

BAB 4

PROSES PERANGKAT LUNAK & METRIK PROYEK

Lord Kelvin berkata :

Bila Anda dapat mengukur apa yg sedang Anda bicarakan dan mengekspresikannya dalam angka, berarti Anda memahaminya.

Tujuan pengukuran perangkat lunak adalah :

- Untuk menyatakan kualitas produk
- Untuk menilai kualitas manusia yg terlibat dalam pembuatan produk.
- Untuk menilai keuntungan pemakaian metode & alat bantu yg baru.
- Sebagai dasar untuk melakukan perkiraan.
- Untuk membantu penyesuaian pemakaian alat bantu yg baru atau pelatihan tambahan.

Metrik perangkat lunak mengacu pada pengukuran perangkat lunak komputer. Pengukuran digunakan untuk membantu perhitungan, kontrol kualitas, perkiraan produktivitas, & kontrol proyek, serta untuk membantu mengambil keputusan taktis pada saat proyek sudah berjalan.

PENGUKURAN, METRIK, DAN INDIKATOR

Measure (mengukur) :

Mengindikasikan kuantitatif dari luasan, jumlah, dimensi, kapasitas, atau ukuran dari atribut sebuah proses atau produk.

Measurement (pengukuran) :

Kegiatan menentukan sebuah measure (pengukuran)

Metrics (metrik) :

Ukuran kuantitatif dari tingkat dimana sebuah sistem, komponen, atau proses memiliki atribut tertentu.

RPL mengumpulkan pengukuran & mengembangkan metrik sehingga diperoleh suatu indicator.

Indicator (indicator) :

Sebuah metrik atau kombinasi dari metrik yg memberikan pengetahuan kedalam proses PL, sebuah proyek PL, atau produk itu sendiri.

Indikator memberikan pengetahuan yang memungkinkan manajer proyek atau perekayasa PL menyesuaikan proses, proyek, dan produk, untuk membuat semuanya menjadi lebih baik.

METRIK DALAM PROSES DAN DOMAIN PROYEK

- METRIK PROSES
- METRIK PROYEK

Metrik harus dikumpulkan sehingga indikator proses dan indikator produk (proyek) dapat dipastikan.

Indikator proses memungkinkan :

1. sebuah organisasi rekayasa PL memperoleh pengetahuan tentang reliabilitas sebuah proses yg sedang berlangsung
2. manajer & pelaksana memperkirakan apa yg harus dikerjakan dan yang tidak.

Indikator proyek memungkinkan manajer proyek PL :

1. memperkirakan status sebuah proyek yg sedang berlangsung
2. menelusuri resiko-resiko potensial
3. menemukan area masalah sebelum masalah 'menjadi semakin kritis'.
4. menyesuaikan aliran kerja atau tugas-tugas.
5. mengevaluasi kemampuan tim proyek untuk mengontrol kualitas hasil kerja RPL.

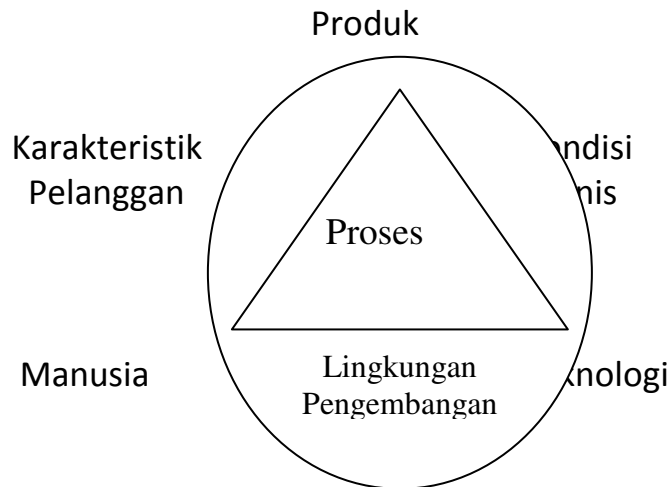
METRIK PROSES

Metrik proses digunakan untuk tujuan strategis.

Cara untuk meningkatkan proses perangkat lunak :

- mengukur atribut tertentu dari proses
- mengembangkan serangkaian metrik yg berarti

- menggunakan metrik itu untuk memberikan indikator yg akan membawa kepada sebuah strategi pengembangan.



Gbr. Determinan untuk kualitas dan efektivitas organisasional PL.

Mengukur reliabilitas proses PL secara tidak langsung yaitu dengan mengambil serangkaian metrik berdasarkan **keluaran** yg dapat diambil oleh proses.

Keluaran menyangkut :

- pengukuran kesalahan yg ditemukan sebelum pelepasan PL.
- cacat yg disampaikan & dilaporkan oleh pemakai akhir.
- produk kerja yg dikirim.
- usaha manusia yg dilakukan
- waktu kalender yg digunakan
- konfirmasi jadwal
- dll

Pada saat organisasi menjadi lebih nyaman dengan kumpulan & manfaat metrik proses, derivasi dari indikator sederhana memberikan suatu cara kepada suatu pendekatan yg lebih teliti yg disebut **SSPI (Statistical Software Process Improvement)**.

SSPI menggunakan **analisis kegagalan PL** untuk mengumpulkan informasi seputar semua kesalahan & cacat yg terjadi pada saat sebuah aplikasi, sistem, atau produk dikembangkan dan dipakai.

Kesalahan :

Ketidaksempurnaan pd sebuah produk kerja yg ditemukan oleh perekayasa PL **sebelum** PL itu disampaikan kepada pemakai akhir.

Cacat :

Ketidaksempurnaan pd sebuah produk kerja yg ditemukan oleh perekayasa PL **setelah** PL itu disampaikan kepada pemakai akhir.

Analisis kegagalan bekerja dengan cara sbb. :

1. Semua kesalahan & cacat dikategorikan dari awal
2. Biaya untuk mengkoreksi setiap kesalahan & cacat dicatat.
3. Jumlah kesalahan & cacat dari setiap kategori dihitung dan ditata dalam urutan naik.
4. Biaya keseluruhan dari kesalahan & cacat dalam setiap kategori dihitung.
5. Data resultan dianalisis untuk menemukan kategori yg menelan biaya besar.
6. Rencana dikembangkan untuk memodifikasi proses guna mengeliminasi kelas kesalahan & cacat yg paling membutuhkan banyak biaya.

Berdasarkan langkah 1&2 diatas, ditemukan ada 8 penyebab kerusakan dan sumbernya :

- Sumber spesifikasi/persyaratan :
 - a. Logic 20%
 - b. Penanganan data 10,5%
 - c. Standar 6,9%

Sumber desain :

- a. Spesifikasi 25,5%
- Sumber kode :
 - a. Interface perangkat lunak 6,0%
 - b. Interface perangkat keras 7,7%

- c. Pemeriksaan kesalahan 10,9%
- d. Interface pemakai 11,7%

METRIK PROYEK

Tujuan metrik proyek :

1. untuk meminimalkan jadwal pengembangan dengan melakukan penyesuaian yg diperlukan untuk menghindari penundaan serta mengurangi masalah & resiko potensial.
2. untuk memperkirakan kualitas produk pada basis yg berlaku, dan bila dibutuhkan, memodifikasi pendekatan teknis untuk meningkatkan kualitas.

Pengukuran proyek PL bersifat taktis, yaitu bahwa metrik proyek & indikator yg berasal dari pengukuran digunakan oleh manajer proyek dan tim PL untuk mengadaptasi aliran kerja proyek & aktifitas teknis.

Selagi sebuah proyek berjalan, pengukuran usaha dan waktu kalender yg digunakan dibandingkan dengan perkiraan awal (dan jadwal proyek). Manajer proyek menggunakan data tersebut untuk memonitor & mengontrol kemajuan.

Selagi PL berjalan dari spesifikasi ke perancangan, metrik teknik dikumpulkan untuk memperkirakan kualitas desain serta memberikan indikator yg akan mempengaruhi pendekatan yg akan diambil untuk memunculkan kode & modul serta pengujian integrasi (integrated test).

Kualitas meningkat ----→ kesalahan menjadi minimal
Biaya-
berkurang
Kesalahan berkurang --→ jumlah kerja ulang berkurang

Model lain dari metrik proyek mengusulkan bahwa setiap proyek seharusnya mengukur :

- input (pengukuran sumber daya)

- output (pengukuran kemampuan penyampaian atau produk kerja yg diciptakan selama proses RPL)
- hasil (pengukuran yg menunjukkan kemampuan penyampaian)

PENGUKURAN PERANGKAT LUNAK

Pengukuran perangkat lunak dibedakan menjadi dua yaitu :

- Pengukuran langsung (direct)
 - Metrik Size-Oriented
- Pengukuran tidak langsung (indirect)
 - Metrik Function-Oriented
 - Metrik Function Point

Yang diukur pada pengukuran langsung adalah :

- Biaya
- Pengaruh
- Jumlah baris perintah (LOC) yg diproduksi
- Kecepatan eksekusi
- Ukuran memori
- Kesalahan

Yang diukur pada pengukuran tidak langsung adalah :

- fungsi
- kualitas
- kompleksitas
- efisiensi
- keandalan
- kemampuan pemeliharaan

Metrik Size-Oriented

Proyek	LOC	Usaha	Dolar	halaman	Kesalahan	cacat	Manusia
alpha	12,100	24	168	365	134	29	3
betha	27,200	62	440	1224	321	86	5
gamma	20,200	43	314	1050	256	64	6

--	--	--	--	--	--	--	--

Produktivitas = KLOC / usaha

Kualitas = kesalahan / KLOC

Biaya = biaya / KLOC

Dokumentasi = halaman / KLOC

Metrik size-oriented tidak diterima sebagai cara terbaik untuk mengukur proses pengembangan perangkat lunak. Sebagian besar berkisar di seputar pemakaian LOC.

Metrik Function-Oriented

Metode pendekatan yg digunakan dapat disebut : **Function Point (FP)**.

FP dihitung dgn melengkapi tabel dibawah ini :

Parameter pengukuran	jumlah	X	Faktor pembobotan			=	[]
			sederhana	rata-rata	kompleks		
Jumlah input pemakai	[]	X	3	4	6	=	[]
Jumlah output pemakai	[]	X	4	5	7	=	[]
Jml penyelidikan pemki	[]	X	3	4	6	=	[]
Jumlah file	[]	X	7	10	15	=	[]
Jml interface internal	[]	X	6	7	10	=	[]
Total ----->							[]

FP = jumlah total x [0,65 + 0,01 x jumlah(fi)]

Jumlah(fi) didapat dari jumlah range jawaban dari 14 pertanyaan berikut :

1. apakah sistem membutuhkan backup & recovery yg reliable ?
2. apakah komunikasi data dibutuhkan ?
3. apakah fungsi pemrosesan didistribusikan ?
4. apakah kinerja penting
5. apakah sistem akan berjalan pd lingkungan operasional yg sudah ada yg paling banyak digunakan ?
6. apakah sistem membutuhkan entry data online ?
7. apakah entry data online membutuhkan ada transaksi input terhadap layar atau operasi ganda
8. apakah file master diperbarui secara online ?
9. apakah input, output, file, atau inquiry kompleks ?
10. apakah pemrosesan internal kompleks ?
11. apakah kode didesain untuk dapat dipakai kembali ?
12. apakah desain melibatkan konversi dan instalasi
13. apakah sistem didesain untuk instalasi ganda dalam organisasi berbeda ?
14. apakah aplikasi didesain untuk memfasilitasi perubahan & mempermudah pemakai untuk menggunakannya ?

range jawaban (skala) untuk pertanyaan diatas antara 0 s/d 5 :

0 : tidak berpengaruh

1 : kurang penting

2 : cukup penting

3 : rata-rata

4 : penting

5 : sangat penting

Lima faktor penting yg mempengaruhi produktivitas perangkat lunak menurut Basili dan Zelkowitz :

1. faktor manusia
2. faktor masalah
3. faktor proses
4. faktor produk
5. faktor sumber daya

Faktor – faktor untuk mengukur kualitas perangkat lunak (4 metrik kualitas):

1. Cara yang benar

2. Maintainabilitas
3. Integritas
4. Usebilitas

Faktor – faktor yang mempengaruhi biaya pengembangan PL :

1. kemampuan programmer dan tenaga kerja
2. kekompleksan produk
3. ukuran produk
4. waktu yang tersedia
5. keandalan yang diperlukan
6. teknologi yang dipergunakan