

Standar Penilaian Kualitas Perangkat Lunak

Studi Kasus Pembangunan dan Pengelolaan Aplikasi Pemerintah Provinsi Jawa Barat

Muhamad Deni Hendriawan¹, Hadi Prasetyo Utomo², Agus Iim Suryana³

Magister Teknik Informatika, Pascasarjana, Universitas Langlangbuana Bandung^{1,2,3}
1250200017.mdenih@gmail.com, hadi@informatika.unla.ac.id, agus.iim.suryana@unla.ac.id

Abstrak— Penilaian kualitas perangkat lunak merupakan aspek krusial dalam pembangunan dan pengembangan perangkat lunak untuk memastikan bahwa perangkat lunak yang dihasilkan memenuhi standar yang ditetapkan serta kebutuhan pengguna. Berbagai model *software quality assurance* dapat digunakan untuk menilai kualitas perangkat lunak. Model-model tersebut memberikan kerangka kerja yang lengkap untuk mengevaluasi berbagai atribut kualitas termasuk fungsionalitas, keandalan, kegunaan, efisiensi, pemeliharaan, dan portabilitas. Penelitian ini membahas proses membandingkan model-model *software quality assurance*, mengidentifikasi kebutuhan, serta memilih model yang tepat untuk diimplementasikan pada penilaian kualitas perangkat lunak di lingkungan obyek penelitian yaitu pemerintahan Provinsi Jawa Barat. Penelitian ini juga memberikan rekomendasi dalam mengadopsi standar penilaian yang efektif. Penerapan standar yang konsisten dapat meningkatkan keandalan dan kepuasan pengguna, serta mengurangi biaya pemeliharaan jangka panjang.

Pengembangan SPBE pasal 9 ayat 2 poin g yaitu melaksanakan penjaminan mutu aplikasi (quality assurance). Penelitian ini melakukan pemilihan model Software Quality Assurance yang paling sesuai dengan kebutuhan untuk diterapkan di lingkungan obyek penelitian yaitu Pemerintah Provinsi Jawa Barat.

II. METODE

Penelitian ini menggunakan metode *action research*, dimana penelitian dilakukan dengan menggabungkan tindakan (intervensi dalam sistem) dengan refleksi yang bertujuan untuk meningkatkan praktik atau memecahkan masalah praktis terhadap permasalahan di obyek penelitian.

Penelitian dilakukan dengan melakukan pendekatan yang dinamis dan fleksibel, beradaptasi dan merespons kebutuhan serta situasi yang berkembang selama proses penelitian pada obyek penelitian.

Kata kunci— model, perangkat lunak, *software*, kualitas, standar

I. PENDAHULUAN

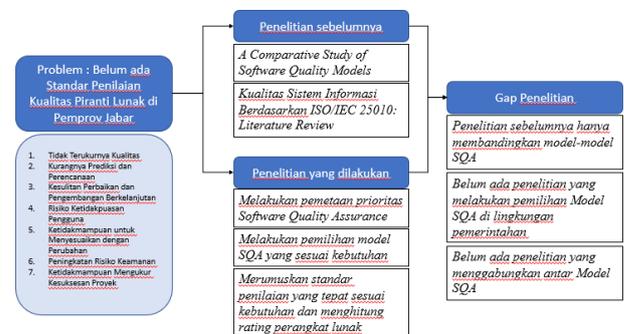
Menurut Suman & Manoj Wadhawa (2014) kualitas suatu produk penting bagi pengguna dan pengembang karena *user* ingin bekerja dengan piranti lunak berkualitas baik dan juga penting bagi pengembang untuk reputasinya. Kepentingan terhadap Software Quality Assurance didasarkan kepada 3 (tiga) hal :

1. Meningkatnya kepedulian terhadap kualitas perangkat lunak
2. Ketidakberwujudan Perangkat Lunak
3. Mengumpulkan error selama Pembngunan Perangkat Lunak

Sebaliknya, sebuah perangkat lunak jika tidak melalui pengukuran kualitas perangkat lunak, dapat timbul beberapa dampak dan risiko yang dapat mempengaruhi kinerja, keberlanjutan, dan keberhasilan perangkat lunak tersebut. Berikut adalah beberapa konsekuensi yang mungkin terjadi:

1. Tidak Terukurnya Kualitas;
2. Kurangnya Prediksi dan Perencanaan;
3. Kesulitan Perbaikan dan Pengembangan Berkelanjutan;
4. Risiko Ketidakpuasan Pengguna;
5. Ketidakmampuan untuk Menyesuaikan dengan Perubahan;
6. Peningkatan Risiko Keamanan;
7. Ketidakmampuan Mengukur Kesuksesan Proyek.

Sementara itu di lingkungan obyek penelitian yaitu Pemerintah Provinsi Jawa Barat saat ini belum ada standar penilaian perangkat lunak, sementara penilaian perangkat lunak sudah diamanatkan pada Pergub Jabar Nomor 13 Tahun 2020 mengenai Rencana Induk



Gambar. 1 Metodologi Penelitian

III. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Hasil dari polling yang dilakukan kepa 25 responden yang mewakili seluruh bidang dan level pegawai, menunjukkan kebutuhan akan penilaian kualitas perangkat lunak pada elemen-elemen SQA sebagai berikut :

TABEL I
HASIL PEMILIHAN ELEMEN SQA

No	Elemen	Suara
1	Testing	13
2	Standar	11
3	Review and Audit	10

Jika disandingkan dengan kesesuaian pada standar teknis Pembangunan Aplikasi Pemprov Jabar, maka dihasilkan tabel perbandingan sebagai berikut :

TABEL II
ELEMEN SQA PADA STANDAR TEKNIS

Standar	arsitektur, <i>database, platform, containerization, source code management, CI/CD</i> , disain template, <i>observability, object storage</i> , dan dokumentasi.
Testing	Uji Fungsionalitas, Uji Integrasi, Uji Beban, dan Uji Keamanan.
Review and Audit	Monitoring evaluasi

Selama ini untuk pengukuran yang sudah dilaksanakan di Pemerintah Daerah Provinsi Jawa Barat adalah melakukan checklist terhadap kelengkapan sesuai elemen yang ada pada dokumen standar teknis.

TABEL III
CHECKLIST KELENGKAPAN STANDAR TEKNIS

No	Elemen	Kelengkapan	
		Ada/Sesuai	Tidak ada/ Tidak sesuai
1	Arsitektur		
2	Database		
3	Platform		
4	Containerization		
5	Source code management		
6	CI/CD		
7	Disain template		
8	Observability		
9	Storage object		
10	dokumentasi		
11	Uji Fungsionalitas		
12	Uji Integrasi		
13	Uji Beban		
14	Uji Keamanan		

Dengan demikian, dapat dilihat bahwa pengukuran Standar sudah dilakukan, dimana tugas *Software Quality Assurance* pada elemen standar adalah untuk memastikan bahwa standar yang telah diimplementasikan diikuti, dan bahwa semua produk kerja sesuai dengan standar yang telah ditetapkan.

Selain itu, juga untuk pelaksanaan testing juga telah dilakukan, dengan melaksanakan beberapa testing yaitu Uji Fungsionalitas (User Acceptance Test), Uji Integrasi, Uji Beban, dan Uji Keamanan (penetration test).

Sehingga penelitian akan berfokus kepada bagaimana pemilihan model untuk melakukan review dengan implementasinya berupa penilaian rating atau kualitas perangkat lunak yang dibangun, dikarenakan untuk pengukuran dari sisi kualitas belum dilakukan, hanya melakukan pengecekan apakah sudah sesuai dengan dokumen standar teknis, atau sudah dilakukan sesuai dengan dokumen standar teknis.

B. Pemilihan Model

Software Quality Assurance memiliki beberapa framework yang digunakan untuk mengukur, menilai, dan meningkatkan kualitas perangkat lunak. Framework ini dikenal dengan istilah Software Quality Models. Tujuan dari model-model ini adalah memberikan

pendekatan sistematis untuk mengelola dan memastikan kualitas perangkat lunak selama siklus hidup pengembangan.

Pada penelitian ini dipilih 3 model yang akan dilakukan perbandingan, yaitu ISO 9126, BOEHM, dan FURPS/FURPS+. Pada perbandingan pertama dilakukan dengan membandingkan karakteristik yang ada pada masing-masing model. Hal ini dilakukan untuk mendapatkan data model mana yang mendekati kebutuhan sesuai standar teknis.

TABEL IV
PERBANDINGAN MODEL SQA

Faktor/ Karakteristik	Boehm	FURPS / FURPS+	ISO 9126
Efficiency	√	√	√
Functionality		√	√
Maintainability		√	√
Portability	√		√
Reliability	√	√	√
Usability			√
Testability	√		
Understandability	√	√	
Modifiability	√		

Proses selanjutnya dengan mempelajari masing-masing karakteristik model, didapat sebagai berikut :

TABEL V
KARAKTERISTIK MODEL ISO 9126

No	Karakteristik	Sub Karakteristik	Penjelasan
1.	Functionality	Suitability	Sistem harus mampu melakukan tugas yang diperlukan
		Accurateness	Sistem harus mampu memberikan hasil sesuai harapan
		Interoperability	Sistem harus mampu berinteraksi dengan sistem lain
		Security	Sistem harus mampu mencegah akses yang tidak sah
2.	Reliability	Maturity	Sistem harus mampu menghilangkan sebagian besar kesalahan (error) seiring berjalannya waktu
		Fault tolerance	Sistem harus mampu menangani error
		Recoverability	Sistem harus mampu kembali bekerja dan memulihkan data yang hilang setelah error terjadi
3.	Usability	Understandability	pengguna harus dapat memahami cara menggunakan sistem dengan mudah
		Learnability	pengguna harus dapat belajar menggunakan sistem dengan mudah
		Operability	pengguna harus dapat menggunakan sistem tanpa banyak effort
		Attractiveness	Antarmuka aplikasi harus terlihat bagus

4.	Efficiency	Time Behaviour	Sistem harus memberikan respon dengan cepat
		Resource Utilisation	sistem harus daapt efisien dalam memanfaatkan sumber daya
5.	Maintainability	Analysability	kesalahan pada system harus dapat didiagnosis dengan mudah
		Changeability	System harus dapat dimodifikasi dengan mudah
		Stability	Sistem harus tetap dapat berfungsi Ketika terjadi perubahan
		Testability	Sistem harus dapat diuji dengan mudah
6.	Portability	Adaptability	Sistem harus dapat dipindahkan ke environment/perangkat lain
		Installability	Sistem harus dapat diinstall dengan mudah
		Conformance	Sistem harus mematuhi standar portabilitas
		Replaceability	Sistem harus dapat dengan mudah menggantikan system lain
7.	Semua karakteristik terdapat sub karakteristik Compliance		Sistem harus mematuhi undang-undang atau peraturan yang ada

TABEL VI
 KARAKTERISTIK MODEL FURPS/FURPS+

No	Karakteristik	Sub Karakteristik	Penjelasan
1.	Functionality	Audit	Sistem harus dapat dilakukan review dengan mudah
		Hak akses	Sistem harus mampu mengatur hak akses
		Helpdesk	Tersedia saluran untk berinteraksi dengan user
		Security	Sistem harus aman dari serangan
		Report	Sistem harus dapat memberikan report berkala
		System Management	Sistem manajemen harus berjalan dengan baik
		Workflow	Alur kerja system harus berjalan dengan baik
2.	Reliability	Availability	Sistem harus tersedia ketika diakses
		Accuration	Sistem harus dapat mengeluarkan output sesuai dengan yang diharapkan
		System Recovery	Sistem harus dengan mudah dapat dikembalikan Ketika terjadi kesalahan
3.	Usability	Accesability	Sistem harus dapat diakses dengan mudah
		UI aesthetic	System harus menyediakan user

			interface yang baik dan mudah digunakan
		UI Consistency	System harus menyediakan user interface yang memudahkan user dan tidak membingungkan
4.	Efficiency	Output	System harus dapat menghasilkan output dengan mudah
		Time Behaviour	Sistem harus memberikan respon dengan cepat
		Time Recovery	Sistem harus dapat melakukan recovery dengan mudah Ketika terjadi error
5.	Understandability/ Maintainability	Maintainability	Maintain system harus dapat dilakukan dengan mudah
		Adaptability	System harus dapat beradaptasi dengan perangkat yang berbeda
		Compatibility	Sistem harus tetap dapat berfungsi di berbagai perangkat
		Testability	Sistem harus dapat diuji dengan mudah
		Configurability	System harus dapat dengan mudah dilakukan konfigurasi
		Installability	Sistem harus dengan mudah dapat diinstall
		Scaleability	Sistem harus dapat dilakukan perubahan skala dengan mudah

TABEL VII
 KARAKTERISTIK MODEL BOEHM

No	Level 1	Level 2	Penjelasan
1.	Portability		System harus dapat dengan mudah berjalan pada perangkat yang berbeda
2.	As is utility	Reliability	Sistem harus dapat memberikan hasil yang lengkap, akurat, dan konsisten
		Efficiency	Sistem harus efisien dalam pemanfaatan perangkat, dan kemudahan akses
		Human Engineering	Sistem harus dengan mudah dapat berkomunikasi dan mudah diakses
3.	Maintainability	Testability	Sistem harus dapat dilakukan test terutama untuk aksesibilitas, komunikatif, struktur, dan self-deskriptif
		Understandability	Sistem harus mudah dipahami dari sisi konsistensi, struktur aplikasi, self deskriptif, dan keterbacaan

		Modifiability	Sistem harus dengan mudah dapat dimodifikasi dari sisi struktur dan dapat dilakukan dengan mudah penambahan dari berbagai hal
--	--	---------------	---

Hasil perbandingan ketiga karakteristik model yang dipilih, maka didapat kesesuaian dengan standar teknis sebagai berikut :

TABEL VIII
 KARAKTERISTIK MODEL BOEHM

No.	Elemen Standar Teknis Pemprov Jabar	ISO 9126	BOEHM	FURPS/FURPS+
1	Arsitektur	Ya	Ya	Ya
2	Database			
3	Platform	Ya	Ya	Ya
4	Containerization	-	-	-
5	Source code management	Ya	-	-
6	CI/CD	Ya	Ya	Ya
7	Disain template	Ya	-	Ya
8	Observability	Ya	-	Ya
9	Storage object	-	-	-
10	Dokumentasi	-	-	-
11	Uji Fungsionalitas	Ya	Ya	Ya
12	Uji Integrasi	Ya	-	-
13	Uji Beban	Ya	-	-
14	Uji Keamanan	Ya	-	checklist
	Jumlah Kesesuaian	10 item	4 item	7 item

Hasil padanan antara ketiga model dengan Standar Teknis dan Pedoman Pembangunan/Pengembangan Aplikasi Pemprov Jabar menunjukkan bahwa model ISO 9126 yang paling banyak memenuhi kebutuhan dan paling banyak kesesuaian dengan standar teknis. Untuk itu kami mengambil kesimpulan bahwa model 9126 yang akan digunakan untuk menjadi standar penilaian kualitas perangkat lunak.

C. Menentukan Rating Perangkat Lunak

Untuk menilai kualitas sebuah perangkat lunak (rating) menggunakan model Software Quality Assurance ISO 9126 dilakukan dengan melakukan penilaian kepada setiap karakteristik dari ISO 9126.

1. Menghitung functionality

Uji functionality digunakan untuk mengetahui apakah aplikasi sesuai dengan kebutuhan pengguna. Instrumen untuk uji fungsionalitas sebuah aplikasi dapat melibatkan responden yang akan mengisi tabel instrument uji fungsionalitas aplikasi.

2. Menghitung reliability

Uji reliability dapat dilakukan dengan menggunakan perangkat lunak bantuan untuk melakukan tes load, stress dan performa pada sebuah aplikasi. Pengujian dilakukan dengan metode constant dimana jumlah virtual user ketika tes dilakukan tidak berubah-ubah.

3. Menghitung usability

Uji usability dapat menggunakan kuesioner yang telah menjadi standar internasional yaitu USE Questionnaire yang dikembangkan STC Usability and User Experience Community dari Arnold M.Lund . Terdapat 4 komponen uji yaitu Usefulness, Ease of Use, Easy of learning, dan Satisfaction.

4. Menghitung efisiensi

Pengujian matriks pada efficiency merupakan ukuran seberapa banyak sumber daya komputasi dan baris kode program yang diperlukan sebuah perangkat lunak dalam menjalankan fungsinya. Pengujian efisiensi dapat menggunakan aplikasi bantu seperti GMetrix. Hasil uji menggunakan aplikasi bantu akan didapatkan nilai efisiensi dari aplikasi pengujian efisiensi.

5. Menghitung portability

Pengujian portability dilakukan dengan melakukan ujicoba aplikasi terhadap beberapa perangkat ataupun platform yang berbeda.

6. Menghitung maintainability

Menghitung/mengukur karakter maintainability dilakukan dengan cara melakukan wawancara terhadap pengelola aplikasi, bagaimana tingkat kesulitan maintain aplikasi.

IV. SIMPULAN

Setelah dilakukannya penelitian mengenai studi dan analisis untuk mendapatkan model terbaik untuk pengukuran standar penilaian kualitas perangkat lunak dengan mengambil objek kerja di Pemerintah Daerah Provinsi Jawa Barat, maka dapat diambil kesimpulan sebagai berikut :

1. Penilaian kualitas perangkat lunak di lingkungan pemerintahan dapat dilakukan dengan mengaplikasikan model Software Quality Assurance;
2. Didapatkan model terbaik untuk penilaian kualitas perangkat lunak di lingkungan pemerintahan yaitu dengan menggunakan model ISO 9126;
3. Terhadap elemen Standar Teknis yang tidak ada padanan pada model, dapat dilakukan penilaian dengan checklist kelengkapan.

Untuk penelitian lebih lanjut, kami menyampaikan beberapa saran :

1. Perlu disusun format standar untuk melakukan checklist terhadap perangkat lunak yang sudah dibangun;
2. Perlu ada simulasi kombinasi dari beberapa model yang dipilih, sehingga dapat memperkaya model penilaian yang telah ada;
3. Perlu adanya penelitian lebih lanjut mengenai pengembangan model yang sudah ada dengan mengikuti perkembangan teknologi.

REFERENSI

- [1] Suman and Manoj Wadhwa, *A Comparative Study of Software Quality Models*, International Journal of Computer Science and Information Technologies, Vol. 5 (4), 2014 . 5634-5638
- [2] Made Dwi Mulyawan, I Nyoman Satya Kumara, Ida Bagus Alit Swamardika, dan Komang Oka Saputra, *Kualitas Sistem Informasi Berdasarkan ISO/IEC 25010: Literature Review*, Majalah Ilmiah Teknologi Elektro, Vol. 20, No.1, Januari – Juni 2021.
- [3] Sanjaya Alamsyah dan Humaningsih, “*Analisis Kualitas dan Penerapan Software Quality Assurance Pada Website Lembaga Kursus Menggunakan Model ISO 9126*” Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK) Volume 3 Nomor 1, 22 Agustus 2019, ISSN : 2581-2327.
- [4] Kothuri Parashu Ramulu, Dr. B.V. Ramana Murhtyr, “*Importance of Software Quality Models In Software Engineering*”, International Journal of Engineering Technologies and Management Research, 5(3), 200-218
- [5] José P. Miguel , David Mauricio and Glen Rodríguez, “*A Review of Software Quality Models for the Evaluation of Software Products*” International Journal of Software Engineering & Applications (IJSEA), Vol.5, No.6, November 2014.