam orang-bulan, **D** adalah waktu pengerjaan dalam satuan bulan, **KLOC** adalah estimasi jumlah baris kode dalam ribuan, dan P adalah jumlah orang yang diperlukan. Koefisien ***ab***, ***bb***, ***cb,*** dan ***db*** diberikan pada tabel berikut:

Tabel 2.1 Koefisien model COCOMO dasar

Proyek Perangkat Lunak	a_b	b_b	c_b	d_b
Organik	2.4	1.05	2.5	0.38
Sedang	3.0	1.12	2.5	0.35
Terintegrasi	3.6	1.20	2.5	0.32

Pengembangan model COCOMO adalah dengan menambahkan atribut yang dapat menentukan jumlah biaya dan tenaga dalam pengembangan perangkat lunak, yang dijabarkan dalam kategori dan subkategori sebagai berikut:

1. Atribut produk
    a. Reliabilitas perangkat lunak yang diperlukan
    b. Ukuran basis data aplikasi
    c. Kompleksitas produk
2. Atribut perangkat keras
    a. Performa program ketika dijalankan
    b. Memori yang dipakai
    c. Stabilitas mesin virtual
    d. Waktu yang diperlukan untuk mengeksekusi perintah
3. Atribut Sumber Daya Manusia
    a. Kemampuan analisis
    b. Kemampuan ahli perangkat lunak
    c. Pengalaman membuat aplikasi
    d. Pengalaman menggunakan mesin virtual
    e. Pengalaman dalam menggunakan bahasa pemrograman
4. Atribut proyek
    a. Menggunakan perangkat lunak tambahan
    b. Metode rekayasa perangkat lunak
    c. Waktu yang diperlukan

Masing-masing sub-katergori diberi bobot antara 0 (sangat rendah) sampai 6 (sangat tinggi), dan kemudian dijumlahkan. Dari pengembangan ini diperoleh persamaan:

$$\mathbf{E}=a_i(\mathbf{KLOC})^{(b_i)}.\mathbf{EAF}$$

Dimana E adalah usaha dalam satuan orang-bulan, KLOC adalah estimasi jumlah baris kode dalam ribuan, dan **EAF** adalah faktor hasil penghitungan dari subkategori di atas. Koefisien **ai** dan eksponen **bi** diberikan pada tabel berikut:

Tabel 2.2 Koefisien model COCOMO lanjut

Proyek Perangkat Lunak	a_i	b_i
Organik	3.2	1.05
Sedang	3.0	1.12
Terintegrasi	2.8	1.20

BAB III

KESIMPULAN

1. Software engineering biasa disebut sebagai perangkat lunak, yaitu sebuah produk yang menyajikan potensi komputasi serta menghasilkan dan mengelola informasi
2. Aplikasi perangkat lunak dapat diklasifikasikan sebagai system software, application software, engineering software, embedded software, product line software, Web Applications, dan AI software.
3. Praktek-Praktek RPL digunakan untuk memahami bingkai kerja proses umum yang antara lain adalah komunikasi, perencanaan, pemodelan, konstruksi, dan deployment.
4. COCOMO adalah sebuah model yang didesain oleh Barry Boehm untuk memperoleh perkiraan dari jumlah orang-bulan yang diperlukan untuk mengembangkan suatu produk perangkat lunak.

DAFTAR PUSTAKA

B, Hetzel. 1993. *Making Software Measurement Work: Building and Effective Measurement Program. QED Technical Publishing Group.* Boston. Masssachusetts. ISBN.

C, Jones. 1986. *Programming Productivity.* New York. New York. ISBN.

Lowell Jay Arthur. 1985. *Measuring Program Productivity and Software Quality.* John Wiley and son Inc. New York. NY. ISBN.

Presmann. 2000. *R.S software Engineering A Practiotioner Approach. 5thedition.* New York. Mc Graw Hill. ISBN.

Wikipedia, http://www.wikipedia.org diakses tanggal 19 mei 2009.

ConvertAll, http://www.beltz.org/convertall diakses tanggal 19 mei 2009.