

## ANALISIS KUALITAS PERANGKAT LUNAK TERHADAP SISTEM INFORMASI UNIKOM

ADAM MUKHARIL BACHTIAR, DIAN DHARMAYANTI, MIRA KANIA SABARIAH  
Program Studi Teknik Informatika – Fakultas Teknik dan Ilmu Komputer  
Universitas Komputer Indonesia

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 UNIKOM.

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 UNIKOM, 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 yang ada di UNIKOM untuk mengukur kesiapan UNIKOM dalam membangun sebuah perangkat lunak yang berkualitas.

**Kata Kunci** - Penjaminan kualitas, faktor kualitas perangkat lunak, SQA

### 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, di antaranya adalah semua proyek perangkat lunak yang baik harus memenuhi perhitungan yang tepat untuk kebutuhan dasar, semua proyek perangkat lunak menderita 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.

Universitas Komputer Indonesia (UNIKOM) 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 UNIKOM. 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 dilakukan peninjauan kesiapan UNIKOM dalam melakukan penjaminan kualitas perangkat lunak yang dibangun baik dalam proses pembangunannya maupun hasil perangkat lunak yang dibangunnya.

## TINJAUAN PUSTAKA

### Model Faktor 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 semua kebutuhan perangkat lunak ke dalam 11 faktor kualitas. Kesebelas faktor tersebut dibagi menjadi tiga kategori sebagai berikut:

- Faktor operasi produk
- Faktor revisi produk
- 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:

- Model Evans dan Marciniak
- Model Deutsch dan Willis.

### Model Kualitas McCall

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:

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

- **Menurut McCall (1977)**  
Penggunaan sumber daya seperti waktu pemrosesan processor (eksekusi), pemakaian media penyimpanan (memori, space, bandwidth).
- **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

developer harus mampu melihat kebutuhan akan hak akses perangkat lunak tersebut pada setiap penggunaannya.

#### 5. Usability

Faktor ini melihat dari kemudahan perangkat lunak untuk digunakan dan dipelajari. Usability mempunyai unsur akademis seperti psikologis, ergonomi, dan human factors [Nielsen, 1993].

#### 6. Maintainability

Maintainability adalah kemudahan dari perangkat lunak untuk dipelihara, seperti:

- Memperbaiki kerusakan
- Menemukan kebutuhan baru
- Membuat pemeliharaan selanjutnya lebih mudah
- 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:

- **Menurut McCall**  
Kemudahan yang didalam membuat perubahan yang dibutuhkan akibat perubahan lingkungan.
- **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

*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

sistem. Mempunyai 5 kriteria, yaitu:

- *Decomposability*
- *Composability*
- *Understandability*.
- *Continuity*.
- *Protection*.

## 2. *Expandability*

*Expandability* adalah kemampuan sebuah perangkat lunak untuk dikembangkan. Beberapa sub faktor dalam *expandability* antara lain:

### *Extensibility*

*Extensibility* adalah kemampuan sistem untuk dapat ditambahkan suatu modul tanpa harus menimbulkan efek samping yang tidak diinginkan. Beberapa kriteria yang bisa digunakan untuk menilai tingkat

### *Modularity*

Kemandirian suatu fungsional dari suatu komponen program.

### *Generality*

Seberapa bisa perangkat lunak tersebut bisa menyelesaikan masalah pada domainnya.

### *Simplicity*

Tingkat dimana perangkat lunak dapat dimengerti tanpa kesulitan.

## 3. *Safety*

*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:

- Identifikasi, mencari dan menentukan hazard yang mungkin terjadi.
- Analisis, menganalisa hazard yang ditemukan untuk mengetahui resiko yang dapat terjadi

- Mempelajari, mempelajari hasil analisa untuk mencari solusi yang dapat digunakan
- Mengontrol, mengontrol hazard yang telah ditemukan untuk meminimalisasi resiko yang mungkin terjadi

## 4. *Manageability*

*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*

Berkaitan dengan aktifitas pemantauan (termasuk pencatatan)

### *Tracking*

Berkaitan dengan aktifitas penelusuran

### *Control*

Berkaitan dengan aktifitas pengendalian / perubahan

## 5. *Survivability*

Terdapat dua pengertian untuk *survivability*, yaitu:

- Kehandalan sistem untuk memberikan layanan ketika terkena bencana.
- Kehandalan sistem diukur dari lamanya waktu failure dan lamanya waktu recovery.

Beberapa langkah yang bisa dilakukan untuk pengendalian bencana, yaitu:

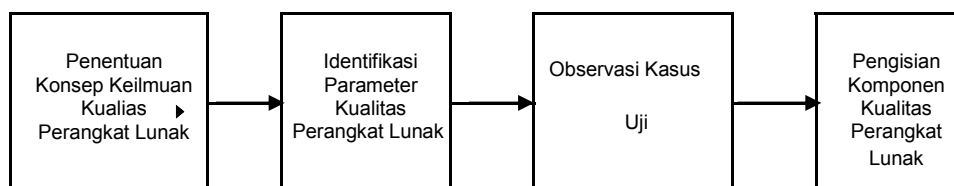
- *Identify the Business Continuity Components That You Will Focus On (people, property, system, data).*
- *Define What You're Protecting.*
- *Prioritize Business Functions.*
- *Classify Outage Types, Frequencies, and Duration.*
- *Calculate The Cost of Downtime*

## TUJUAN DAN MANFAAT PENELITIAN

### Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Membantu pihak UNIKOM untuk



Gambar-1 Metodologi Penelitian

mendapatkan gambaran tentang kondisi kesiapan UNIKOM dalam melakukan penjaminan kualitas dari perangkat lunak yang dibangun.

2. Memudahkan pihak UNIKOM dalam menentukan komponen untuk meningkatkan kualitas perangkat lunak yang ada terutama dari segi proses pembangunan perangkat lunaknya.

#### Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Gambaran tentang kondisi kesiapan UNIKOM dalam melakukan penjaminan kualitas perangkat lunak dapat digunakan untuk pengembangan divisi khusus SQA di UNIKOM sehingga perangkat lunak yang dibangun akan lebih berkualitas.
2. Komponen kualitas perangkat lunak yang telah ditentukan dapat dijadikan acuan bagi UNIKOM untuk pembangunan perangkat lunak berikutnya.

#### METODOLOGI PENELITIAN

##### Jenis Penelitian

Jenis penelitian 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 :

1. Penentuan Konsep Keilmuan Kualitas Perangkat Lunak
2. Identifikasi Parameter Kualitas Perangkat Lunak
3. Observasi Kasus Uji
4. Pengisian Komponen Kualitas Perangkat Lunak.

Ilustrasi dari metode penelitian ini dapat dilihat pada Gambar 1.

#### HASIL DAN PEMBAHASAN

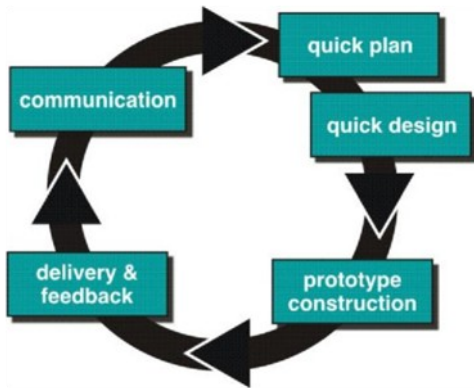
##### Hasil Observasi Terhadap Sistem Informasi UNIKOM

UNIKOM 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 UNIKOM dibangun sendiri oleh Divisi ICT UNIKOM. Perangkat lunak yang dihasilkan oleh divisi ICT UNIKOM merupakan perangkat lunak Sistem Informasi yang umumnya berbasis Web. Berikut ini adalah beberapa contoh dari aplikasi yang ada di lingkungan UNIKOM:

1. Sistem Informasi Akademik (SIKAD)
2. Sistem Informasi Penerimaan Mahasiswa Baru
3. Perwalian/Pengisian KRS
4. Nilai Online
5. Keuangan
6. Autodebet Online

- 7. Alumni Online
- 8. Kuliah Online
- 9. Perpustakaan Online
- 10. Dosen Online
- 11. Manajemen Aset/Inventaris
- 12. Jejaring Sosial
- 13. Blog UNIKOM
- 14. Dosen dan Karyawan
- 15. Sistem Informasi Lowongan Kerja
- 16. Informasi beasiswa.

Model proses yang sering digunakan UNIKOM dalam mengembangkan perangkat lunak adalah model *prototyping* dan *incremental*. Ilustrasi dari model *prototyping* dapat dilihat pada Gambar 2.



Gambar 2. Model Proses Prototyping

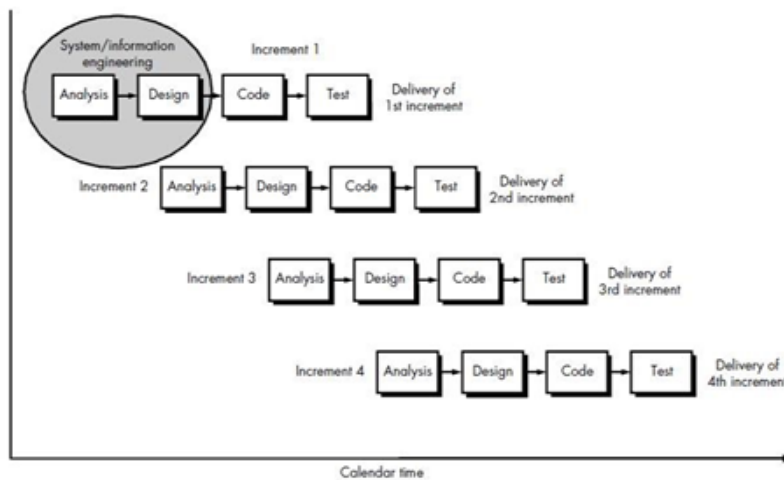
**Hasil Pencocokan Komponen Penjaminan Kualitas Perangkat Lunak di UNIKOM**

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 UNIKOM dalam melakukan penjaminan kualitas perangkat lunak yang dibangun. Borang yang digunakan dibentuk berdasarkan konsep penjaminan kualitas perangkat lunak yang telah dipelajari.

**Fakta Kepedulian UNIKOM Terhadap SQA**

Dari hasil wawancara dan observasi yang kami lakukan di UNIKOM didapatkan fakta sebagai berikut:

1. Di UNIKOM 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 Organization	Ketersediaan	Penjelasan
SQA Management	Tidak ada	-
SQA Unit	Tidak ada	-
SQA Trustees	Tidak ada	-
SQA Committees	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 UNIKOM sendiri yang berorientasi pada produk bukan pada proses SDLC.
SQA Supporting Devices	Tidak ada	-
SQA Training Instructions	Tidak ada	-
SQA Preventive Actions	Tidak ada	-
SQA Configuration Managements	Tidak ada	-
SQA Documentation Control	Tidak ada	-

2. Komponen infrastruktur kualitas perangkat lunak pada UNIKOM ditunjukkan pada Tabel 2.
3. Manajemen kualitas perangkat lunak pada UNIKOM belum ada. Hal ini ditunjukkan oleh Tabel 3 berikut ini:
4. Beberapa fakta lain yang berkaitan dengan kepedulian terhadap SQA yang ditemukan di UNIKOM, yaitu:
  - UNIKOM tidak memiliki *Coding Standard* yang harus ditaati oleh *programmer*.
  - Sering terjadi *roundtrip* (bolak-balik pada tahap desain dan coding untuk mencari desain perangkat lunak yang tepat diimplementasikan pada lingkungan bahasa pemrograman dan teknologi yang digunakan) pada tahap *development* sehingga memperlambat waktu untuk

Tabel 3. Manajemen Kualitas PL

SQA Management	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
SQ Metrics	Metrics process	Tidak ada
	Metrics product	Tidak ada
	Timetable	Tidak ada
	Efektifitas penghilangan error	Ada
	Produktifitas proses software	Tidak ada
	Help desk services	Tidak ada
	Corrective maintenance services	Ada
SQ Costs	Prevention (CC)	Tidak ada
	Appraisal (CC)	Tidak ada
	Managerialpreparation 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

menyelesaikan pekerjaan.

- UNIKOM menerapkan testing standard berdasarkan *best practice* yang ditentukan oleh UNIKOM dan *customer* (tidak mengikuti standarisasi internasional tertentu). Pengujian dan dokumentasinya dikerjakan oleh developer itu sendiri.
- UNIKOM tidak menganggarkan *appraisal cost* (biaya lembur dan bonus di akhir proyek) untuk karyawannya. Bonus hanya akan ada jika ada kelebihan budget saja.
- Pola komunikasi pada proses Quality Control hanya dilakukan di lingkungan internal direktorat ICT.
- Tidak ada SOP umum yang

terdefiniskan secara jelas untuk SQA.

- SOP yang diterapkan UNIKOM adalah fase planning dan *delivery software*.
- Defect dan bug pada perangkat lunak tidak dicatat. *Defect* dan *bug* yang telah ditambah juga tidak didokumentasikan.
- Masalah-masalah pada seluruh tahap pembangunan perangkat lunak tidak didokumentasikan.
- Tidak ada kontrol terpusat pada perusahaan untuk mendokumentasikan semua permasalahan yang terjadi di perusahaan tersebut. Salah satu contoh: Jika client ingin menelepon karena mempunyai masalah dengan perangkat lunak



yang telah dibangun UNIKOM maka nomor telfon yang dihubungi bukanlah nomor telepon UNIKOM akan tetapi langsung ke nomor telepon PM proyek bersangkutan.

- Ada *training* untuk *programmer* yang baru bergabung dengan tim dan *programmer* lama jika ada teknologi baru yang dibutuhkan. *Training* dilakukan sesuai dengan kebutuhan saja.

### Evaluasi SQA UNIKOM

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 UNIKOM tetapi UNIKOM 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 UNIKOM terkadang 1 orang anggota divisi bertugas lebih dari 1 pekerjaan.
2. Organisasi penjaminan perangkat lunak masih perlu dibentuk.
3. Manajemen kualitas pada UNIKOM masih banyak yang belum terpenuhi.
4. Template dokumen yang digunakan merupakan hasil kustomisasi dari setiap kebutuhan perangkat lunak di UNIKOM.
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.

### KESIMPULAN DAN SARAN

#### Kesimpulan

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

1. UNIKOM tidak memiliki tim SQA yang independen untuk menjamin kualitas setiap perangkat lunak yang

dikembangkan.

2. Penerapan SQA di UNIKOM sangat berbeda dengan konsep kualitas perangkat lunak yang ada.
3. Untuk mencapai kinerja yang lebih baik, UNIKOM harus membentuk Organisasi SQA secara khusus, sehingga perangkat lunak yang dihasilkan terjamin kualitasnya.

#### Saran

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

- [1] D. Galin, Software Quality Assurance from Theory to Implementation, England: Pearson, 2004
- [2] Program Studi Teknik Informatika UNIKOM, Borang Akreditasi Program Studi Teknik Informatika UNIKOM, Bandung, 2012

