

MANAJEMEN PROYEK PERANGKAT LUNAK

Ada 3 fokus manajemen proyek perangkat lunak (PL) :

1. People (manusia)
 2. Problem (masalah)
 3. Process (proses)
- ❖ Manusia → mempertinggi kesiapan organisasi PL untuk mengerjakan aplikasi yang semakin kompleks
 - ❖ Masalah → Objektifitas dan ruang lingkupnya harus ditetapkan, pemecahan alternatifnya harus dipertimbangkan, teknik dan batasanpun harus didefinisikan
 - ❖ Proses → memberikan suatu kerangka kerja dimana rencana komprehensif bagi pengembangan perangkat lunak.

MANAJEMEN PROYEK PERANGKAT LUNAK

Adalah aktifitas dalam memanajemen rekayasa perangkat lunak, dimulai sebelum aktifitas teknis di inisialisasi dan berlanjut pada keseluruhan batasan, perkembangan dan pemeliharaan perangkat lunak komputer.

Masalah terbagi menjadi 2 bagian :

1. Ruang Lingkup :
 - Konteks
 - Tujuan / informasi
 - Fungsi / unjuk kerja
2. Dekomposisi Masalah :
 - Fungsionalitas
 - Proses

PARA PEMAIN

Manajer Senior → Menentukan isu-isu bisnis

Manajer (Teknik) Proyek → Memimpin pembuatan proyek PL

Pelaksana → Menyampaikan ketrampilan teknik yang diperlukan untuk mengembangkan Perangkat Lunak.

Pelanggan → Menentukan jenis kebutuhan bagi perangkat lunak.

Pemakai Akhir → Yang menggunakan / berinteraksi dengan PL.

Pemimpin harus memiliki kemampuan : *Motivasi, Organisasi, Inovatif*

Pilihan-pilihan yang dapat diambil untuk menerapkan sumber daya manusia kepada sebuah proyek, dimana n manusia.

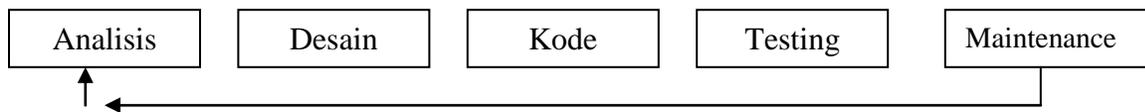
1. n manusia mengerjakan tugas-tugas fungsional yang berbeda sebanyak m dengan sedikit kombinasi kerja.
2. $m < n$, dimana m tugas dan n manusia sehingga tim informal terbentuk.
3. n orang diatur dalam t tim dimana setiap tim bertugas mengerjakan satu tugas fungsional atau lebih

Struktur tim “terbaik” tergantung pada gaya manajemen sebuah organisasi . Ada 3 organisasi tim yang umum :

1. Demokratis Desentralisasi
Pemimpin tidak permanen, tugasnya pendek, koordinator berganti, keputusan dibuat oleh konsensus kelompok, komunikasi bersifat horisontal.
2. Terkontrol Desentralisasi
Pemimpin primer mengkoordinasi tugas khusus dan pemimoin sekunder bertanggung jawab atas sub-sub masalah. Komunikasi horisontal tapi kontrol dilakukan vertikal.
3. Terkontrol Sentralisasi
Koordinasi pemecahan masalah dilakukan di pimpinan puncak dan dibentuk tim internal, komunikasi secara vertikal.

MODEL PROSES REKAYASA PL

1. Sekuensial Linier (waterfall)



Analisis Kebutuhan PL

Pengumpulan kebutuhan, domain informasi, unjuk kerja, antar muka diperlukan

Desain

Struktur data, arsitektur PL, representasi interface, algoritma detail.

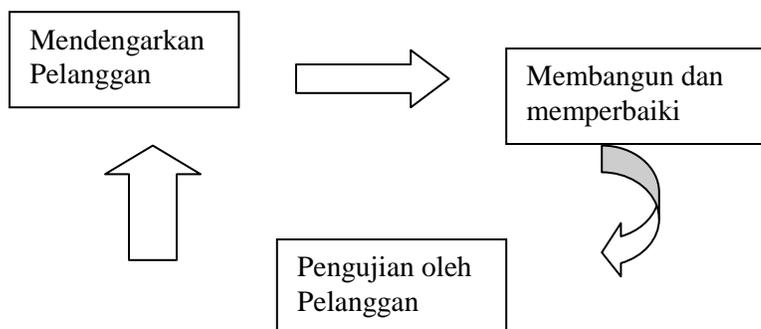
Kode / Coding

Pengubahan kedalam bentuk mesin yang dapat dibaca.

Pengujian

Logika internal PL, memastikan semua sesuai dengan keinginan

2. Prototipe

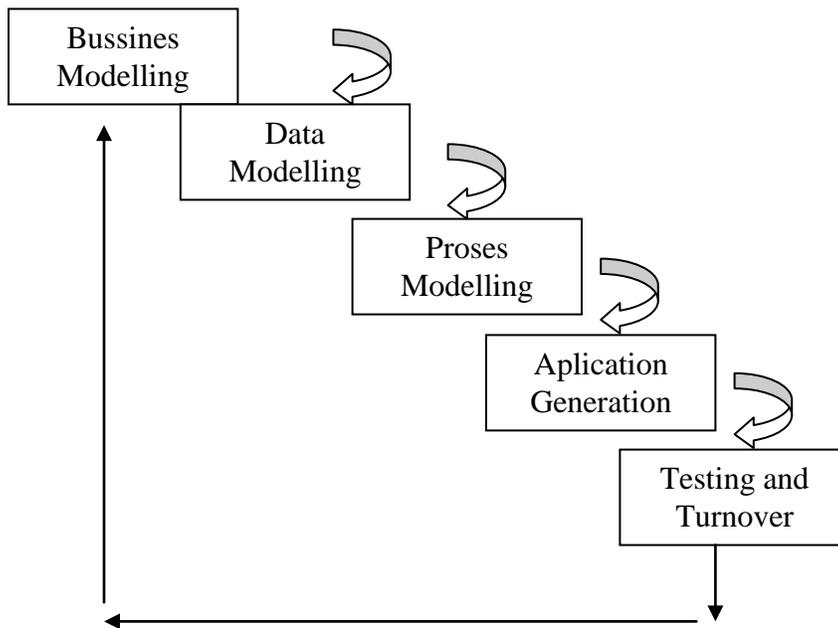


Secara ideal prototipe sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan PL. Apa kelebihan / kelemahan prototipe??

3. RAD (Rapid Application Development)

Pengembangan sekuensial linier yang dipercepat dengan menggunakan pendekatan konstruksi berbasis komponen.

Fase-fase RAD

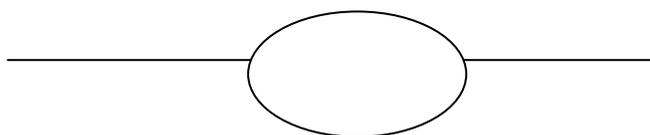


Bila sistem tidak dapat dimodulkan secara sistematis / teratur, komponen pengembangan RAD akan menjadi problematis. RAD menekankan perkembangan komponen program yang dapat *Reusable*.

4. SPIRAL

Model proses PL yang evalusioner, yang merangkai sifat interatif dari prototipe dengan cara kontrol dan aspek sistematis dari model sekuensial linear.

Komunikasi Dgn pelanggan Perencanaan analisis resiko



PENJELASAN TAHAPAN :

- Komunikasi Pelanggan
→ mencatat kebutuhan pelanggan
- Perencanaan
→ Mendefinisikan sumber daya, schedul waktu
- Analisis Resiko
→ Menaksir resiko baik teknis maupun manajemen
- Rekayasa
→ Membangun satu/lebih representasi aplikasi
- Konstruksi
→ Mengkontruksi, menguji, menginstal dan pelatihan
- Evaluasi Pelanggan
→ Umpan balik dari pelanggan selama implementasi

Pengaruh karakteristik proyek pada struktur tim

Tipe tim	DD	CD	CC
1. Tingkat kesulitan			
Tinggi	X		
Rendah		X	X
2. Ukuran			
Besar		X	X
Kecil	X		
3. Umur Tim			
Singkat		X	X
Panjang	X		
4. Modularitas			
Tinggi		X	X
Rendah	X		
5. Keandalan			
Tinggi	X	X	
Rendah			X
6. Tanggal Pengiriman			
Ketat/pasti			X
Longgar	X	X	
7. Sosiabilitas			

Tinggi
Rendah

X

X

X

DD = Demokratis Desentralisasi
CD = Terkontrol Desentralisasi
CC = Terkontrol Sentralisasi

METRIK PENGUKURAN PROYEK PL / SISTEM

Pengukuran merupakan sesuatu hal yang pokok bagi disiplin ilmu eksak. Pengukuran dapat digunakan pada proyek PL untuk membantu perhitungan, kualitas kontrol, perkiraan produktivitas dan kontrol proyek.

Pengukuran Output → Sebuah fungsi usaha dan waktu yang diaplikasikan
Pengukuran Kesesuaian Pemakaian → Produk kerja yang dihasilkan

Pengukuran, Metrik dan Indikator

Measure (mengukur) : mengidentifikasi kuantitatif dari jumlah, dimensi, kapasitas atau ukuran dari atribut sebuah produk.

Measurement (Pengukuran) : kegiatan menentukan sebuah ukuran.

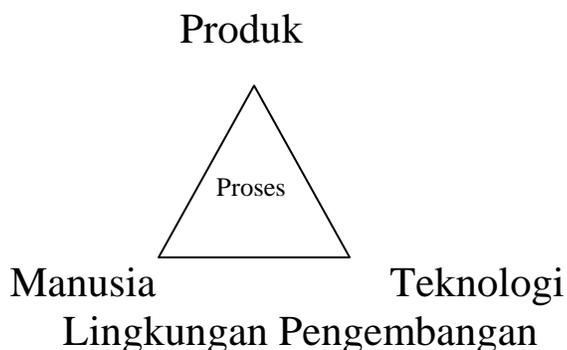
Metrik : ukuran kuantitatif dari tingkat dimana sebuah sistem, komponen atau proses memiliki atribut tertentu.

Rekayasa Perangkat Lunak mengumpulkan pengukuran dan mengembangkan metrik sehingga diperoleh suatu indikator yang dapat memungkinkan manajer proyek agar dapat menyesuaikan proses, proyek dan produk untuk menjadi lebih baik.

Indikator : sebuah metrik atau kombinasi dari metrik yang memberikan pengetahuan kedalam proses PL.

Indikator → Proses : memperoleh pengetahuan tentang reliabilitas
Sebuah proses yang sedang berlangsung

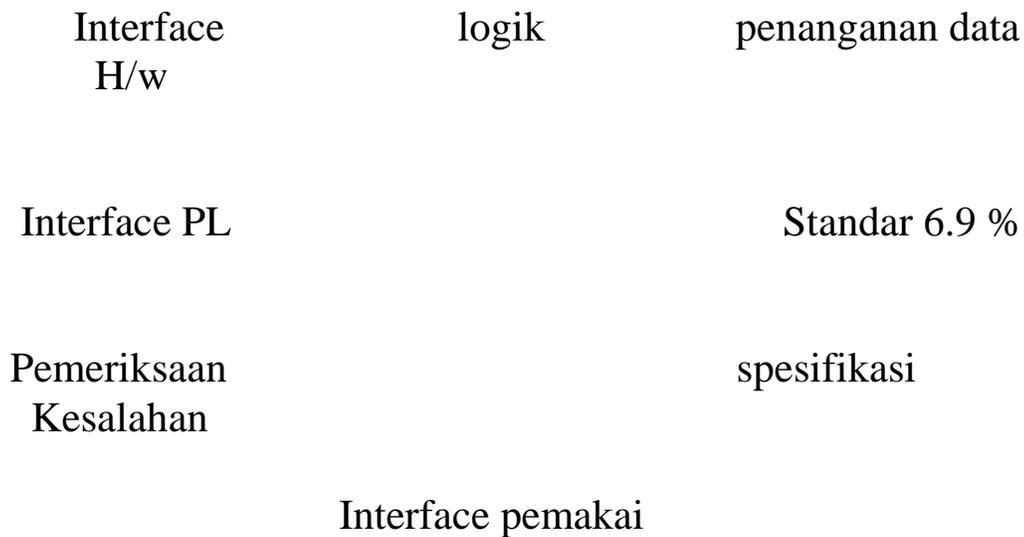
- Proyek : Memungkinkan manajer proyek PL untuk :
- memperkirakan status sebuah proyek yang sedang berlangsung
 - Menelusuri resiko-resiko potensial
 - Menemukan area masalah
 - Menyesuaikan aliran kerja atau tugas-tugas
 - Mengevaluasi kemampuan tim proyek untuk mengontrol kualitas hasil kerja RPL



Gambar : Determinan untuk kualitas dan efektifitas organisasional perangkat lunak

Statistical Software Proses Improvement (SSPI)

Analisis kegagalan PL untuk mengumpulkan informasi seputar semua kesalahan dan cacat yang terjadi pada sebuah aplikasi, sistem atau produk.



Gambar : Penyebab cacat dan asalnya pada proyek PL

Metrik proyek memiliki tujuan ganda :

1. Untuk meminimalkan jadwal pengembangan dengan melakukan penyesuaian yang diperlukan untuk menghindari penundaan serta mengurangi masalah resiko potensial.
 2. Untuk memperkirakan kualitas produk pada basis yang berlaku dan modifikasi untuk meningkatkan kualitas.
- Pada saat kualitas meningkat ; kesalahan menjadi minimal dan selagi kesalahan berkurang, jumlah kerja ulang yang dibutuhkan selama proyek berlangsung juga berkurang. Dengan demikian pembiayaan proyek secara keseluruhan dapat berkurang.

Bahwa setiap proyek seharusnya mengukur :

- ❖ Input → SDM, lingkungan yang dibutuhkan
- ❖ Output → produk kerja selama proses RPL
- ❖ Hasil → Efektifitas kemampuan penyampaian dari aktifitas satu ke aktifitas kerangka kerja selanjutnya.

Pengukuran Perangkat Lunak

1. Pengukuran langsung → biaya & usaha yang diaplikasikan

- ❖ Listing program
- ❖ Kecepatan eksekusi
- ❖ Ukuran memori
- ❖ Error pada sejumlah periode waktu.

2. Pengukuran tidak langsung → fungsionalitas dan kualitas, kompleksitas, efisiensi, maintenance (kemampuan pemeliharaan), reliabilitas (keandalan).

Metrik proyek memiliki tujuan ganda :

1. Untuk meminimalkan jadwal pengembangan dengan melakukan penyesuaian yang diperlukan, untuk menghindari penundaan serta mengurangi masalah dan resiko potensial.
2. Untuk memperkirakan kualitas produk pada basis yang berlaku dan modifikasi untuk meningkatkan kualitas.

❖ Pada saat kualitas meningkat, kesalahan menjadi minimal dan selagi kesalahan berkurang , jumlah kerja ulang yang dibutuhkan selama proyek berlangsung juga berkurang. Dengan demikian pembiayaan proyek secara keseluruhan dapat berkurang.

Bahwa setiap proyek seharusnya mengukur :

- a. Input → SDM, lingkungan yang dibutuhkan
- b. Output → produk kerja selama proses RPL
- c. Hasil → Efektifitas kemampuan dari aktifitas saru ke aktifitas kerangka kerja selanjutnya.

PENGUKURAN PERANGKAT LUNAK

Pengukuran langsung → biaya dan usaha yang diaplikasikan

- Listing program
- Kecepatan eksekusi
- Ukuran memori
- Error pada sejumlah periode waktu

Pengukuran tidak langsung → fungsionalitas dan kualitas, kompleksitas, efisiensi, maintenance (kemampuan pemeliharaan), Reliabilitas (keandalan).

Beberapa metode yang dapat digunakan untuk pengukuran :

1. Metrik size oriented

Pengukuran produktifitas dengan mempertimbangkan “ukuran” perangkat lunak yang dihasilkan, dapat dibuat tabel dibawah ini :

Proyek	LOC	Dollar \$	Halaman	Error	Cacat	SDM
Alpha	12.000	168	365	134	29	3
Beta	27.200	440	1231	321	86	5
Gamma	20.993	312	1223	256	64	7

Dari data tabel yang belum normalisasi dapat dikembangkan metrik size oriented yang sederhana untuk setiap proyek. Seperti :

- Kesalahan (error) per KLOC
- \$ per KLOC, cacat (defect) per KLOC
- Halaman dokumentasi per KLOC
- Kesalahan / person
- LOC per person-month
- \$ / halaman dokumentasi

2. Metrik function oriented

Pengukuran fungsionalitas yang disampaikan oleh aplikasi sebagai suatu nilai normalisasi. Karena fungsionalitas tidak dapat diukur langsung sehingga butuh suatu cara yang dapat dijadikan acuan yaitu dengan function – point.

Parameter Pengukuran	Jumlah	Faktor pembobotan			total
		Sederhana	Rata-rata	kompleks	
Jumlah input pemakai	6	3	4	6	24
Jumlah interface internal	7	6	7	10	
Jumlah file	5	4	5	7	

Jumlah output pemakai	9	7	10	15	
TOTAL					

Function point didapat dari sebuah hubungan empiris berdasarkan pengukuran langsung domain informasi perangkat lunak yang dapat dihitung.

Untuk menghitung titik-titik fungsi (fp) dipakai hubungan rumus sebagai berikut :

$$Fp = \text{jumlah total} \times [0.65 + 0.01 \times f_i]$$

Dimana f_i adalah harga penyesuaian kompleksitas berdasarkan respon pada pertanyaan

Rata – rata setiap faktor pada skala 0 – 5

0	1	2	3	4	5
Tidak pengaruh	insident al	moderat	Rata- rata	signifik an	essensi al

Daftar pertanyaan :

1. apakah sistem membutuhkan back-up dan recovery yang reliable ?
2. apakah komunikasi data dibutuhkan ?
3. apakah fungsi pemrosesan didistribusikan ?
4. apakah kinerja penting ?
5. apakah sistem membutuhkan entri data secara on line?
6. Apakah kode di desain untuk dapat dipakai kembali ?

Bila sudah dapat fp maka dapat dijadikan acuan untuk produktivitas, kualitas serta atribut-atribut yang lain seperti :

- Kesalahan per fp
- Cacat per fp
- \$ per fp

Ukuran LOC (list of code) dan fp sering digunakan untuk mendapatkan metrik produktifitas walaupun tidak mutlak sebagai ukuran produktivitas (masih dalam perdebatan)

Basili dan Zelkowitz menetapkan 4 faktor yang mempengaruhi produktifitas PL, yaitu :

- a. Faktor manusia → ukuran dan keahlian organisasi pengembangan
- b. Faktor masalah → kompleksitas masalah yang dipecahkan dan jumlah perubahan/ batasan
- c. Faktor proses → teknik analisis dan desain yang digunakan bahasa pemrograman yg dipakai
- d. Faktor sumber daya → pemakaian h/w dan s/w yang lain

Jika faktor-faktor tersebut diatas rata-rata (sangat baik) maka produktifitas akan secara signifikan menjadi baik.

MENGUKUR KUALITAS PL

Gilb T. mendefinisikan ukuran kualitas PL dilihat dari beberapa aspek, yaitu :

- a. Catat per KLOC; dimana cacat diartikan kurangnya kesesuaian dengan persyaratan
- b. Maintainabilitas; kemudahan dimana program dapat dikoreksi jika ditemukan kesalahan, dapat beradaptasi jika lingkungan berubah. Mudah dikembangkan bila pelanggan menginginkan perubahan.
- c. Integritas; kemampuan sistem untuk menahan serangan terhadap sekuritasnya. Serangan dapat berupa virus maupun hacker yang mengganggu program data maupun dokumen.

$$\text{Integritas} = \Sigma [1 - \text{ancaman} \times (1 - \text{sekuritas})]$$

Ancaman : kemungkinan serangan yang terjadi dalam periode waktu yang ditentukan.

- d. Usabilitas; usaha untuk mengukur user friendly dengan 4 karakter, yakni :
 - ketrampilan fisik atau intelektual untuk mempelajari sistem
 - waktu yang diperlukan untuk menggunakan sistem
 - penilaian subjektifitas terhadap sistem (menggunakan kuisioner)
 - peningkatan produktifitas ketika sistem digunakan

EFISIENSI PENGHAPUSAN CACAT (DEFECT)

Defect Removal Efficiency (DRE) dapat dihitung dengan rumus :

$$DRE = E / (E + D)$$

Dimana E : jumlah kesalahan yang ditemukan sebelum PL dikirim ke user

D : jumlah cacat yang ditemukan setelah pengiriman

Nilai ideal, DRE adalah 1 (karena tidak ada cacat), jika E bertambah maka kemungkinan besar nilai akhir D akan berkurang (kesalahan disaring sebelum menjadi cacat).

DRE dapat digunakan untuk menilai kemampuan tim menemukan kesalahan sebelum ke proses pengembangan PL berikutnya. Rumus yang digunakan :

$$DRE_i = E_i / (E_i + E_{i+1})$$

PERENCANAAN PROYEK PERANGKAT LUNAK

Aktifitas pertama dari perencanaan proyek PL adalah estimasi, meskipun estimasi merupakan sebuah seni tetapi ada beberapa teknik dalam membuat estimasi baik waktu maupun biaya. Ada beberapa komponen dalam membuat estimasi :

1. Project Complexity

Akan berpengaruh terhadap ketidakpastian yang inheren dalam perencanaan, walaupun kompleksitas merupakan pengukuran relatif bagi setiap pengembang, sebagai contoh pengembang yang terbiasa membuat aplikasi real time merasa kesulitan jika mendapatkan proyek aplikasi batch.

2. Project Size

Bila ukuran bertambah maka ketergantungan diantara elemen PL akan meningkat dengan cepat. Sehingga ukuran proyek PL akan mempengaruhi akurasi estimasi. Dekomposisi masalah merupakan salah satu teknik mengatasi proyek PL yang besar.

3. Structural Uncertainty

Akan berpengaruh pada resiko estimasi. Bila proyek masa lalu dapat dikerjakan secara komprehensif dan terdokumentasi dengan rapi maka estimasi dapat dilakukan dengan kepastian yang lebih tinggi. Jadwal dapat dibuat untuk menghindari kesulitan-kesulitan yang terjadi dimasa lalu dan resiko keseluruhan dapat dikurangi.