BINUS UNIVERSITY ONLINE LEARNING

People
Innovation
Excellence

LECTURE NOTES

Software Engineering

Minggu 8

Software Project Management

And Software Metrics

LEARNING OUTCOMES

Setelah menyelesaikan pembelajaran ini, mahasiswa akan mampu:

- ☑ LO 4 – Menganalisa proyek *management software*

Outline Materi (Sub-Topic) :

1. *Team Coordination & Communication*
2. *Problem Decomposition*
3. *A Framework for Product Metrics*
4. *Metrics for the Requirements Model*
5. *Metrics for the Design Model*
6. *Design Metrics for WebApps*
7. *Code Metrics*
8. *Metrics for Testing*
9. *Maintenance Metrics*
10. *Metrics in the Process and Project Domain*
11. *Software Measurement*
12. *Metrics for Software Quality*
13. Studi kasus

ISI MATERI

1. *Team Coordination & Communication*

Koordinasi dan komunikasi sangat diperlukan di dalam proyek pembangunan software. Seorang manajer proyek ataupun software engineer tentunya harus melakukan koordinasi dan komunikasi baik secara internal atau pun eksternal.

Berdasarkan sifat dan pendekatannya koordinasi dan komunikasi tim dibagi menjadi:

- *Formal, impersonal approaches*

  Komunikasi ini bersifat formal dan dapat digunakan untuk dokumen-dokumen yang umumnya bersifat baseline dan berdampak pada target-target proyek.

  Meliputi *document software engineering* dan produk kerja (termasuk *source code*), memo teknikal, milestone proyek, jadwal dan *project control tools*, permintaan perubahan, laporan penelurusan eror, dan data *repository*.

- *Formal, interpersonal procedures*

  Komunikasi formal ini digunakan untuk hal-hal yang sifatnya koordinasi antar tim, tetapi nanti hasilnya dapat dibuatkan formal sebagai landasan untuk target-target proyek seterusnya.

  Fokus pada kegiatan penjaminan kualitas yang diterapkan dalam produk kerja *software engineering*. Hal ini meliputi status *review* meeting, perancangan dan inspeksi kode.

- *Informal, interpersonal procedures*

  Komunikasi ini bersifat informal sehingga cocok juga untuk hal-hal yang bukan kegiatan formal, misalnya meeting kelompok untuk diseminasi informasi dan pemecahan masalah. *Team building* juga menggunakan komunikasi ini.

- *Electronic communication*

  Meliputi komunikasi dalam bentuk e-mail, *electronic bulletin board*, atau *video conference.*

- *Interpersonal networking*

  Meliputi diskusi informal dengan anggota tim dan anggota di luar tim pengembang yang memiliki pengalaman yang dapat membantu anggota tim.

2. Problem Decomposition

Dekomposisi masalah dikenal juga dengan istilah partisi atau elaborasi masalah. Satu cakupan berarti:

- Didekomposisikan menjadi fungsi-fungsi
- Didekomposisikan menjadi obyek data yang dapat dilihat oleh user, atau
- Didekomposisikan menjadi sekumpulan kelas masalah

Proses dekomposisi dilanjutkan hingga semua fungsi atau masalah dapat didefinisikan. Ketika sebuah kerangka proses telah ditentukan, maka karakteristik dari proyek harus dipertimbangkan.

3. A Framework for Product Metrics

Sebuah ukuran menentukan indikasi kuantitatif dari tingkatan, jumlah, dimensi, kapasitas, atau ukuran dari suatu attribute proses atau produk. Berdasarkan glossary dari IEEE, *metric* didefinisikan sebagai ukuran

kuantitatif mengenai tingkatan dari sebuah sistem, komponen, atau proses yang dimiliki dalam atribut. Sebuah indikator adalah sebuah metric atau kombinasi memtrik yang menyediakan wawasan mengenai proses *software*, proyek *software* atau produk itu sendiri.

Karakteristik dari *attribute metric:*

- Sederhana dan dapat dihitung

  Sebuah metric seharusnya mudah dipelajari mengenai bagaimana *metric* tersebut diturunkan. Perhitungan *metric* seharusnya dapat dilakukan tanpa menyita terlalu banyak waktu dan usaha.

- Secara empiris dan intuitif *persuasive*.

  Metrik harus dapat memenuhi intuisi *engineer* mengenai atribut produk yang dipertimbangkan

- Konsisten dan obyektif

  Metric harus selalu memberikan hasil yang jelas dan tidak ambigu, serta bersifat obyektif.

- Konsisten dalam penggunaan unit dan dimensi

  Perhitungan matematika dari metric harus menggunakan ukuran yang tidak memberikan hasil yang aneh.

- Tidak bergantung pada bahasa pemrograman

  Metric harus berdasarkan pada model analisis, model perancangan atau struktur program, bukan berdasarkan bahasa pemrograman.

- Mekanisme yang efektif untuk mendapatkan *feedback* kualitas

  Metric harus menyediakan informasi yang dapat membantu dalam menghasilkan produk akhir yang berkualitas tinggi.

4. *Metrics for the Requirements Model*

a. *Metric* berdasarkan fungsi

Menggunakan *function point* sebagai *factor* normalisasi atau sebagai ukuran dari besar kecilnya suatu spesifikasi. *Function point* pertama kali diusulkan oleh Albrecht yang digunakan secara efektif sebagai alat untuk mengukur fungsionalitas sistem yang diberikan.

b. Metric berdasarkan spesifikasi

Digunakan sebagai indikasi terhadap kualitas dengan mengukur jumlah dari requirement berdasarkan tipenya.

5. *Metrics for the Design Model*

Terdapat 9 karakteristik dari perancangan berorientasi obyek:

a. Ukuran, ukuran didefinisikan melalui 4 sudut pandang: populasi, jumlah, panjang, dan fungsionalitas

b. Kompleksitas, bagaimana kelas-kelas dalam perancangan berorientasi obyek dihubungkan satu dengan yang lain.

c. *Coupling*, hubungan fisik antara elemen-elemen perancangan berorientasi obyek.

d. *Sufficiency*, ukuran dimana abstraksi memiliki *feature-feature* yang dibutuhkan, atau tingkatan dimana komponen design memiliki feature-feature dalam abstraksinya, dalam sudut pandang aplikasi saat ini.

e. Kelengkapan, implikasi secara tidak langsung mengenai tingkat abstraksi atau komponen perancangan yang dapat digunakan kembali

f. Kohesi, ukuran dimana semua opearasi dalam sebuah kelas bekerja bersama untuk mencapai satu tujuan yang telah didefinisikan dengan baik.

g. Primitiveness, dapat diterapkan pada kelas maupun operasi yang merupakan tingkatan atomic sebuah operasi atau kelas.

h. Persamaan , ukurang dimana dua atau lebih kelas dianggap mirip dari segi struktur, fungsi, behavior, atau tujuan.

i. *Volatility*, ukuran kemungkina perubahan akan muncul.

Salah satu *metric* berorientasi obyek yang dapat digunakan diusulkan oleh Chidamber and Kemerer yang dibagi menjadi beberap *metric*:

- *weighted methods per class*
- *depth of the inheritance tree*
- *number of children*
- *coupling between object classes*
- *response for a class*
- *lack of cohesion in methods*

Sedangkan Lorenz and Kidd mengusulkan beberapa metric lain untuk pengukurang berorientasi obyek:

- *class size*
- *number of operation overridden by a subclass*
- *number of operation added by a subclass*
- *specialization index*

6. Design Metrics for WebApps

Beberapa pertanyaan berikut dapat digunakan sebagai acuan dalam merancang metric dalam pengukuran aplikasi mobile atau web:

a. Apakah *user interface* yang ada mempromosikan kegunaan dari aplikasi?

b. Apakah unsur aestetik dari aplikasi layak untuk domain aplikasi dan menyenangkan user?

c. Apakah konten sudah dirancang dengan cara yang membagi sebagian besar informasi dengan usaha yang minim?

d. Apakah navigasi yang ada efisien dan langsung?

e. Apakah arsitektur dari aplikasi telah dirancang untuk mengakomodasi tujuan dari aplikasi.

f. Apakah komponen dirancang dalam cara yang mengurangi kompleksitasi procedural dan meningkatkan ketepatan, kepercayaan dan performance dari aplikasi?

7. Code Metrics

Halstead's software science: kumpulan komperhensif dari metrik semuanya memprediksi jumlah (angka atau intensitas) dari operator dan operand dalam sebuah program atau komponen.

8. Metrics for Testing

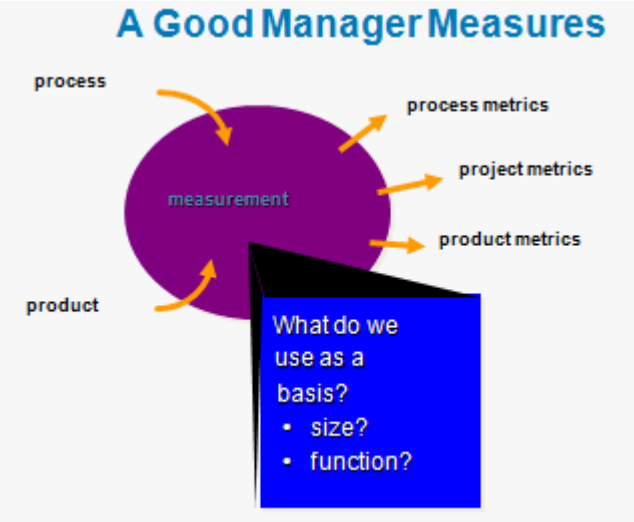
Usaha yang dilakukan untuk pengujian dapat diperkirakan dengan menggunakan metric yang diturunkan dari ukuran halstead. Berikut ini adalah metric yang dapat digunakan untuk mengukur pengaruh dari proses pengujian:

- *Lack of cohesion in methods (LCOM)*
- *Percent public and protected (PAP).*
- *Public access to data members (PAD).*
- *Number of root classes (NOR).*
- *Fan-in (FIN).*
- *Number of children (NOC) and depth of the inheritance tree (DIT).*

9. *Maintenance Metrics*

Software maturity index (SMI) merupakan salah satu *metric* yang diusulkan dalam pengukuran proses *maintenance* yang menyediakan indikasi stabilitas dari sebuah produk *software* (berdasarkan perubahan yang muncul untuk setiap produk yang dikeluarkan).

10. *Metrics in the Process and Project Domain*



Proses dalam pengembangan suatu softwar diukur secara tidak langsung. Oleh karena itu, dibutuhkan metric yang dalam mengukur sebuah proses berdasarkan hasil yang diberikan. Hasil/keluaran dapat berupa:

- *Error* yang tidak tercover sebelum softare dikeluarkan
- *Defect* yang dikirim dan dilaporkan oleh end user
- Produktifitas, jumlah produk kerja yang dikirimkan
- Usaha manusia yang dikeluarkan
- Waktu kalender yang dihabiskan
- Pendekatan jadwal

11. *Software Measurement*

Pengukuran sebuah software secara keseluruhan dapat diukur melalui ukuran dan fungsi dari software itu sendiri. Metrik berorientasi ukuran:

- *errors* per *KLOC (thousand lines of code)*
- *defects* per *KLOC*
- *$* per *LOC*
- *pages of documentation per KLOC*
- *errors* per *person-month*
- *errors* per *review hour*
- *LOC* per *person-month*
- *$* per *page of documentation*

Metrik berorientasi fungsi:

- *errors* per *FP (thousand lines of code)*
- *defects* per *FP*
- *$* per *FP*
- *pages of documentation per FP*
- *FP* per *person-month*

Perbandingan antara *Line of Code* dengan *Function Point:*

Programming Language	LOC per Function point			
	avg.	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	63	53	77	-
JavaScript	58	63	42	75
Perl	60	-	-	-
PL/1	78	67	22	263
Powerbuilder	32	31	11	105
SAS	40	41	33	49
Smalltalk	26	19	10	55
SQL	40	37	7	110
Visual Basic	47	42	16	158

12. Metrics for Software Quality

Mengukur kualitas sebuah softwar dapat dilihat dari criteria berikut:

- ***Correctness***

  Ukurang dimana program dalam melakukan operasi-operasi yang seharusnya sesuai dengan spesifikasi.

- ***Maintainability***

  Ukuran dimana program dapat melakukan penyesuaian terhadap perubahan.

- **Integrity**

  Ukuran dimana program dapat bertahan dari serangan luar

- **Usability**

  Ukuran dimana program mudah untuk digunakan oleh user.

13. Studi Kasus

Perusahaan dapat menggunakan beberapa metodologi yang ada di dalam pelaksanaan manajemen pengelolaan perangkat lunak, misalnya:

- PMBOK (Project Management Body of Knowledge)
- PRINCE2 (Project IN Controlled Environment)
- Scrum
- Extreme Programming (XP)
- Kanban

Penggunaan metode tersebut tentunya disesuaikan dengan kondisi proyek, misalnya untuk pembangungan proyek dengan kondisi requirement yang sudah dapat ditentukan di awal, maka dapat menggunakan PMBOK dan SDLC. Untuk pembangunan proyek di mana requirement belum dapat ditentukan di awal, maka cocok untuk menggunakan metode agile dengan Scrum.

Software metrics nanti juga berhubungan dengan *quality metrics*, dan ini juga dapat digambarkan sebagai KPI (*Key Process Improvement*) yang merupakan target-target yang perlu dicapai pada proyek.

DAFTAR PUSTAKA

- Pressman, R.S. (2015). ***Software Engineering : A Practioner's Approach. 8th ed***. McGraw-Hill Companies.Inc, Americas, New York. ISBN : 978 1 259 253157.
- Project Management http://www.pricesystems.com/resources/mf_risks_remedies_facts.asp
- Project Portfolio Management http://www.daptiv.com/index.htm
- SW Metrics http://www.spc.ca/resources/metrics/
- Function Point Measurement http://www.functionpoints.com
- SW Metrics service Estimation http://www.charismatek.com/_public4/html/services/pdf/service_estimate.pdf