

ANALISIS KUALITAS PERANGKAT LUNAK TERHADAP SISTEM INFORMASI STT WASTUKANCANA PURWAKARTA

SHOFA SHOFIAH HILABI

Program Studi Teknik Informatika, STT Wastukancana Purwakarta
shofashofia@gmail.com

ABSTRAK

Kebutuhan perangkat lunak di Perguruan Tinggi semakin meningkat tiap tahunnya. Perangkat lunak ini dibutuhkan dalam membantu proses bisnis yang berjalan di Perguruan Tinggi. Keberhasilan perangkat lunak yang dibangun dilihat berdasarkan sesuai atau tidaknya kerja perangkat lunak terhadap proses bisnis yang berjalan. Hal ini juga berlaku di STT WASTUKANCANA.

Pada konsep keilmuan rekayasa perangkat lunak, keberhasilan perangkat lunak tidak hanya dilihat dari kesesuaian produk yang dihasilkan terhadap kebutuhan yang ada. Keberhasilan perangkat lunak juga dilihat dari proses pengembangan perangkat lunaknya. Akan tetapi pada fakta yang ditemukan pada penelitian ini, proses pengembangan perangkat lunak kurang diperhatikan dengan baik.

Berdasarkan hasil studi literatur, kualitas perangkat lunak tidak hanya dilihat berdasarkan kesesuaian produk yang dihasilkan akan tetapi dilihat juga berdasarkan penjaminan kualitas selama proses pengembangan perangkat lunaknya. Akan tetapi pada berdasarkan hasil observasi di STT Wastukancana Purwakarta, didapat fakta bahwa belum ada proses pengawasan terhadap proses pembangunan perangkat lunaknya beserta faktor kualitas perangkat lunak pada setiap perangkat lunak yang dibangun. Oleh karena itu dilakukan identifikasi terhadap komponen penjaminan kualitas perangkat lunak yang ada di STT Wastukancana untuk mengukur kesiapan STT Wastukancana dalam membangun sebuah perangkat lunak yang berkualitas.

Kata Kunci :Penjaminan kualitas, faktor kualitas perangkat lunak,

1. PENDAHULUAN

Dalam pembangunan perangkat lunak diperlukan adanya penjaminan kualitas dalam setiap tahap daur hidup perangkat lunak. Ada beberapa karakteristik yang umum tentang kebutuhan penilaian kualitas perangkat lunak, diantaranya adalah semua proyek perangkat lunak yang baik harus memenuhi perhitungan yang tepat untuk kebutuhan dasar, semua proyek perangkat lunak mengalami performansi yang buruk terutama di dalam area-area yang penting yaitu perawatan, kehandalan, *software reuse*, dan pelatihan, dan penyebab dari performansi yang buruk tersebut adalah kurangnya definisi kebutuhan yang menunjang terbentuknya fungsional pada perangkat lunak tersebut. Dilihat dari beberapa karakteristik tersebut, diperlukan penilaian penjaminan kualitas perangkat lunak secara baik dan benar. Masing-masing faktor kualitas akan dinilai secara detil dan lengkap. STT Wastukancana adalah merupakan Institusi pendidikan yang telah menggunakan perangkat lunak sebagai alat bantu dalam melakukan kegiatan administrasinya. Perangkat lunak yang digunakan dibuat oleh salah satu divisi yang ada di lingkungan STT WASTUKANCANA.

Mengingat telah banyaknya Perangkat Lunak Sistem Informasi yang dibuat, maka tentunya perlu diperhatikan kualitas dari perangkat lunak tersebut sehingga dapat terukur performansinya dengan baik dan sesuai dengan kebutuhan. Dalam penelitian ini, maka akhirnya dapat disimpulkan suatu rumusan yaitu “Bagaimana mengalisispenjaminan kualitas perangkat lunak dari Sistem Informasi Akademik STT Wastukancana” yang dibangun baik dalam proses pembangunannya.

Adapun Maksud dari penelitian ini adalah : Membantu pihak STT Wastukancana untuk mendapatkan gambaran tentang kondisi kesiapan STT Wastukancana dalam melakukan penjaminan kualitas dari perangkat lunak yang dibangun, dan memudahkan pihak dalam menentukan komponen untuk meningkatkan kualitas perangkat lunak yang ada terutama dari segi proses pembangunan perangkat lunaknya. Sedangkan Manfaat Penelitiannya adalah menggambarkan tentang kondisi kesiapan STT Wastukancana dalam melakukan penjaminan kualitas perangkat lunak dapat digunakan untuk pengembangan divisi khusus *Software Quality Assurance (SQA)* di STT

Wastukencana sehingga perangkat lunak yang dibangun akan lebih berkualitas, sehingga komponen kualitas perangkat lunak yang telah ditentukan dapat dijadikan acuan bagi STT Wastukencana untuk pembangunan perangkat lunak berikutnya.

2. TINJAUAN PUSTAKA

Jaminan kualitas perangkat lunak (*Software Quality Assurance/SQA*) adalah aktifitas pelindung yang diaplikasikan kepada seluruh proses perangkat lunak. Di dalam proses penjaminan kualitas perangkat lunak terdapat model faktor. Model Faktor Kualitas Perangkat Lunak ini menjadi referensi utama dalam proses penjaminan kualitas perangkat lunak. Beberapa model faktor kualitas perangkat lunak dan kategorisasinya sudah diusulkan selama bertahun-tahun. Model klasik dari faktor kualitas perangkat lunak dikemukakan oleh McCall yang terdiri dari 11 faktor [McCall et al, 1977].

Model berikutnya dikemukakan oleh Deutsch dan Willis (1988) terdiri dari 12 sampai 15 faktor dan oleh Evans dan Marciniak (1987). Alternatif model tidak berbeda jauh dari model McCall. Perbedaannya terletak pada penambahan sudut pandang yang dirasa belum dinilai pada model McCall. Pembahasan ini akan terbagi menjadi dua bagian besar, yaitu:

1. Model faktor McCall

Model faktor McCall mengklasifikasi-kan semua kebutuhan perangkat lunak ke dalam 11 faktor kualitas. Kesebelas faktor tersebut dibagi menjadi tiga kategori sebagai berikut:

- a. Faktor operasi produk
- b. Faktor revisi produk
- c. Faktor transisi produk.

2. Model faktor alternatif

Selain model faktor McCall terdapat beberapa model alternatif yang merupakan perkembangan dari model McCall. Seperti yang telah disebutkan pada poin awal bahwa ada dua model alternatif yang dipergunakan, yaitu:

- a. Model Evans dan Marciniak
- b. Model Deutsch dan Willis.

Model Kualitas McCall terbagi menjadi 11 faktor kualitas. Penjelasan dari masing-masing faktor kualitas tersebut adalah sebagai berikut:

1. *Correctness*

Sebuah perangkat lunak dapat dikatakan benar jika memenuhi persyaratan sebagai berikut:

- a. Menghasilkan keluaran yang benar untuk setiap kemungkinan masukan oleh pengguna.
- b. Melakukan proses yang seharusnya (tidak kurang dan tidak berlebihan).

- c. Secara formal harus bisa dibuktikan secara matematis.

2. *Reliability*

Sudut pandang reliabilitas pada poin ini lebih menekankan pada kemungkinan dari *failure-free* suatu operasi perangkat lunak terhadap periode waktu tertentu di dalam lingkungan tertentu. Software reliability bukan fungsi langsung terhadap waktu.

3. *Efficiency*

Ada dua pengertian tentang efisiensi sebuah perangkat lunak, yaitu:

- a. Menurut McCall (1977)
Penggunaan sumber daya seperti waktu pemrosesan processor (eksekusi), pemakaian media penyimpanan (memori, space, bandwidth).
- b. Menurut ISO 9126 (1993)
Berkaitan dengan hubungan antara kinerja perangkat lunak dan jumlah sumber daya yang digunakan.

4. *Integrity*

Integritas perangkat lunak pada model McCall lebih menekankan kepada keamanan sebuah perangkat lunak. Pihak Adam M. Bachtiar, Dian Dharmayanti, Mira Kania S. Developer harus mampu melihat kebutuhan akan hak akses perangkat lunak tersebut pada setiap penggunaannya.

5. *Usability*

Usability mempunyai unsur akademis seperti psikologis, ergonomi, dan human factors [Nielsen, 1993].

6. *Maintainability*

Maintainability adalah kemudahan dari perangkat lunak untuk dipelihara, seperti:

- a. Memperbaiki kerusakan
- b. Menemukan kebutuhan baru
- c. Membuat pemeliharaan selanjutnya lebih mudah
- d. Mengatasi lingkungan yang berubah. Sebuah perangkat lunak dikatakan dapat dipelihara jika koreksi dari minor bugs memerlukan usaha yang kecil.

7. *Flexibility*

Ada dua pengertian tentang faktor fleksibilitas perangkat lunak, yaitu:

- a. Menurut McCall
Kemudahan yang didalam membuat perubahan yang dibutuhkan akibat perubahan lingkungan.
- b. Menurut Boehm
Kemampuan melakukan modifikasi kode untuk memfasilitasi perubahan yang telah ditentukan.

8. *Testability*

Testability adalah kemampuan perangkat lunak untuk diuji. Selain itu testability adalah derajat yang dimiliki sebuah sistem untuk memfasilitasi kriteria pengujian dan performansi dari pengujian

tersebut untuk mengukur sejauh mana kriteria tersebut dipenuhi [IEEE, 1990].

9. *Portability*

Perangkat lunak dikatakan portabel jika biaya untuk memindahkannya (*transport* dan adaptasi) ke lingkungan yang baru lebih kecil jika dibandingkan dengan biaya untuk membangun perangkat lunak tersebut dari awal.

10. *Reusability*

Reusability adalah properti dari perangkat lunak yang memungkinkan perangkat lunak atau modul-modulnya digunakan kembali untuk sistem lain. Suatu perangkat lunak dikatakan *Reusable* yang baik jika modul-modulnya dapat digunakan kembali untuk aplikasi lainnya.

11. *Interoperability*

Interoperability adalah kemampuan suatu perangkat lunak untuk bekerja dengan perangkat lunak lainnya tanpa mengalami kesulitan.

Model Kualitas Alternatif ada beberapa faktor kualitas yang akan dibahas di sini, antara lain:

1. *Verifiability*

Menggambarkan semudah apa memverifikasi performa dari suatu program. Beberapa sub faktor pada verifiability adalah sebagai berikut:

- Coding and documentation guidelines* berfokus untuk memberikan panduan dalam menuliskan kode dalam berbagai bahasa pemrograman dan petunjuk untuk mendokumentasikan suatu perangkat lunak dengan baik.
- Compliance (Complexity)* berfokus untuk menjaga kompleksitas kode program yang dibangun sehingga tingkat verifikasi tetap terjaga.
- Document Accessibility* berfokus terhadap kemudahan untuk mengakses dokumentasi yang sudah disebutkan pada sub bab sebelumnya.
- Traceability* berfokus terhadap kemudahan developer untuk melakukan penelusuran suatu dokumentasi yang dimiliki oleh perangkat lunak tersebut.
- Modularity* berfokus kepada kefleksibelan suatu Majalah Ilmiah STT WASTUKANCANA. Yaitu kemandirian fungsional dari suatu komponen program.
- Generality* adalah seberapa bisa perangkat lunak tersebut bisa menyelesaikan masalah pada domainnya.
- Simplicity* mengenai tingkat dimana perangkat lunak dapat dimengerti tanpa kesulitan.
- Safety* dapat didefinisikan sebagai kemampuan untuk memperkecil resiko yang dapat membahayakan ke tingkat /level yang dapat diterima. Safety dapat didefinisikan sebagai kemampuan untuk melakukan: Identifikasi, Analisis, Mempelajari, Mengontrol Terhadap software hazard atau fungsi berbahaya (data &

command) untuk memastikan melakukan operasi yang aman [NASA Software Assurance]. Safety dapat dipecah menjadi bagian: Pertama Identifikasi, mencari dan menentukan hazard yang mungkin terjadi. Kedua Analisis, menganalisa hazard yang ditemukan untuk mengetahui resiko yang dapat terjadi. Ketiga Mempelajari, mempelajari hasil analisa untuk mencari solusi yang dapat digunakan dan terakhir mengontrol, mengontrol hazard yang telah ditemukan untuk meminimalisasi resiko yang mungkin terjadi.

- Manageability* adalah Manageability dapat didefinisikan sebagai kemampuan untuk melakukan tindakan administrasi, melakukan pengawasan serta memperoleh informasi yang relevan dengan tindakan yang terkait. Beberapa kaitan *manageability* antara lain: *Monitoring: Berkaitandengan aktifitas pemantauan* (termasuk pencatatan)

- Tracking* : Berkaitan dengan aktifitas penelusuran
- Control* : Dengan aktifitas pengendalian perubahan.
- Survivability* Terdapat dua pengertian untuk *survivability*, yaitu: Keandalan sistem untuk memberikan layanan ketika terkena bencana, dan keandalan sistem diukur dari lamanya waktu failure dan lamanya waktu recovery.

Beberapa langkah yang bisa dilakukan untuk pengendalian bencana, yaitu:

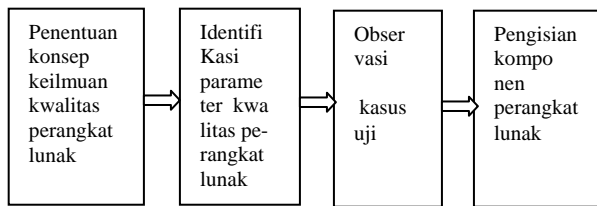
- Identify the Business Continuity Components That You Will Focus On (people, property, system, data).*
- Define What You're Protecting.*
- Prioritize Business Functions.*
- Classify Outage Types, Frequencies, and Duration.*
- Calculate The Cost of Downtime.*

3. METODOLOGI PENELITIAN

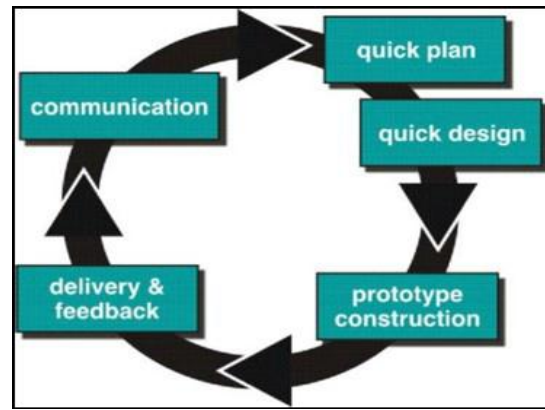
Jenis penelitian yang dilakukan dalam pelaksanaan penelitian ini adalah penelitian analisis (*analytical research*). Jenis penelitian analisis adalah jenis penelitian yang melibatkan konsep ilmu pengetahuan dalam menilai suatu kasus. Penelitian analisis memilih konsep yang akan dianalisis yang berhubungan dengan masalah yang ada pada suatu organisasi atau lingkungan.

Metode Penelitian: Metodologi penelitian yang digunakan pada penelitian ini adalah : Penentuan Konsep Keilmuan Kualitas Perangkat Lunak, Identifikasi Parameter Kualitas Perangkat Lunak, Observasi Kasus Uji, Pengisian komponen kualitas perangkat lunak.

Ilustrasi dari metode penelitian ini dapat dilihat pada Gambar 1.



Gambar-1 Metodologi Penelitian



Gambar 2. Model Proses Prototyping

4. HASIL DAN PEMBAHASAN

Hasil Observasi Terhadap Sistem Informasi STT Wastukancana. STT Wastukancana adalah merupakan salah satu institusi pendidikan yang dalam kegiatan sehari-harinya menggunakan perangkat lunak sebagai alat bantu dalam melakukan kegiatannya. Perangkat lunak yang ada di STT Wastukancana dibangun sendiri oleh Divisi ICT STT Wastukancana. Perangkat lunak yang dihasilkan oleh divisi ICT STT Wastukancanamerupakan perangkat lunak Sistem Informasi yang umumnya berbasis Web. Berikut ini adalah beberapa contoh dari aplikasi yang ada di lingkungan STT Wastukancana:

1. Sistem Informasi Akademik (SIMAK STT)
2. Sistem Informasi Penerimaan Mahasiswa Baru
3. Perwalian/Pengisian KRS Online
4. KHS mahasiswa Online
5. Bagian Administrasi dan Keuangan
6. Kegiatan kemahasiswaan
7. Alumni Online
8. Perkuliahan Online
9. Perpustakaan Online
10. Kegiatan kemahasiswaan Online
11. Manajemen Aset/Inventaris
12. Jejaring Sosial (grup Facebook, Twitter)
13. Blog STT Wastukancana
14. Dosen dan Karyawan.
15. Informasi beasiswa.

Model proses yang sering digunakan STT Wastukancana dalam mengembangkan perangkat lunak adalah model *prototyping* dan *incremental*. Ilustrasi dari model *prototyping*, dapat di lihat pada gambar di bawah ini.

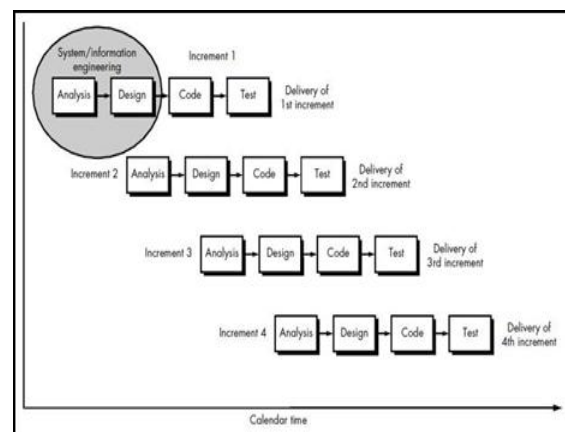
4.1. Hasil Pencocokan Komponen Penjaminan Kualitas Perangkat Lunak di STT Wastukancana

Setelah melakukan observasi terhadap perangkat lunak yang dibangun dan kondisi yang ada di Direktorat ICT dan Multimedia, hal berikutnya adalah mengisi borang untuk mengukur sejauh mana kesiapan STT Wastukancana dalam melakukan penjaminan kualitas perangkat lunak yang dibangun pada Perencanaan strategis sistim Infomasi yang terdapat dalam borang STT yang digunakan dibentuk berdasarkan konsep penjaminan kualitas perangkat lunak yang telah dipelajari.

4.2 Fakta Kepedulian STT Wastukancana Terhadap SQA

Dari hasil wawancara dan observasi yang kami lakukan di STT Wastukancana didapatkan fakta sebagai berikut: Di STT Wastukancana belum memiliki tim sendiri untuk mengurus masalah penjaminan kualitas perangkat lunak, akan tetapi sudah ada divisi tersendiri untuk membangun produk/perangkat lunak, yaitu

Direktorat ICT dan Multimedia. Berdasarkan kondisi tersebut maka kondisi organisasi Penjamin Kualitas PL dapat dijelaskan pada Tabel 1.



Gambar 3. Model Proses Incremental

Tabel-1. Organisasi Penjamin Kualitas Perangkat Lunak

SQA Orgnisasi	Ketersediaan	Penjelasan
SQA manajemen	Tidak ada	---
SQA Unit	Tidak ada	---
SQA Trustees	Tidak ada	---
SQA commiteess	Tidak ada	---
SQA Forums	Tidak ada	Forum khusus SQA belum ada namun pada saat akan membangun perangkat lunak dilakukan rapat koordinasi yang dilakukan pada akhir tahap SDLC.

Tabel 2. Komponen Infrastruktur Kualitas PL

SQA Infrastructure	Ketersediaan	Penjelasan
SQA Procedures	Ada	SQA procedure yang digunakan berupa prosedur maupun standar yang dibuat oleh STT Wastukanca na sendiri yang berorientasi pada produk bukan pada proses SDLC .
SQA Supporting Devices	Tidak ada	-----
SQA Training Instructions	Tidak ada	-----
SQA Infrastructure	Ketersediaan	Penjelasan
SQA Preventive Actions	Tidak ada	-----
SQA Configuration Managements	Tidak ada	-----
SQA Documentation Control	Tidak ada	-----

Tabel 3. Manajemen Kualitas PL

SQA Manajemen	Komponen	Ketersediaan
SQA Project Progress Control	Pengontrolan aktifitas manajemen resiko	Tidak ada
	Pengontrolan jadwal proyek.	Tidak ada
	Pengontrolan sumber daya proyek	Tidak ada
	Pengontrolan biaya proyek.	Tidak ada
SQA Project Progress Control	Metrics process	Tidak ada
	Metrics product	Tidak ada
	Timetable	Tidak ada
	Efektifitas penghilangan error	Tidak ada
	Produktifitas proses software	Tidak ada
	Help desk services	Tidak ada
	Corrective maintenance services	Tidak ada
SQA Cost	SQ Costs	Tidak ada
	Prevention (CC)	Tidak ada
	Appraisal (CC)	Tidak ada
	Manajerial Preparation control cost (C and C)	Tidak ada
	Internal failure costs (FoCC)	Tidak ada
	External failure costs (FoCC)	Tidak ada
Managerial failure costs (FoCC)	Tidak ada	

4.3. Evaluasi SQA STT Wastukanca

Dari fakta yang didapat dilakukan evaluasi di beberapa hal. Hasil evaluasi tersebut adalah sebagai berikut:

1. Tidak tersedianya tim tersendiri untuk mengurus masalah QA di STT Wastukanca, tetapi STT Wastukanca memiliki Divisi khusus yang menangani pembangunan Perangkat Lunak. Berdasarkan kondisi diatas maka jumlah personil tim SQA kurang memenuhi standar dan tidak ada pembagian pekerjaan yang jelas. Di STT Wastukanca terkadang 1 orang anggota divisi bertugas lebih dari 1 pekerjaan.
2. Organisasi penjaminan perangkat lunak masih perlu dibentuk.
3. Manajemen kualitas pada STT Wastukanca masih banyak yang belum terpenuhi.
4. Template dokumen yang digunakan merupakan hasil kustomisasi dari setiap kebutuhan perangkat lunak di STT Wastukanca.
5. Tidak ada *Coding Standard* yang harus ditaati programmer sehingga mempersulit kerja dalam tim.

6. Belum ada perhatian khusus terhadap model kualitas perangkat lunak. Baik model Mc Call maupun model alternatif.

5. Kesimpulan dan Saran

Dari pembahasan persoalan penanganan kualitas perangkat lunak di STT Wastukencana dapat kami tarik kesimpulan sebagai berikut:

1. STT Wastukencana belum memiliki tim SQA yang independen untuk menjamin kualitas setiap perangkat lunak yang akan dikembangkan.
2. Penerapan SQA di STT Wastukencana sangat berbeda dengan konsep kualitas perangkat lunak yang ada.
3. Untuk mencapai kinerja yang lebih baik, STT Wastukencana harus membentuk Organisasi SQA secara khusus, sehingga perangkat lunak yang dihasilkan terjamin kualitasnya.

Sedangkan saran pada penelitian ini adalah sebagai berikut:

1. Standar borang yang digunakan untuk mengukur kesiapan penjaminan kualitas perangkat lunak pada penelitian ini masih didasarkan pada konsep secara umum sehingga perlu dilakukan penelitian lebih lanjut untuk isian borang yang digunakan dalam pengukuran tersebut.
2. Pengukuran kesiapan penjaminan kualitas perangkat lunak pada penelitian ini dilakukan terhadap perangkat lunak yang sudah dibangun sehingga dibutuhkan penelitian lebih lanjut untuk melakukan pengukuran terhadap pembangunan perangkat lunak yang baru akan dibangun untuk menilai kesiapan dari segi proses.

DAFTAR PUSTAKA

Adler, Paul S. *Practice and Process: The Socialization Software Development*. 2006.

D,Galin, *Software Quality Assurance from Theory to Implementation*, Eng-land: Pearson, 2004.

Eko,R,Indrajit,*KiatMemilih Software*. EbizzAsia, 2004.

Program Studi Teknik Informatika STT Wastukencana, *Perencanaan Strategis Sistim Informasi STT Wastukencana*, Purwakarta, 2012.

McCall,Et al,*software engineerin*, 1977.

Pressman, Roger S. *RekayasaPerangkatLunak (PendekatanPraktisi)*. Yogyakarta: Penerbit Andi, 2002.