



# Metrik Proses dan Proyek Perangkat Lunak

KARMILASARI

# Outline

- Pendahuluan
- Metrik dalam domain PROSES
- Metrik dalam domain PROYEK
- Pengukuran Perangkat Lunak
- Menintegrasikan Metrik dalam Proses Perangkat Lunak



# Pendahuluan



# Apa itu Metrik ?

- ▶ Proses perangkat lunak dan proyek metrik merupakan :
  - ▶ ukuran kuantitatif
  - ▶ alat manajemen
  - ▶ Mengukur efektivitas proses perangkat lunak dan proyek-proyek yang dilakukan dengan menggunakan proses sebagai kerangka
- ▶ Dasar kualitas dan produktivitas data dikumpulkan
- ▶ Data dianalisis, dibandingkan dengan rata-rata masa lampau, dan dinilai
- ▶ Tujuannya adalah untuk menentukan apakah kualitas dan produktivitas perbaikan telah terjadi
- ▶ Data tersebut juga dapat digunakan untuk menentukan area masalah
- ▶ Solusi dapat dikembangkan dan proses perangkat lunak dapat ditingkatkan

# Penggunaan Pengukuran

5

- ▶ Dapat diterapkan pada proses perangkat lunak dengan maksud memperbaikinya secara terus menerus
- ▶ Dapat digunakan di seluruh proyek perangkat lunak untuk membantu dalam estimasi, kontrol kualitas, penilaian produktivitas, dan pengendalian proyek
- ▶ Dapat digunakan untuk membantu menilai kualitas produk kerja perangkat lunak dan membantu pengambilan keputusan taktis sebagai hasil proyek



# Alasan dilakukan Pengukuran

6

## ▶ Karakterisasi

- ▶ Diperoleh pemahaman tentang proses, produk, sumber daya, dan lingkungan
- ▶ Ditetapkan baseline sebagai perbandingan penilaian masa depan

## ▶ Evaluasi

- ▶ Penentuan status terkait dengan rencana

## ▶ Prediksi

- ▶ Diperoleh pemahaman tentang hubungan antara proses dan produk
- ▶ Dibangun model hubungan diantara proses dan produk

## ▶ Peningkatan

- ▶ Identifikasi hambatan, akar masalah, inefisiensi, dan peluang lainnya untuk meningkatkan kualitas produk dan kinerja proses



# Metrik dalam Domain PROSES




# Metrik dalam Domain PROSES

- ▶ Metrik Proses merupakan kumpulan seluruh proyek pada jangka waktu tertentu
- ▶ Metrik Proses digunakan untuk membuat keputusan stragegis
- ▶ Tujuannya, memberikan seperangkat indikator proses yang mengarah pada perbaikan proses perangkat lunak dalam jangka panjang
- ▶ Cara untuk mengetahui bagaimana tuk meningkatkan proses :
  - ▶ Mengukur atribut tertentu dari proses
  - ▶ Mengembangkan satu set metrik bermakna berdasarkan atribut tersebut
  - ▶ Menggunakan metrik untuk memberikan indikator yang mengarah pada strategi untuk perbaikan



# Metrik dalam Domain PROSES *(lanjutan)*

- ▶ Mengukur efektivitas proses dengan menggunakan satu set metrik berdasarkan pada hasil/luaran dari proses seperti :
    - ▶ Kesalahan ditemukan sebelum rilis dari perangkat lunak
    - ▶ Cacat pengiriman ke dan dilaporkan oleh pengguna akhir
    - ▶ Penyebaran produk kerja
    - ▶ Usaha manusia yang dikeluarkan
    - ▶ Waktu kalender yang dikeluarkan
    - ▶ Kesesuaian dengan jadwal
    - ▶ Waktu dan usaha untuk menyelesaikan setiap kegiatan generik
- 

# Etika Metrik Proses

10

- ▶ Gunakan akal sehat dan sensitivitas organisasi ketika menginterpretasi data metrik
- ▶ Memberikan umpan balik reguler kepada individu dan tim yang mengumpulkan pengukuran dan metrik
- ▶ Jangan menggunakan metrik untuk mengevaluasi individu
- ▶ Bekerja dengan praktisi dan tim untuk menetapkan tujuan yang jelas dan metrik yang akan digunakan untuk mencapainya
- ▶ Jangan gunakan metrik untuk mengancam individu atau tim
- ▶ Metrik data yang menunjukkan adanya masalah tidak harus dianggap "negatif"
  - ▶ Data tersebut hanyalah indikator untuk perbaikan proses
- ▶ Jangan terobsesi pada satu metrik dengan mengesampingkan metrik penting lainnya

# Metrik dalam Domain PROYEK



# Metrik dalam Domain PROYEK

12

- ▶ Metrik proyek memungkinkan manajer proyek perangkat lunak untuk :
  - ▶ Menilai status proyek yang sedang berlangsung
  - ▶ Melacak potensi risiko
  - ▶ Mengungkap masalah dalam suatu daerah sebelum statusnya menjadi kritis
  - ▶ Sesuaikan alur kerja atau tugas
- ▶ Mengevaluasi kemampuan tim proyek untuk mengontrol kualitas produk kerja perangkat lunak
- ▶ Banyak metrik yang sama yang digunakan pada kedua proses dan domain proyek
- ▶ Metrik proyek yang digunakan untuk membuat keputusan taktis
- ▶ Metrik proyek digunakan untuk menyesuaikan alur kerja proyek dan kegiatan teknis



# Penggunaan Metrik Proyek

13

- ▶ Aplikasi pertama dari metrik proyek terjadi selama estimasi
  - ▶ Metrik dari proyek-proyek masa lalu digunakan sebagai dasar untuk memperkirakan waktu dan usaha
- ▶ Sebagai hasil proyek, jumlah waktu dan usaha yang dikeluarkan dibandingkan dengan estimasi awal
- ▶ Saat pekerjaan teknis dimulai, metrik proyek lainnya menjadi penting :
  - ▶ **Tingkat produksi** diukur (diwakili dalam hal model dibuat, ulasan jam, fungsi poin, dan disampaikan baris kode sumber)
  - ▶ **Kesalahan** ditemukan selama kegiatan kerangka kerja umum (yaitu, komunikasi, perencanaan, pemodelan, konstruksi, penyebaran) diukur



# Penggunaan Metrik Proyek *(lanjutan)*

14

- ▶ Metrik proyek digunakan untuk
  - ▶ Meminimalkan jadwal pengembangan dengan membuat penyesuaian yang diperlukan untuk menghindari keterlambatan dan mengurangi potensi masalah dan risiko
  - ▶ Menilai kualitas produk secara berkelanjutan dan, bila perlu, memodifikasi pendekatan teknis untuk meningkatkan kualitas
- ▶ Kesimpulan
  - ▶ Kualitas membaik → produk cacat diminimalkan
  - ▶ Produk cacat turun → jumlah pekerjaan ulang yang diperlukan selama proyek berkurang
  - ▶ Pengerjaan ulang turun → biaya proyek secara keseluruhan berkurang

# Perangkat Lunak Pengukuran



# Kategori Perangkat Lunak Pengukuran

16

- ▶ Dua kategori pengukuran perangkat lunak
  - ▶ Pengukuran Langsung dari
    - ▶ proses perangkat lunak (biaya, usaha, dll)
    - ▶ produk perangkat lunak (baris kode yang dihasilkan, kecepatan eksekusi, cacat dilaporkan dari waktu ke waktu, dll)
  - ▶ Pengukuran Tidak Langsung dari
    - ▶ produk perangkat lunak (fungsi, kualitas, kompleksitas, efisiensi, kehandalan, pemeliharaan, dll)
- ▶ Metrik proyek dapat dikonsolidasikan untuk membuat metrik proses sebuah organisasi

# Metrik berorientasi SIZE (ukuran)

17

- ▶ Diperoleh melalui normalisasi kualitas dan / atau produktivitas tindakan dengan mempertimbangkan ukuran perangkat lunak yang dihasilkan
- ▶ Seribu baris kode (Kilo Line Of Code/KLOC) dipilih sebagai nilai normalisasi
- ▶ Metrik mencakup
  - ▶ Kesalahan per KLOC - Kesalahan per orang-bulan
  - ▶ Cacat per KLOC - KLOC per orang-bulan
  - ▶ Dolar per KLOC - dolar per halaman dokumentasi
  - ▶ Halaman dokumentasi per KLOC

# Metrik berorientasi SIZE (ukuran)

18

- ▶ Metrik berorientasi size, tidak diterima secara universal sebagai cara terbaik untuk mengukur proses perangkat lunak
- ▶ Pendapat yang berlawanan dengan pengukuran KLOC, mengatakan :
  - ▶ Tergantung pada bahasa pemrograman
  - ▶ Menghukum program yang dirancang dengan baik tapi pendek
  - ▶ Tidak dapat dengan mudah mengakomodasi bahasa nonprocedural
  - ▶ Membutuhkan tingkat detail yang mungkin sulit dicapai



# Metrik berorientasi FUNGSI

19

- ▶ Metrik berorientasi fungsi menggunakan ukuran fungsi yang ditunjukkan oleh aplikasi sebagai nilai normalisasi
- ▶ Umumnya metrik ini menggunakan Titik Fungsi (Function Point)
  - ▶  $FP = Total * [0.65 + 0.01 * \text{jumlah (nilai Faktor)}]$
- ▶ Nilai-nilai Titik Fungsi pada proyek-proyek masa lalu dapat digunakan untuk perhitungan, misalnya, rata-rata jumlah baris kode per titik fungsi (misalnya, 60)

# Kontroversi Titik Point (Function Point/FP)

20

- ▶ Seperti ukuran KLOC, penggunaan titik fungsi juga memiliki pendukung dan penentang
- ▶ Para pendukung mengklaim bahwa
  - ▶ FP adalah bahasa pemrograman independen
  - ▶ FP didasarkan pada data yang lebih mungkin diketahui pada tahap awal proyek, sehingga lebih menarik sebagai pendekatan estimasi
- ▶ Para penentang mengklaim bahwa
  - ▶ FP memerlukan beberapa "sulap" karena perhitungan didasarkan pada data subjektif
  - ▶ Hitungan dari domain informasi bisa sulit dikumpulkan setelah fakta
  - ▶ FP tidak memiliki arti fisik langsung ... itu hanya nomor

# Rekonsiliasi Metrik LOC dan FP

21

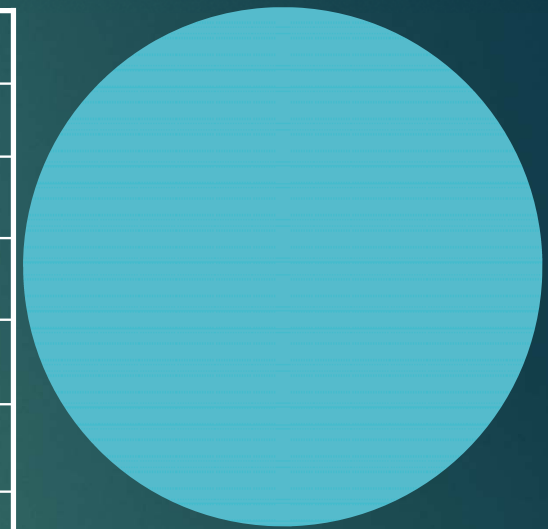
- ▶ Hubungan antara LOC dan FP tergantung
  - ▶ bahasa pemrograman yang digunakan untuk mengimplementasikan perangkat lunak
  - ▶ Kualitas desain
- ▶ FP dan LOC yang telah ditemukan untuk menjadi prediktor yang relatif akurat dari upaya pengembangan software dan biaya
  - ▶ Namun, dasar histori informasi pertama harus dibentuk
- ▶ LOC dan FP dapat digunakan untuk memperkirakan proyek perangkat lunak berorientasi objek
  - ▶ Namun, tidak memberikan cukup rincian untuk jadwal dan usaha penyesuaian yang diperlukan dalam iterasi dari proses evolusi atau incremental
- ▶ Tabel di slide berikutnya memberikan perkiraan kasar dari LOC rata-rata untuk satu FP dalam berbagai bahasa pemrograman

# LOC Per Function Point

22

Language	Average	Median	Low	High
Ada	154	--	104	205
Assembler	337	315	91	694
C	162	109	33	704
C++	66	53	29	178
COBOL	77	77	14	400
Java	55	53	9	214
PL/1	78	67	22	263
Visual Basic	47	42	16	158

[www.qsm.com/?q=resources/function-point-languages-table/index.html](http://www.qsm.com/?q=resources/function-point-languages-table/index.html)





# Metrik berorientasi OBJEK

- ▶ Jumlah skrip skenario (contoh, kasus penggunaan)
  - ▶ Jumlah ini secara langsung berkaitan dengan ukuran aplikasi dan
- ▶ Jumlah kasus uji yang diperlukan untuk menguji sistem
- ▶ Jumlah kelas kunci (komponen yang sangat independen)
  - ▶ Kelas utama didefinisikan di awal analisis berorientasi objek dan merupakan inti dari domain masalah
  - ▶ Angka ini menunjukkan jumlah usaha yang dibutuhkan untuk mengembangkan perangkat lunak
  - ▶ Hal ini menunjukkan jumlah potensi penggunaan kembali untuk diterapkan selama pengembangan
- ▶ Jumlah kelas dukungan
  - ▶ Kelas dukungan diperlukan untuk menerapkan sistem tetapi tidak serta merta terkait dengan masalah domain (misalnya, user interface, database, perhitungan)
  - ▶ Angka ini menunjukkan jumlah usaha dan potensi penggunaan kembali



# Metrik berorientasi OBJEK *(lanjutan)*

24

- ▶ Jumlah rata-rata kelas dukungan per kelas utama
  - ▶ Kelas utama diidentifikasi pada awal proyek (misalnya, pada analisis kebutuhan)
  - ▶ Estimasinya sejumlah kelas dukungan dapat dibuat dari sejumlah kelas utama
  - ▶ Aplikasi GUI memiliki kelas dukungan antara dua dan tiga kali lebih banyak sebagai kelas kunci
  - ▶ Aplikasi non-GUI memiliki dukungan kelas antara satu dan dua kali lebih sebagai kelas utama
- ▶ Jumlah subsistem
  - ▶ Sebuah subsistem merupakan agregasi dari kelas yang mendukung fungsi yang dapat dilihat oleh pengguna akhir dari suatu sistem

# Metrik untuk Kualitas Perangkat Lunak

25

- ▶ Kebenaran (*Correctness*)
  - ▶ jumlah cacat per KLOC, di mana cacat adalah adanya perbedaan verifikasi dengan kesesuaian persyaratan yang ditentukan
  - ▶ cacat adalah masalah-masalah yang dilaporkan oleh pengguna program setelah program ini dirilis untuk penggunaan umum
- ▶ Pemeliharaan
  - ▶ menggambarkan kemudahan program untuk dikoreksi jika kesalahan ditemukan, dapat beradaptasi dengan perubahan lingkungan, atau dapat ditingkatkan jika pelanggan memerlukan perubahan persyaratan
  - ▶ Waktu yang Berarti untuk Perubahan (Mean Time To Change/MTTC): waktu untuk menganalisis, merancang, melaksanakan, tes, dan mendistribusikan perubahan ke semua pengguna
    - ▶ program dipertahankan rata-rata memiliki MTTC rendah

# Efisiensi Penghapusan Kecacatan

26

- ▶ Efisiensi penghapusan kecacatan memberikan manfaat baik pada tingkatan proyek maupun proses
- ▶ Merupakan ukuran dari kemampuan penyaringan kegiatan Tanya Jawab (QA) seperti yang diterapkan di seluruh aktivitas kerangka proses
  - ▶ Menunjukkan persentase kesalahan perangkat lunak yang ditemukan sebelum perangkat lunak di-release
- ▶ Didefinisikan :  $DRE = E / (E + D)$ 
  - ▶ E adalah jumlah kesalahan yang ditemukan sebelum pengiriman perangkat lunak untuk pengguna akhir
  - ▶ D adalah jumlah cacat yang ditemukan setelah pengiriman
- ▶ Jika D meningkat → DRE menurun
- ▶ Nilai ideal DRE adalah 1, yang berarti tidak ada cacat yang ditemukan setelah didistribusikan
- ▶ DRE mendorong tim perangkat lunak untuk lembaga teknik untuk menemukan sebanyak kesalahan mungkin sebelum pengiriman

# Integrasi Metrik dengan Proses Perangkat Lunak





# Argumen Metrik Perangkat Lunak

28

- ▶ Kecenderungan pengembang perangkat lunak tidak melakukan pengukuran, dan sebagian besar memiliki sedikit keinginan untuk memulai pengukuran
- ▶ Untuk mendirikan perusahaan perusahaan lunak yang sukses menerapkan metrik perangkat lunak memerlukan upaya bertahun-tahun.
- ▶ Tetapi jika kita tidak melakukan pengukuran, tidak ada cara nyata untuk menentukan apakah terjadi peningkatan dari apa yang kita lakukan atau produksi
- ▶ Pengukuran digunakan untuk membangun dasar proses agar perbaikan dapat dinilai
- ▶ Metrik perangkat lunak membantu perusahaan mengembangkan estimasi proyek yang lebih baik, menghasilkan sistem berkualitas tinggi, dan mendapatkan produk yang dihasilkan tepat waktu



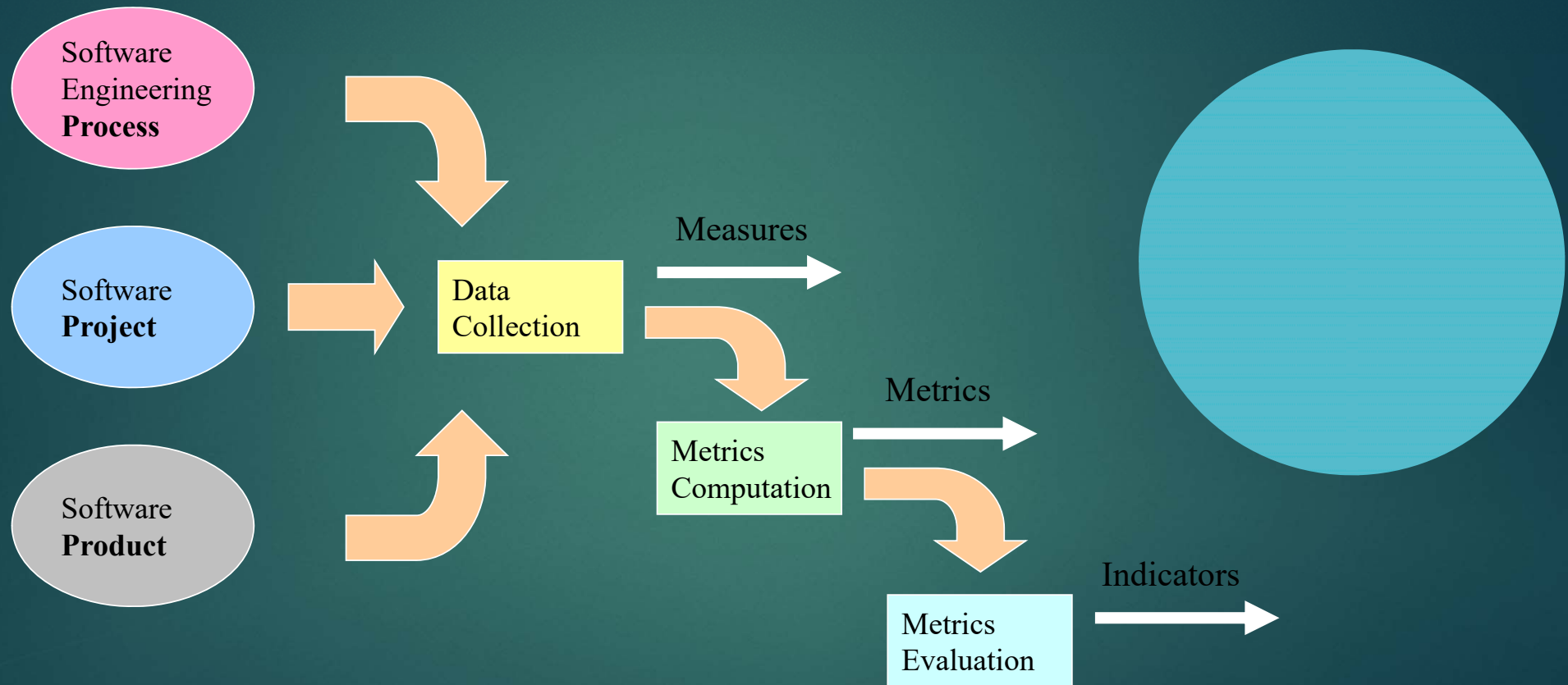
# Membangun Dasar Metrik

29


- ▶ Dengan membangun dasar metrik, manfaat bisa diperoleh pada tingkat proses, produk, dan proyek perangkat lunak
- ▶ Metrik yang sama dapat melayani banyak master
- ▶ Dasar tersebut terdiri dari data yang dikumpulkan dari proyek-proyek masa lalu
- ▶ Data dasar harus memiliki atribut sebagai berikut
  - ▶ Data harus cukup akurat (dugaan harus dihindari)
  - ▶ Data harus dikumpulkan dari beragam proyek yang dilakukan
  - ▶ Pengukuran dilakukan secara konsisten (misalnya, baris kode harus ditafsirkan secara konsisten di semua proyek)
  - ▶ aplikasi masa lalu harus sama dengan pekerjaan yang akan diperkirakan
- ▶ Setelah data dikumpulkan dan metrik dihitung, metrik harus dievaluasi dan diterapkan selama estimasi, pekerjaan teknis, pengendalian proyek, dan perbaikan proses

# Proses Dasar Metrik Perangkat Lunak

30




# Memulai Metrik.....

1. Memahami proses yang ada
  2. Menentukan tujuan yang akan dicapai dengan membentuk program metrik
  3. Mengidentifikasi metrik untuk mencapai tujuan tersebut
    - ▶ Menjaga metrik sederhana
    - ▶ Memastikan metrik menambah nilai proses dan produk
  4. Mengidentifikasi langkah-langkah yang harus dikumpulkan untuk mendukung metrik tersebut
- 

# Memulai Metrik.....

32

5. Menetapkan proses pengumpulan pengukuran
    - a) Apa sumber data?
    - b) Apakah diperlukan alat untuk mengumpulkan data?
    - c) Siapa yang bertanggung jawab untuk mengumpulkan data?
    - d) Kapan data yang dikumpulkan dan dicatat?
    - e) Bagaimana data disimpan?
    - f) Apa mekanisme validasi yang digunakan untuk memastikan data sudah benar ?
  6. Gunakan alat-alat yang tepat untuk membantu dalam pengumpulan dan penilaian
  7. Membangun database metrik
  8. Menentukan mekanisme umpan balik yang sesuai pada apa metrik menunjukkan tentang proses sehingga proses dan program metrik dapat ditingkatkan
- 



# Tugas....

33

- ▶ Carilah suatu aplikasi/sistem informasi yang akan dilakukan pengukuran metrik terhadapnya. Jelaskan tujuan dan proses dari aplikasi/sistem informasi tersebut.
- ▶ Cari dan eksplorasi-lah tools/alat bantu (freeware) pengukuran metrik yang sesuai dengan bahasa pemrograman yang digunakan pada aplikasi/sistem informasi di atas. Berikan penjelasan mengenai tools pengukuran metrik tersebut.
- ▶ Lakukan pengukuran metrik pada aplikasi/sistem informasi di atas dengan menggunakan tools yang anda pilih.



Diskusi .....

