

## Survei Teknik-Teknik Pengukuran Kualitas Perangkat Lunak

Muhammad Andryan Wahyu Saputra<sup>1,\*</sup>, Wildan Alif Rioditama<sup>2</sup>, Hanis Setyowati<sup>3</sup>, Muhammad Ainul Yaqin<sup>4</sup>

Jurusan Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim, Indonesia

<sup>1</sup>18650030@student.uin-malang.ac.id; <sup>2</sup>18650018@student.uin-malang.ac.id; <sup>3</sup>18650125@student.uin-malang.ac.id;

<sup>4</sup>yaqinov@ti.uin-malang.ac.id

### INFO ARTIKEL

#### Sejarah Artikel

Diterima: 21 Desember 2020

Direvisi: 12 Februari 2021

Diterbitkan: 30 April 2021

#### Kata Kunci

Teknik Pengukuran Kualitas

Perangkat Lunak

Kualitas Perangkat Lunak

### ABSTRAK

Tujuan dari penelitian ini adalah menganalisis teknik teknik pengukuran untuk kualitas dari perangkat lunak. Dengan menggunakan data yang berasal dari perpustakaan digital dan pola survey *System Literature Review* (SLR) sehingga paper ini menjabarkan berbagai jenis software quality model yang ada beserta komponen-komponen yang digunakan dalam melakukan penilaian dari masing-masing model. Hasil penelitian ini berupa perbandingan dari beberapa teknik yang digunakan dalam mengukur kualitas perangkat lunak. Berdasarkan survei yang telah dilakukan maka dapat disimpulkan bahwa pengukuran kualitas perangkat lunak dapat dilakukan dengan menggunakan salah satu dari berbagai model yang ada. Pada penggunaan model Boehm dan ServQual, kekurangan yang ada sama yaitu kurangnya kriteria yang digunakan. Akan tetapi model Boehm dapat mempertimbangkan kualitas dari sudut pandang yang berbeda, dan model ServQual menitik beratkan pada faktor kualitas.

## PENDAHULUAN

Penggunaan teknologi informasi sangat berkaitan erat dalam pengembangan bisnis dewasa ini, faktor penting dalam perusahaan yang mulai bergantung dengan perangkat lunak. Saat ini perangkat lunak sudah menjadi kekuatan yang menentukan, dan menjadi mesin yang mengendalikan pengambilan keputusan di dalam dunia bisnis. Jaminan kualitas perangkat lunak merupakan aktivitas mendasar bagi banyak bisnis yang menghasilkan produk yang akan digunakan oleh user baik internal perusahaan maupun eksternal. Untuk memperoleh kualitas perangkat lunak yang diharapkan, mengevaluasi kualitas produk suatu perangkat lunak merupakan elemen kritis dari jaminan perangkat lunak sehingga dapat merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean [1]. Beberapa organisasi, seperti ISO dan IEEE, mencoba untuk membuat standar kualitas software dengan mengkombinasikan model dan mengaitkan karakteristik dan sub-karakteristik quality model. Beberapa penelitian juga mengusulkan metrik perangkat lunak sebagai alat untuk mengukur source code program, arsitektur, dan kinerja dari perangkat lunak [2].

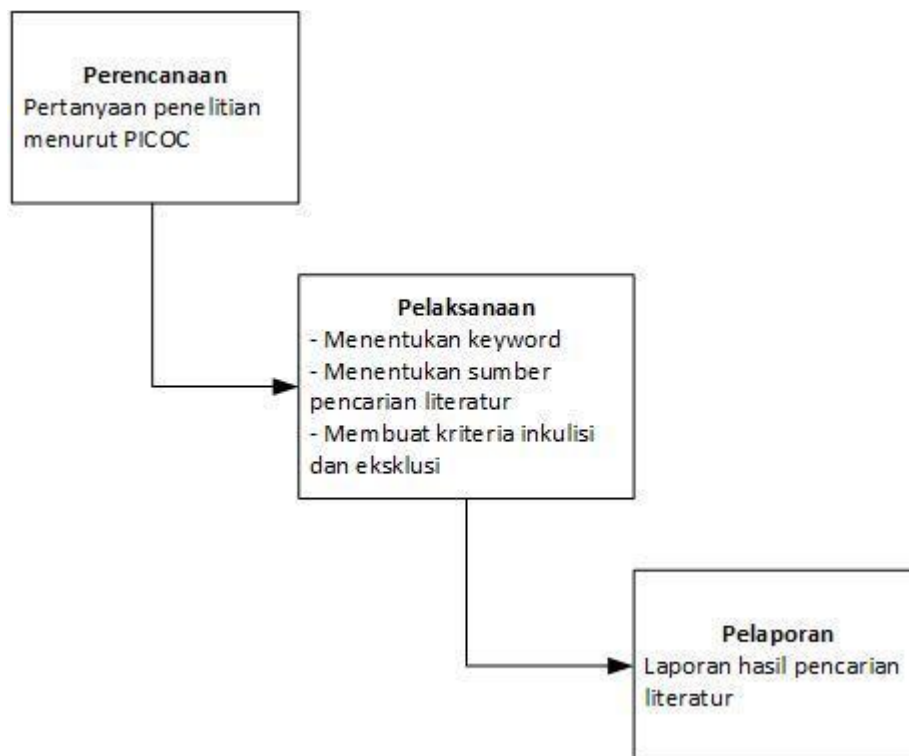
Sampai saat ini proses penilaian kualitas perangkat lunak tetap menjadi masalah terbuka dengan banyak model karena belum jelas dan belum adanya kesepakatan hubungan antara software quality model serta hubungan antara model dan metrik [3]. Masalah yang umum dalam software quality model adalah mencari model optimal dan memadai untuk memenuhi tujuan peningkatan kualitas perangkat lunak. Maka, berkaitan dengan hal tersebut diperlukan survey teknik teknik pengukuran kualitas perangkat lunak.

Berdasarkan uraian, peneliti menganalisis beberapa teknik atau model yang sudah didapatkan dari sumber digital library dan beberapa sumber lainnya. Maka peneliti menggunakan pola survei *System Literature Review* (SLR) sehingga paper ini menjabarkan

berbagai jenis software quality model yang ada beserta komponen-komponen yang digunakan dalam melakukan penilaian dari masing-masing model. Penelitian ini berfokus pada analisis teknik-teknik pengukuran kualitas software. Penelitian ini memberikan hasil berupa pembagian teknik perangkat lunak berdasarkan subjektif atau objektif dengan melihat dari faktor-faktor kualitas perangkat lunak. Kelebihan penelitian ini dibanding penelitian lainnya ialah penelitian ini mengumpulkan beberapa metode dari teknik pengukuran kualitas perangkat lunak, menjanjikan berbagai faktor dari setiap metode. Beberapa paper lain hanya mengupas satu metode saja atau hanya membahas tentang pengertian setiap metode, maka penelitian ini memperbandingkan dari berbagai faktor di metode dari beberapa teknik pengukuran kualitas software.

## METODE

Penelitian ini menggunakan metode *Systematic literature review* atau sering disingkat SLR yang dalam bahasa Indonesia disebut tinjauan pustaka sistematis adalah metode *literature review* yang mengidentifikasi, menilai, dan menginterpretasi seluruh temuan-temuan pada suatu topik penelitian, untuk menjawab pertanyaan penelitian (*research question*) yang telah ditetapkan sebelumnya [4]. Gambar 1 merupakan tahapan dari *Systematic literature review*.



Gambar 1. Tahapan SLR (*Systematic literature review*)

### **Planning (Perencanaan)**

Metode penelitian yang pertama yaitu perencanaan. *Research Question* (RQ) digunakan untuk menentukan proses pencarian literatur. Analisis dan sintesis data, sebagai hasil dari SLR, adalah jawaban RQ yang sudah ditentukan. Formula RQ harus berdasarkan lima elemen atau PICOC. Berikut tabel PICOC dari penelitian yang ditampilkan dalam Tabel 1 yang menunjukkan kriteria PICOC:

Tabel 1. Kriteria PICOC

PICOC	Kriteria	Penelitian
P	<i>Population</i> (target dari penelitian)	Teknik-teknik pengukuran kualitas software
I	<i>Intervention</i> (aspek detail dari penelitian)	Kriteria dari masing-masing teknik pengukuran kualitas software
C	<i>Comparison</i> (membandingkan hasil <i>intervention</i> )	Membandingkan kriteria dari teknik-teknik pengukuran kualitas software
O	<i>Outcomes</i> (hasil dari penelitian)	Teknik pengukuran kualitas software yang paling efektif untuk digunakan
C	<i>Context</i> (lingkungan dari penelitian)	Di bidang akademik dan industri yang memerlukan teknik pengukuran kualitas software

### Conducting (Pelaksanaan)

Tahapan ini berisi pelaksanaan SLR. diawali dengan menentukan *keyword* pencarian literatur berdasarkan PICOC yang telah direncanakan. Pemahaman terhadap sinonim dan alternatif kata akan menentukan akurasi pencarian literatur. *Keyword* dalam penelitian ini yaitu:

(“*method*” atau “*how*” atau “*model*” atau “*technique*” atau “*step*”) dan (“*measurement*” atau “*development*”) dan (“*quality of software*” atau “*evaluation of the quality of the software*”) dan (“*software*”)

Kemudian menentukan sumber (*digital library*) dari pencarian literatur. Yaitu, *ACM Digital Library*, *Research Gate* and *Science Direct*. Berikut Tabel 2 yang menampilkan sumber pencarian literatur:

Tabel 2. Sumber Pencarian Literatur

Perpustakaan Digital	Halaman WEB	Penggunaan <i>Keyword</i>	Batasan Pencarian
ACM <i>Digital Library</i> (ACM DL)	dl.acm.org	Normal	Berdasarkan waktu publikasi
<i>Research Gate</i>	researchgate.net	Normal	Memiliki judul yang relevan terhadap penelitian
<i>Science Direct</i> (SD)	www.sciencedirect.com	Normal	Memiliki judul yang relevan terhadap penelitian

Setelah mendapatkan literatur, kemudian memilih literatur yang sesuai. Agar mudah dalam proses menyeleksi, maka membuat kriteria (*inclusion and exclusion criteria*). Tabel 3 menerangkan ciri dari kriteria inklusi dan eksklusi.

Tabel 3. Kriteria Inklusi dan Eksklusi

Kriteria Inklusi	Kriteria Eksklusi
Membahas mengenai teknik teknik pengukuran kualitas software	Diterbitkan sebelum tahun 2000
Membahas mengenai faktor faktor kualitas software	Tidak menjelaskan metode teknik pengukuran kualitas software
Membahas cara mengukur kualitas software	Tidak menjelaskan detail dari teknik pengukuran kualitas software
Membahas implementasi dari teknik kualitas software	Tidak membahas implementasi teknik pengukuran kualitas software

### Reporting (Pelaporan)

Tahapan terakhir dari metode SLR yaitu pelaporan. Pencarian literatur dalam penelitian ini berlangsung selama 10 hari. Tercatat mulai tanggal 09 sampai 18 Februari 2020. Setelah mendapat literatur dan menganalisisnya berdasarkan kriteria inklusi dan eksklusi, berikut hasilnya yang dimasukkan dalam Tabel 4 yaitu hasil dari pencarian literatur:

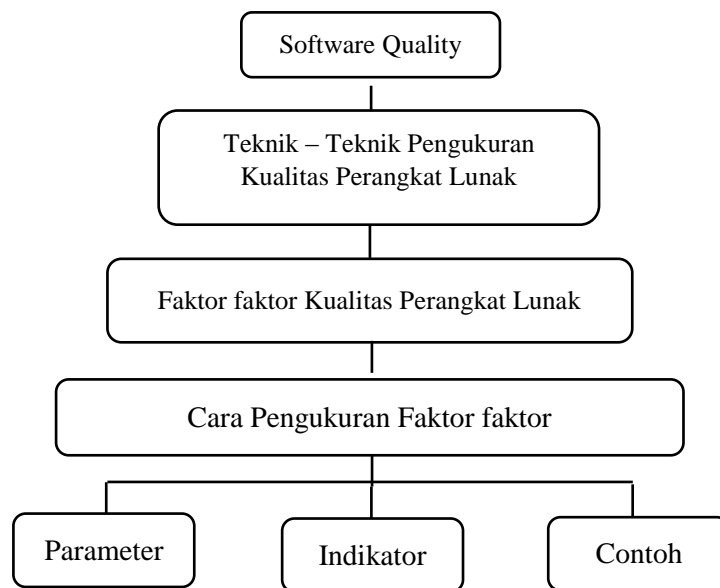
Tabel 4. Hasil Pencarian Literatur

	ACM	Research Gate	SD
Literatur yang didapat	5	15	5
Literatur setelah di analisis	1	9	4

## HASIL DAN PEMBAHASAN

Setiap Langkah awal untuk memahami kualitas perangkat lunak adalah dengan menjawab pertanyaan penting yang sering ditanyakan, yaitu: apa itu kualitas? Ada dua kubu besar ketika membahas makna dan definisi kualitas perangkat lunak [5] :

1. Kesesuaian dengan spesifikasi (objektif): kualitas yang didefinisikan sebagai materi produk dan layanan yang terukur dimana memenuhi karakteristik spesifikasi tetap yaitu, kesesuaian dengan spesifikasi yang sebelumnya didefinisikan.
2. Memenuhi kebutuhan pelanggan (subjektif): kualitas yang diidentifikasi independen dari setiap karakteristik terukur. Artinya, kualitas didefinisikan sebagai kemampuan produk atau jasa untuk memenuhi harapan pelanggan baik secara eksplisit atau tidak.

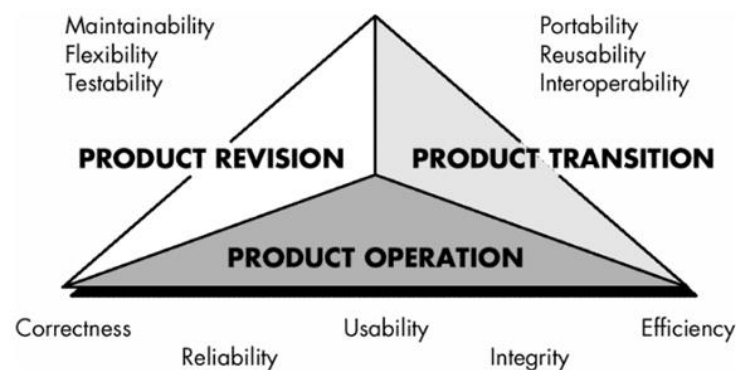


Gambar 2. Taksonomi Software Quality

Gambar 2 merupakan alur proses penelitian dimulai dengan cara mengelompokkan teknik teknik pengukuran kualitas perangkat lunak dan dari teknik teknik tersebut diambil faktor faktor dari setiap teknik teknik kemudian diambil lagi bagaimana cara mengukur faktor tersebut dimana cara pengukuran faktor tersebut dijabarkan melalui parameter, indikator, dan contoh pengukuran.

### Teknik-teknik Pengukuran Kualitas Software

#### McCall Model (1977)

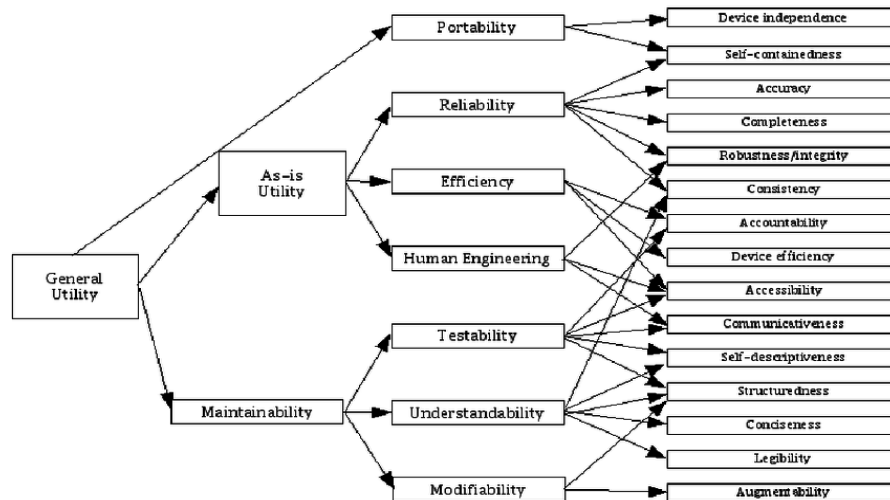


Gambar 3. McCall Model

Model McCall mencoba untuk menjembatani kesenjangan antara pengguna dan pengembang dengan berfokus pada sejumlah faktor kualitas perangkat lunak yang mencerminkan pandangan pengguna dan prioritas pengembang [6]. Gagasan utama dalam model McCall adalah untuk menilai relativitas hubungan sosial antara faktor-faktor kualitas eksternal dan kriteria kualitas produk. Model ini dikembangkan oleh angkatan udara Amerika Serikat pada sistem keputusan elektronik (Electronic System Decision), pusat pengembangan Rome Air (Rome Air Development Center) dan General Electric (GE), dengan tujuan meningkatkan kualitas produk perangkat lunak [7]. Salah satu kontribusi

besar dari model McCall adalah hubungan antara karakteristik kualitas dan metrik, walaupun terdapat kritik bahwa tidak semua metrik adalah obyektif. Salah satu aspek yang tidak dipertimbangkan langsung oleh model ini adalah fungsionalitas dari produk perangkat lunak [8]. Gambar 3 diatas merupakan penjabaran dari faktor faktor pada model McCall.

### Boehm Model (1978)



Gambar 4. Boehm Model

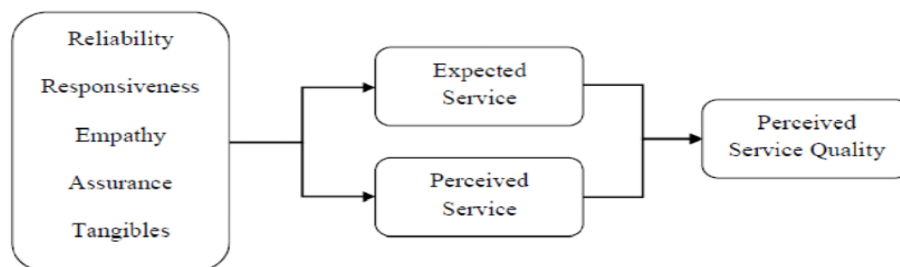
Model Boehm menambahkan beberapa karakteristik pada model McCall dengan penekanan pada pemeliharaan produk perangkat lunak. Model Boehm serupa dengan model McCall dalam merepresentasikan struktur hirarkis karakteristik, yang masing-masing memberikan kontribusi terhadap kualitas keseluruhan.

Tiga level tertinggi pada model kualitas Boehm adalah [9]:

- As-is utility: jangkauan/batasan penggunaan software.
- Maintainability: kemudahan dilakukan perubahan ketika terdapat modifikasi dan uji coba ulang.
- Portability: kemudahan software beradaptasi dengan lingkungan baru

Gagasan dari model Boehm mencakup kebutuhan pengguna, seperti pada model McCall, namun model Boehm hanya memuat diagram tanpa adanya saran tentang pengukuran karakteristik kualitas [8]. Gambar 4 menjelaskan tentang faktor faktor dari Model Boehm.

### ServQual Model (1988)

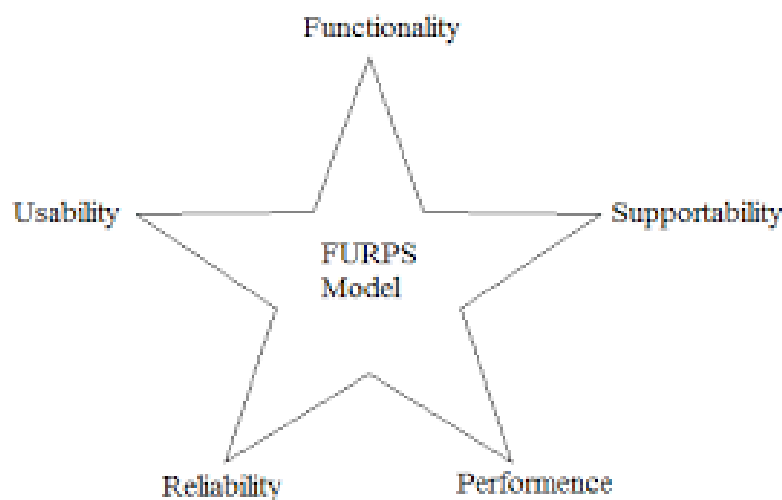


Gambar 5. ServQual Model

Gambar 5 menjelaskan tentang faktor faktor dari Model Servqual. Kuesioner SERVQUAL dibagi menjadi dua bagian, yaitu bagian pertama yang terdiri dari item-item yang menunjukkan harapan pengguna terhadap layanan web yang diberikan sedangkan bagian kedua berisi item-item yang menunjukkan persepsi dari pengguna layanan [10].

- Usability : Tampilan web perguruan tinggi XYZ tidak membingungkan Web perguruan tinggi XYZ dapat ditampilkan dengan sempurna pada browser yang berbeda-beda.
- Information Quality : Informasi di dalam web perguruan tinggi bermanfaat, lengkap, dan jelas
- Service Reliability : Web perguruan tinggi XYZ dapat dengan cepat aktif kembali jika terjadi kerusakan Semua form-form isian yang ada di dalam web berfungsi dengan baik Email atau kontak dalam keadaan aktif Tersedianya pemberitahuan jika ada informasi terbaru
- Assurance : Keamanan web perguruan tinggi XYZ baik Staf pengelola web perguruan tinggi XYZ dapat diandalkan dalam mengelola web.
- Emphaty : Pengguna merasa nyaman dalam mencari informasi di web perguruan tinggi XYZ

### FURPS Model (1992)

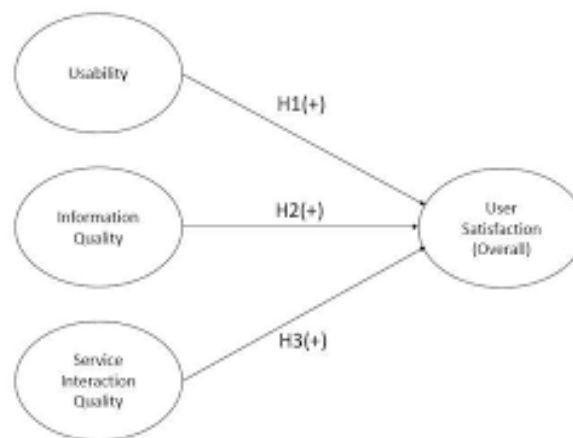


Gambar 6. FURPS Model

Model FURPS diusulkan oleh Robert Grady dan Hewlett-Packard Co. Model ini menguraikan karakteristik dalam dua kategori yang berbeda dari persyaratan (requirement) [11], yaitu:

- Fungsional Requirement (F): Ditetapkan oleh input dan output yang diharapkan.
- Non-Fungsional Requirements (URPS): Kegunaan (usability), kehandalan (reliability), kinerja (performance), daya dukung (supportability).

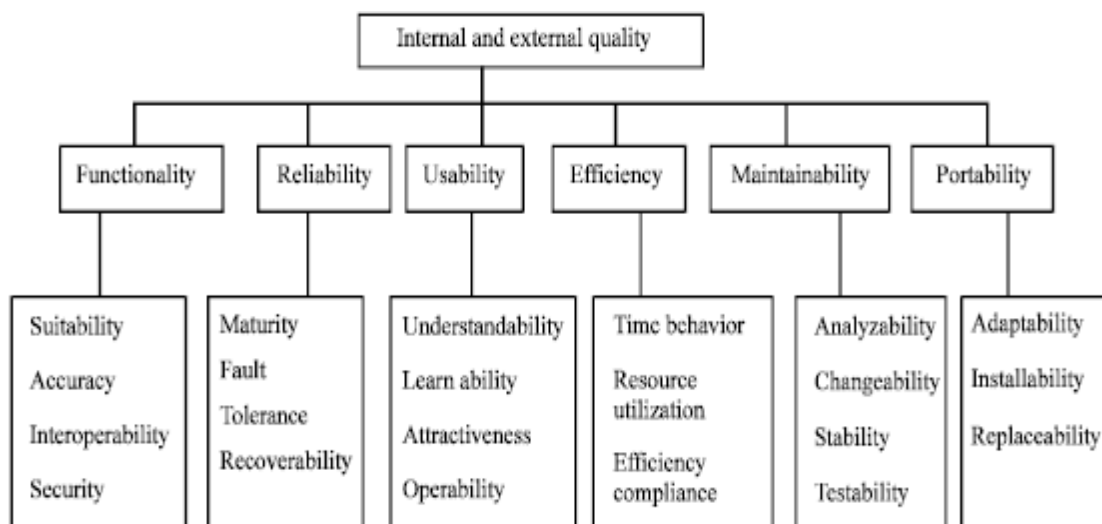
Model FURPS kemudian diperluas oleh IBM Rational Software - menjadi FURPS +. Perluasan (+) yang dilakukan meliputi persyaratan untuk batasan desain, persyaratan implementasi, persyaratan antarmuka dan persyaratan fisik. Salah satu kelemahan dari model FURPS adalah model ini tidak mempertimbangkan portabilitas dari produk perangkat lunak [11]. Gambar 6 menjelaskan tentang faktor faktor dari Model FURPS.

**WebQual 4.0 Model (1998)**

Gambar 7. WebQual Model

Gambar 7 menjelaskan tentang faktor faktor dari Model WebQual. Kualitas dalam tiga dimensi dari WebQual versi 4.0 menekankan pada sebuah sistem dimana pengguna merasa puas dan kembali menggunakan website itu kembali.

- Kualitas kegunaan (usability quality) yang menitik beratkan kepada kegunaan dari sebuah website. Dimensi ini berkaitan dengan rancangan website seperti tampilan website, kemudahan penggunaan, navigasi, dan gambaran yang disampaikan kepada pengguna.
- Kualitas informasi (information quality) yang menitik beratkan pada kualitas atau mutu isi yang terdapat pada website.
- Kualitas interaksi pelayanan (service interaction quality) yang menitikberatkan kepada kualitas atau mutu interaksi pelayanan terhadap pengguna [12].

**ISO 9126 Model (2001)**

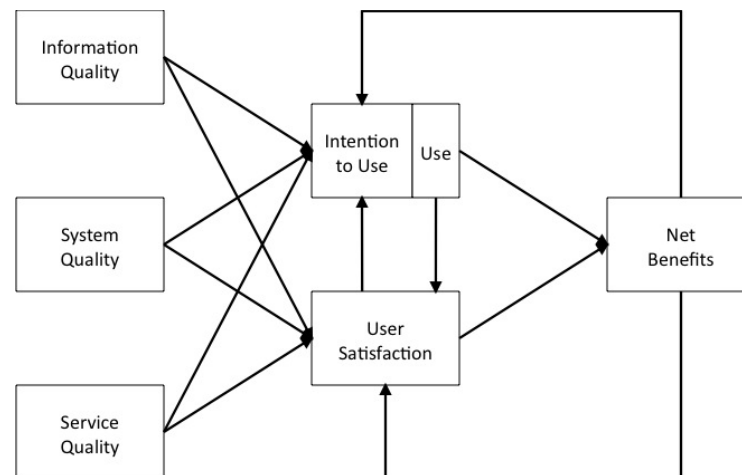
Gambar 8. ISO 9126 Model

Model ini menentukan enam karakteristik termasuk Functionality, Reliability, Usability, Efisiensi, Maintainability dan Portabilitas, yang dibagi lagi menjadi 21 sub karakteristik. Sub karakteristik diwujudkan eksternal ketika perangkat lunak digunakan sebagai bagian dari sistem komputer, dan merupakan hasil dari atribut perangkat lunak internal.



Karakteristik didefinisikan berlaku untuk setiap jenis perangkat lunak, termasuk program komputer dan data yang terdapat dalam firmware dan memberikan terminologi yang konsisten untuk kualitas produk perangkat lunak. Model ini juga menyediakan kerangka kerja untuk membuat timbal balik antara kemampuan produk perangkat lunak. [3] Gambar 8 menjelaskan tentang faktor faktor dari Model ISO 9126.

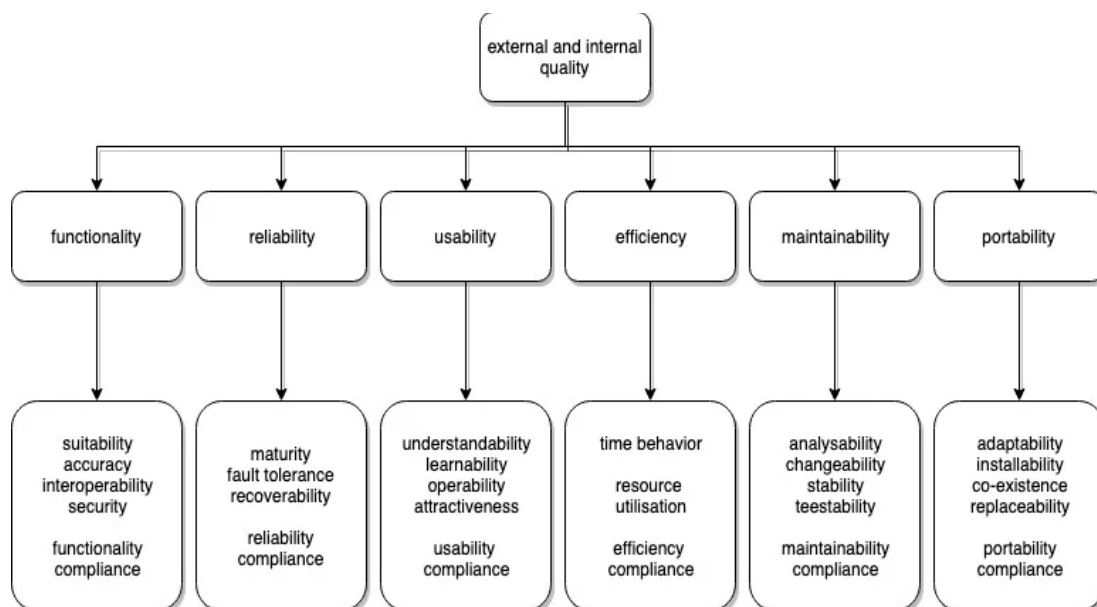
### DeLone & McLean Model (2003)



Gambar 9. DeLone & McLean Model

Gambar 9 merupakan faktor faktor yang dibahas dalam DeLone & McLean model. Kualitas Sistem dan Kualitas Informasi secara mandiri dan bersama-sama mempengaruhi baik Penggunaan dan Kepuasan Pengguna. Besarnya Penggunaan dapat mempengaruhi Kepuasan Pengguna secara positif atau negatif. Penggunaan dan Kepuasan Pengguna mempengaruhi Dampak Individual dan selanjutnya mempengaruhi Dampak Organisasional [6].

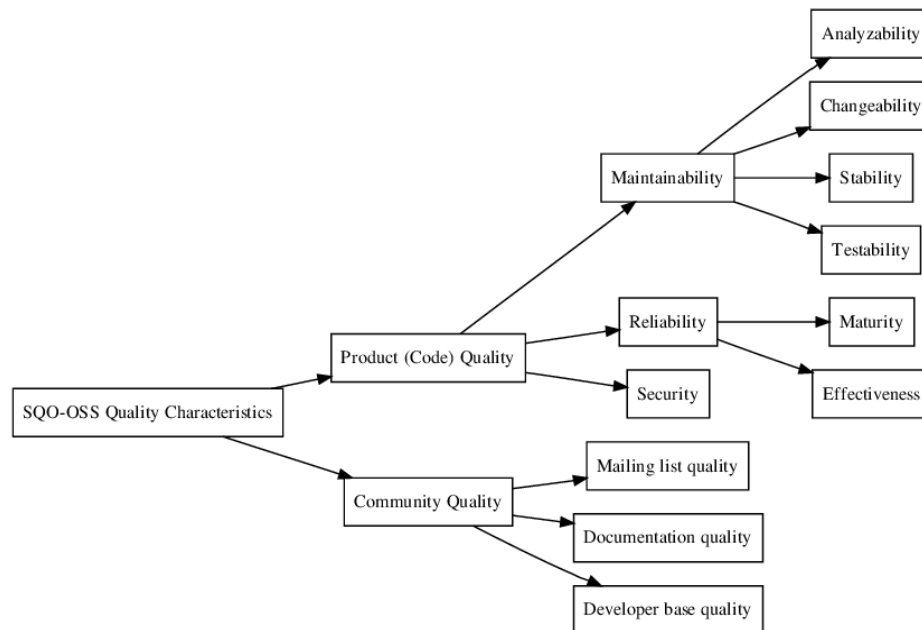
### ISO 25010 Model (2008)



Gambar 10. ISO 25010 Model

ISO versi 25010 yang merupakan versi lanjutan dari ISO/IEC 9126 dengan penambahan beberapa struktur dan bagian dari standar model kualitas. Secara keseluruhan ISO/IEC 25010 memiliki 8 karakteristik untuk mengukur kualitas perangkat lunak secara menyeluruh, antara lain portability, performance efficiency, reliability, security usability, maintainability, compatibility, dan functional suitability.[13] Gambar 10 menjabarkan tentang faktor faktor dari Model ISO 25010.

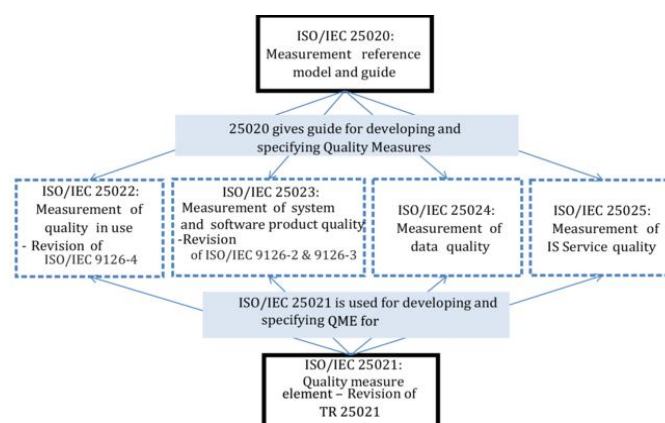
### SQO-OSS Model (2008)



Gambar 11. SQO-OSS Model

Gambar 11 adalah model hierarkis yang mengevaluasi source code dan proses komunitas yang memungkinkan perhitungan metrik secara otomatis. Model ditunjukkan pada Gambar diatas dan menurut, model berbeda dari yang lain dalam aspek-aspek berikut: Fokus ke otomasi berbeda dengan model lain yang memerlukan campur tangan pengguna yang berat. Merupakan inti dari sistem pemantauan kualitas berkelanjutan dan memungkinkan pengumpulan metrik otomatis. Itu tidak mengevaluasi fungsionalitas. Ini berfokus pada source code. **Source code adalah bagian terpenting dari proyek perangkat lunak.** Mempertimbangkan hanya faktor komunitas yang dapat diukur secara otomatis.

### ISO 25022 Model (2016)



Gambar 12. ISO 25022 Model

ISO / IEC 25022: 2016 mendefinisikan kualitas yang digunakan untuk karakteristik yang didefinisikan dalam ISO / IEC 25010 dan dimaksudkan untuk digunakan bersama dengan ISO / IEC 25010. Dapat digunakan bersama dengan ISO / IEC 2503n dan ISO / IEC 2504n standar atau untuk lebih umum memenuhi kebutuhan pengguna berkenaan dengan kualitas produk atau sistem.

ISO / IEC 25022: 2016 berisi sebagai berikut:

- Seperangkat tindakan dasar untuk setiap kualitas dalam karakteristik penggunaan;
- Penjelasan tentang bagaimana kualitas yang digunakan diukur.

Gambar 12 menjabarkan tentang faktor faktor pada Model ISO 25022.

### **Faktor faktor Software Quality Model**

Adapun faktor, atribut dan karakteristik yang umumnya yg digunakan untuk perbandingan adalah sebagai berikut :

- Accuracy : Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.[14]
- Correctness : Sejauh mana program memenuhi spesifikasi dan memenuhi tujuan tugas pengguna. Ketepatan adalah sejauh mana kerja produk dan output yang bebas dari cacat hasil kerja produk setelah dikirimkan. Kebenaran menjawab pertanyaan khas berikut: Apakah aplikasi dan data sudah lengkap, akurat dan konsisten? [3]
- Efficiency : Sejauh mana sesuatu secara efektif menggunakan (yaitu, meminimalkan konsumsi atas) sumber dayanya. Sumber daya ini dapat mencakup semua jenis sumber daya seperti komputer (perangkat keras, perangkat lunak, dan jaringan), mesin, fasilitas, dan personil. Juga, jumlah sumber daya komputasi dan kode yang diperlukan oleh program untuk melakukan fungsi, kumpulan atribut yang digunakan pada hubungan antara tingkat kinerja perangkat lunak dan jumlah sumber daya yang digunakan saat itu. Efisiensi berkaitan dengan "membagi beban, deteksi kesalahan end-to-end: tes sederhana, cacat kinerja yang muncul di bawah beban berat, mengutamakan keselamatan, penskalaan, throughput, latency, ketersediaan. Apakah model memenuhi tujuannya tanpa pemborosan sumber daya?[3]
- Functionality : Kemampuan produk perangkat lunak untuk menyediakan fungsi yang dinyatakan memenuhi dan mengandung yang dibutuhkan ketika perangkat lunak digunakan dalam kondisi tertentu. Fungsionalitas merupakan atribut yang menjaga keberadaan fungsi dan sifat spesifik mereka. Fungsi adalah sesuatu yang memenuhi atau mengandung kebutuhan. Fungsionalitas dinilai dengan mengevaluasi fitur dan kemampuan dari program, sifat umum dari fungsi yang dikirimkan dan keamanan sistem secara keseluruhan. [3]
- Interoperability : Kemampuan produk perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu. Juga, upaya yang diperlukan untuk memasang satu sistem dengan yang lain, atribut perangkat lunak yang bergantung pada kemampuannya untuk berinteraksi dengan sistem tertentu, sejauh mana sistem atau salah satu komponennya terhubung dengan benar dan beroperasi dengan sesuatu yang lain. [3]
- Portability : Kemampuan produk perangkat lunak untuk dapat dipindahkan dari satu lingkungan ke lingkungan yang lain. Juga, upaya yang diperlukan untuk memindahkan program dari satu konfigurasi perangkat keras dan atau lingkungan sistem perangkat lunak ke sistem lain. Portabilitas adalah saat dimana aplikasi atau komponen dapat dipindahkan dari satu lingkungan yang lain. [3]

- **Reliability** : Reliability adalah sebuah kemampuan produk dari perangkat lunak untuk konsisten dan stabil terhadap informasi yang dibutuhkan. Dengan kata lain adalah memiliki keakuratan sistem informasi yang dikeluarkannya atau disajikannya. Serta memiliki sifat handal, tidak terjadi kegagalan sistem atau kegagalan kinerja saat beroperasi. [3]
- **Reusability** : Reusabilitas adalah saat dimana aplikasi atau komponen yang sudah ada dapat digunakan kembali. Ini adalah sejauh mana sebuah program dapat digunakan dalam aplikasi lain yang berkaitan dengan kemas dan ruang lingkup fungsi yang program lakukan. Misalnya, reusabilitas yang mungkin ketika semua modul berisi dua atau lebih fungsi unik yang jika dipisahkan dari kode utama, dapat digunakan kembali oleh program lain. Selain itu, ia merupakan atribut yang berkaitan dengan beban transfer modul atau program untuk aplikasi lain. [3]
- **Security** : Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup (hacker) maupun otorisasi dalam modifikasi data. [14]
- **Understandability** : Kemampuan produk perangkat lunak untuk memungkinkan pengguna untuk memahami apakah perangkat lunak tersebut cocok, dan bagaimana perangkat lunak itu dapat digunakan untuk tugas dan kondisi tertentu. Juga, atribut perangkat lunak yang diperlukan dalam upaya pengguna untuk mengenali konsep logis dan penerapannya. [3] Tabel 5 dibawah ini menunjukkan perbandingan software quality models.

Tabel 5. Perbandingan Software Quality Models

Faktor / Atribut / Karakteristik	McCall	Boehm	ServQual	FURPS	WebQual	ISO 9126	DeLone & McLean	ISO 25010	SQO-OSS	ISO 25022
Accuracy								✓		✓
Correctness	✓		✓							
Efficiency	✓	✓		✓		✓	✓	✓	✓	✓
Functionality			✓	✓	✓	✓	✓	✓	✓	✓
Interoperability	✓		✓		✓	✓	✓	✓		✓
Portability	✓	✓			✓			✓		✓
Reliability	✓	✓	✓	✓		✓	✓	✓	✓	✓
Reusability	✓		✓		✓	✓	✓	✓	✓	✓
Security				✓	✓			✓	✓	✓
Understandability		✓		✓	✓	✓	✓	✓		✓

### Cara Mengukur Faktor - Faktor Kualitas Software

Dibawah ini merupakan Tabel 6 yang menerangkan faktor- faktor, cara pengukuran serta indikator dan parameter.

Tabel 6. Indikator dan Pengukuran Faktor

Faktor	Indikator dan Pengukuran
--------	--------------------------

Accuracy (Akurasi)	<p><b>Parameter</b> : Fitur yang bekerja pada sistem. Apakah sistem bekerja dengan baik tanpa error. Sistem diuji secara manual.</p> <p><b>Indikator</b> : Data ini bisa kita dapatkan melalui penghitungan rata rata dari jumlah antara fitur yang bekerja dengan baik dan fitur yang tidak bekerja dengan baik. Sistem Informasi dikatakan baik jika lebih banyak rata2 fitur yang bekerja dengan baik dibanding fitur yang tidak bekerja dengan baik. Maksud baik disini, jika di klik suatu fungsi, maka fungsi berjalan sesuai yang diinginkan. Misalnya, fitur lihat data jika data berhasil tampil maka fitur dikatakan baik.</p> <p>Contoh: Pengukuran sistem akademik UIN SUKA. Setelah di uji dari 56 fitur, 48 fitur dapat bekerja dengan baik dan 8 fitur tidak sesuai antara fitur yang digunakan. Maka diambil nilai sebanyak <math>48/56 = 85,71\%</math>. Sehingga kesimpulannya sistem informasi berhasil sesuai faktor accuracy.</p>
Correctness (Ketepatan)	<p>Cara mengukur Correctness melalui tiga parameter yaitu Completeness, Availability, dan Consistency.</p> <p>Contoh: Pengukuran sistem akademik UIN SUKA.</p> <p><b>Indikator</b> : Dimasukkan dalam rumus,  <math>Correctness = (Completeness + Consistency + Traceability) / 3</math>.          Sistem Informasi dikatakan baik atau tidak berdasarkan rating indikator</p> <p>a) <i>Completeness</i> .  <b>Parameter</b> : sejauh mana implementasi penuh dari fungsi-fungsi yang diperlukan telah tercapai.          Dalam pengujian kelengkapan untuk operator, komentar berarti bahwa fitur pada sistem berfungsi dengan baik. Hasil yang dinyatakan atau sesuai dari input dan output adalah 45 fitur dari 46 fitur. Jadi persentase kelengkapan operator berdasarkan formula 3.6 adalah <math>(45 \div 46) \times 100\% = 97,8\%</math>. Dalam skala penilaian, kelengkapan operator termasuk dalam kategori sangat baik.</p> <p>b) <i>Consistency</i>  <b>Parameter</b> : adalah derajat penggunaan teknik-teknik desain dan dokumentasi yang seragam pada seluruh proyek pengembangan software.          Dalam pengujian konsistensi dari total metrik fitur yang diuji adalah 49 fitur dan semua komentar. Maka persentase metrik operator konsistensi adalah <math>(49 \div 49) \times 100\% = 100\%</math>. Jadi itu termasuk dalam kategori sangat baik</p> <p>c) <i>Traceability</i>  <b>Parameter</b> : Berfokus terhadap kemudahan untuk melakukan penelusuran suatu dokumentasi yang dimiliki oleh perangkat lunak tersebut.          Dari pengujian pada fitur dapat dihitung sebanyak 50. Hasil yang berkomentar atau cocok dengan sistem dengan dokumen sebanyak 49 fitur. Sehingga nilai persentase keterlacakan operator adalah <math>(49 \div 50) \times 100\% = 98\%</math>. Kemudian nilai admin keterlacakan termasuk dalam kategori sangat baik.          Berdasarkan pengujian, diperoleh hasil untuk setiap metrik yang ada pada faktor ketepatan sistem untuk operator, yaitu <i>Completeness</i> 97,8%, <i>Consistency</i> 100% dan <i>Traceability</i> 98%, Sehingga <math>(97,8\% + 100\% + 98\%) \div 3 = 98,6\%</math>. Dengan persentase ini, kebenaran operator termasuk dalam kategori sangat baik.</p>

Efficiency (Efisiensi)	<p><b>Parameter</b> : Jumlah fungsi yang berjalan sesuai apa yang diharapkan.</p> <p><b>Indikator</b> : Untuk mengetahui fungsi pada program.Maka yang terlibat adalah programmer. Developer membuat dokumentasi sistem mengenai fungsi yang ada pada sistem.Pengukuran secara manual,jika fungsi dijalankan apakah fungsi dapat berjalan sesuai dokumentasi.Jadi misal ada fungsi untuk sorting data sesuai abjad,tapi di sistem sorting data berdasarkan angka.Sehingga ini termasuk fungsi yang tidak berjalan dengan baik.</p> <p>Cara mengukur: <math>X = 1 - (A/B)</math></p> <p>A = Jumlah fungsi yang tidak berjalan atau error.</p> <p>B = Total jumlah fungsi pada sistem/program</p> <p><b>Untuk hasil , <math>0 \leq X \leq 1</math> Semakin dekat ke 1.0 semakin baik</b></p> <p>Contoh pengukuran pada Sistem Informasi Akademik SMK Negeri 2 Depok Sleman. Hasil dari pengukuran adalah <math>X = 1 - A/B = 1 - (0/ 166) = 1</math>. Dari skor tingkat functionality tersebut maka kualitas perangkat lunak dari sisi functionality dapat dikatakan ‘Baik’ sesuai interpretasi dari ISO 9126 yaitu nilai yang baik adalah nilai yang mendekati 1.</p>								
Functionality (Kegunaan)	<p><b>Parameter</b> : Jumlah fungsi yang berjalan sesuai apa yang diharapkan.</p> <p><b>Indikator</b> : Untuk mengetahui fungsi pada program.Maka yang terlibat adalah programmer.Developer membuat dokumentasi sistem mengenai fungsi yang ada pada sistem.Pengukuran secara manual,jika fungsi dijalankan apakah fungsi dapat berjalan sesuai dokumentasi.Jadi misal ada fungsi untuk sorting data sesuai abjad,tapi di sistem sorting data berdasarkan angka.Sehingga ini termasuk fungsi yang tidak berjalan dengan baik.</p> <p>Cara mengukur: <math>X = 1 - (A/B)</math></p> <p>A = Jumlah fungsi yang tidak berjalan atau error.</p> <p>B = Total jumlah fungsi pada sistem/program</p> <p><b>Untuk hasil , <math>0 \leq X \leq 1</math> Semakin dekat ke 1.0 semakin baik</b></p> <p>Contoh pengukuran pada Sistem Informasi Akademik SMK Negeri 2 Depok Sleman. Hasil dari pengukuran adalah <math>X = 1 - A/B = 1 - (0/ 166) = 1</math>. Dari skor tingkat functionality tersebut maka kualitas perangkat lunak dari sisi functionality dapat dikatakan ‘Baik’ sesuai interpretasi dari ISO 9126 yaitu nilai yang baik adalah nilai yang mendekati 1.</p>								
Interoperability (Interoperabilitas)	<p><b>Parameter</b> : Kemudahan pengguna dalam menggunakan sistem.</p> <p><b>Indikator</b> : Pengujian faktor interoperability untuk mengukur kemudahan pengguna mengoperasikan sistem informasi akademik. Alat ukur yang digunakan adalah kuesioner yang diisi oleh pengguna. Pengguna sistem ini adalah dosen. Pengukuran menggunakan skala likert dengan 5 skala dan 10 pernyataan. Kelima skala diberi bobot dari 5 sampai dengan 1 secara berurutan dengan dinyatakan dalam “sangat setuju (SS)”, “setuju (S)”, “netral (SS)”, “tidak setuju (TS)” dan “sangat tidak setuju (STS)”.</p> <p>Skor Maksimum = jumlah responden x skor tertinggi likert .</p> <p>Indeks (%) = (Total Skor / Skor Maksimum) x 100 .</p> <p><b>Interval Penilaian</b></p> <p>Indeks 0% – 19,99% : Sangat Buruk</p> <p>Indeks 20% – 39,99% : Buruk</p> <p>Indeks 40% – 59,99% : Cukup</p> <p>Indeks 60% – 79,99% : Baik</p> <p>Indeks 80% – 100% : Sangat Baik</p> <p>Contoh pada Sistem Informasi Akademik UIN Sunan Kalijaga :</p> <p>Tabel 7. Contoh pengukuran interoperability</p> <table><tr><td>No.</td><td>Pernyataan</td><td>SS</td><td>S</td><td>N</td><td>TS</td><td>STS</td><td>JML</td></tr></table>	No.	Pernyataan	SS	S	N	TS	STS	JML
No.	Pernyataan	SS	S	N	TS	STS	JML		

1	Aplikasi akademik UIN Sunan Kalijaga mudah diakses dari semua device.	18	23	7	1	1	50
2	Menu-menu yang ada di aplikasi akademik UIN Sunan Kalijaga mudah dipahami.	4	19	18	9	0	50
3	Menu-menu yang ada di aplikasi akademik UIN Sunan Kalijaga mudah dipahami.	16	29	3	2	0	50
4	Menu isi presensi MK dosen di akademik UIN Sunan Kalijaga mudah digunakan.	10	19	13	8	0	50
5	Proses cetak laporan rekapitulasi presensi perkuliahan di menu laporan mata kuliah dosen mudah dilakukan.	10	19	14	7	0	50
	Jumlah	58	138	55	27	1	250

Tabel 7 diatas berupa hasil kuesioner untuk pengamatan interoperability, dan diperoleh perhitungan interoperability yaitu indeks 50 %.Maka sistem informasi dikatakan cukup. Sehingga perlu ditingkatkan lagi minimal baik.

Portability  
(Kemampuan beradaptasi)

**Parameter :** Memindahkan komponen dari satu sistem ke sistem yang lain.

**Indikator :** Cara pengujian kualitas portability ini menggunakan sebuah perangkat lunak BrowseEmAll. Proses pengujian dilakukan pada berbagai varian browser yang ada pada perangkat lunak BrowseEmAll tersebut. Ada 8 browser yang digunakan untuk menguji aspek portability.

**Interval Penilaian**

Indeks 0% – 19,99% : Sangat Buruk

Indeks 20% – 39,99% : Buruk

Indeks 40% – 59,99% : Cukup

Indeks 60% – 79,99% : Baik

Indeks 80% – 100% : Sangat Baik

Contoh : Sistem Informasi UNS.

Portability pada sistem informasi dapat disimpulkan berjalan secara baik pada seluruh website dengan persentase:

$$8/8 * 100\% = 100\%$$

Kesimpulan : Dari hasil data diatas bahwa sistem informasi Universitas Negeri Surakarta (UNS) memiliki portability yang baik.

**Parameter :** Keakuratan serta kestabilan sistem dalam pengujian.

**Indikator :** Pengujian diuji menggunakan aplikasi Web Application Load, Stress and Performance Testing (WAPT). Pengujian dilakukan dengan stress testing, yaitu digunakan untuk mengukur metrik sessions, pages dan hits dengan memberikan input 1-20 user simultan selama 10 menit dengan desain setiap 60 detik terdapat tambahan 10 user yang mengakses sistem informasi tersebut.

Contoh : Reliability dari hasil test kedua website universitas hasilnya :



Reliability (Reliabilitas)	<ul style="list-style-type: none"><li>•UNS : bahwa persentase keberhasilan sessions, pages dan hits sebesar 100%. Kegagalan tidak terjadi pada masing-masing metrik.</li><li>•UNY : bahwa persentase keberhasilan sessions, pages dan hits sebesar 99,7%. Kegagalan sebesar 0,3% terjadi pada masing-masing metrik.</li></ul>																																				
Reusability (Kegunaan)	<p><b>Parameter</b> : Seberapa baik perangkat lunak dapat dipahami, dipelajari, dan digunakan.</p> <p><b>Indikator</b> : Sebuah sistem mempunyai fungsi modul help pada sistem. Rumus Reusability = Total fungsi yang dijelaskan/Total fungsi yang ada ×100%. Catatan : total fungsi yang dijelaskan maksudnya total fungsi yang ada dan dijelaskan pada menu help.</p> <p>Contoh : Penerapan pada ejournal UIN Malang..Diperoleh seluruh fungsi pada help dapat diterangkan semua sehingga diperoleh ada 15 fungsi pada ejournal UIN Malang.Sehingga dapat diperoleh 15/15*100% = 100%.Sehingga faktor reusability bernilai baik pada ejournal UIN Malang.</p>																																				
Security (Keamanan)	<p><b>Parameter</b> : Tingkat kemampuan pengawasan akses terhadap data atau software oleh orang-orang tertentu.</p> <p><b>Indikator</b> :</p> <ul style="list-style-type: none"><li>•Audibilitas : Penyesuaian / keselarasan terhadap standar yang dapat diperiksa yaitu dengan memeriksa desain sistem.</li><li>•Instrumentasi: Tingkatan dimana pengawasan program memiliki operasi tersendiri dan mengidentifikasi kesalahan yang terjadi.</li><li>•Keamanan: Ketersediaan mekanisme yang mengontrol atau melindungi program atau data.</li></ul> <p>Rumus Security= Audibilitas + Instrumentasi + Keamanan/ 3.</p> <p><b>Interval Penilaian</b></p> <p>Indeks 0% – 19,99% : Sangat Buruk</p> <p>Indeks 20% – 39,99% : Buruk</p> <p>Indeks 40% – 59,99% : Cukup</p> <p>Indeks 60% – 79,99% : Baik</p> <p>Indeks 80% – 100% : Sangat Baik</p> <p>Contoh pengukuran pada Sistem Informasi UIN Suka :</p> <p>Tabel 8. Contoh pengukuran Security</p> <table><tr><th>No.</th><th>Activity</th><th>Product</th><th>Doc.System</th><th>Remark</th><th>Keterangan</th></tr><tr><td>1</td><td>Login</td><td>✓</td><td>✓</td><td>✓</td><td>Sesuai</td></tr><tr><td>2</td><td>Session Logout</td><td>✓</td><td>✓</td><td>✓</td><td>Sesuai</td></tr><tr><td>3</td><td>Level User</td><td>✓</td><td>✓</td><td>✓</td><td>Sesuai</td></tr><tr><td>4</td><td>Enkripsi Password</td><td>✓</td><td>X</td><td>X</td><td>Tidak Sesuai</td></tr><tr><td>5</td><td>Enkripsi url</td><td>✓</td><td>X</td><td>X</td><td>Tidak Sesuai</td></tr></table>	No.	Activity	Product	Doc.System	Remark	Keterangan	1	Login	✓	✓	✓	Sesuai	2	Session Logout	✓	✓	✓	Sesuai	3	Level User	✓	✓	✓	Sesuai	4	Enkripsi Password	✓	X	X	Tidak Sesuai	5	Enkripsi url	✓	X	X	Tidak Sesuai
No.	Activity	Product	Doc.System	Remark	Keterangan																																
1	Login	✓	✓	✓	Sesuai																																
2	Session Logout	✓	✓	✓	Sesuai																																
3	Level User	✓	✓	✓	Sesuai																																
4	Enkripsi Password	✓	X	X	Tidak Sesuai																																
5	Enkripsi url	✓	X	X	Tidak Sesuai																																



	<p>Tabel 8 diatas merupakan hasil pengukuran dan fungsi yang diukur dari faktor security. Sehingga diperoleh :</p> <ul style="list-style-type: none"><li>• <b>Auditability</b> : Fitur SIA yang diuji sebanyak 47 fitur, terdapat 39 fitur yang sesuai dengan standar yang ditentukan. Maka hasil nya adalah 82.97 %</li><li>• <b>Instrumentation</b> : Fitur SIA yang diuji sebanyak 48 fitur, terdapat 44 fitur yang memiliki identifikasi kesalahan operasi dari program. Maka hasilnya adalah 91.66%</li><li>• <b>Keamanan</b> : Total aktivitas yang diuji sebanyak 5 fitur, terdapat 3 fitur yang memiliki security. Maka diperoleh hasilnya adalah <math>3/5 \times 100\% = 60\%</math></li></ul> <p>Maka kesimpulan dari Security, dari aspek Auditability dan -Instrumentation sudah baik karena di atas 60%, Sedangkan Keamanan harus ditingkatkan karena hampir sama dengan 60%</p>																																								
Understand- ability (Pemahaman Pengguna)	<p><b>Parameter</b> : Seberapa jauh sebuah perangkat lunak dapat dipahami oleh pengguna <b>Indikator</b> : Pemahaman dari pengguna terhadap penggunaan sistem informasi sesuai kuesioner. <b>Interval Penilaian</b> Indeks 0% – 19,99% : Sangat Buruk Indeks 20% – 39,99% : Buruk Indeks 40% – 59,99% : Cukup Indeks 60% – 79,99% : Baik Indeks 80% – 100% : Sangat Baik</p> <p>Contoh pengukuran understandability :</p> <p>Tabel 9. Contoh pengukuran Understandability</p> <table><tr><th rowspan="2">No</th><th rowspan="2">Pernyataan</th><th colspan="4">Pilihan</th></tr><tr><th>TS</th><th>KS</th><th>S</th><th>SS</th></tr><tr><td>1</td><td>Secara keseluruhan, saya merasa puas dengan kemudahan sistem ini.</td><td>0</td><td>0</td><td>1</td><td>4</td></tr><tr><td>2</td><td>Cara penggunaan sistem ini sangat simpel.</td><td>0</td><td>0</td><td>1</td><td>4</td></tr><tr><td>3</td><td>Sistem ini sangat mudah dipelajari.</td><td>0</td><td>1</td><td>1</td><td>3</td></tr><tr><td>4</td><td>Informasi yang disediakan sistem ini sangat jelas.</td><td>0</td><td>0</td><td>0</td><td>5</td></tr><tr><td>5</td><td>Informasi yang diberikan oleh sistem ini sangat mudah dipahami.</td><td>0</td><td>1</td><td>1</td><td>3</td></tr></table> <p>Dari tabel 9 merupakan hasil pengukuran dapat diketahui persentase untuk masing-masing penilaian adalah :</p> <p>Sangat Setuju (SS) = <math>(68/95) \times 100\% = 71,58\%</math> Setuju (S) = <math>(5/7) \times 100\% = 21,05\%</math> Kurang Setuju (KS) = <math>(2/7) \times 100\% = 7,36\%</math> Tidak Setuju (TS) = <math>(0/12) \times 100\% = 0\%</math></p> <p>Dari hasil persentase yang didapatkan, maka didapat persentase kualitas perangkat dari sisi pemahaman pengguna (<i>understandability</i>) adalah 71,58% pengguna sangat setuju, 21,05% pengguna setuju, dan 7,36% pengguna kurang setuju.</p>	No	Pernyataan	Pilihan				TS	KS	S	SS	1	Secara keseluruhan, saya merasa puas dengan kemudahan sistem ini.	0	0	1	4	2	Cara penggunaan sistem ini sangat simpel.	0	0	1	4	3	Sistem ini sangat mudah dipelajari.	0	1	1	3	4	Informasi yang disediakan sistem ini sangat jelas.	0	0	0	5	5	Informasi yang diberikan oleh sistem ini sangat mudah dipahami.	0	1	1	3
No	Pernyataan			Pilihan																																					
		TS	KS	S	SS																																				
1	Secara keseluruhan, saya merasa puas dengan kemudahan sistem ini.	0	0	1	4																																				
2	Cara penggunaan sistem ini sangat simpel.	0	0	1	4																																				
3	Sistem ini sangat mudah dipelajari.	0	1	1	3																																				
4	Informasi yang disediakan sistem ini sangat jelas.	0	0	0	5																																				
5	Informasi yang diberikan oleh sistem ini sangat mudah dipahami.	0	1	1	3																																				

## KESIMPULAN

Berdasarkan survei yang telah dilakukan maka dapat disimpulkan bahwa pengukuran kualitas perangkat lunak dapat dilakukan dengan menggunakan salah satu dari berbagai model yang ada. Pada penggunaan model Boehm dan ServQual, kekurangan yang ada sama yaitu kurangnya kriteria yang digunakan. Akan tetapi model Boehm dapat mempertimbangkan kualitas dari sudut pandang yang berbeda, dan model ServQual menitik beratkan pada faktor kualitas. Model ISO merupakan standar internasional yang ada saat ini, hanya saja model ini bersifat umum. Akan tetapi model ini memiliki kriteria evaluasi dan memisahkan kualitas eksternal dan internal yang ada sehingga model ini cocok digunakan pada saat kebutuhan penilaian kualitas perangkat lunak tidak terlalu dalam. Model McCall dan ServQual memenuhi banyak komponen penilaian, akan tetapi ada penilaian yang tumpang tindih karena banyaknya komponen yang dinilai. Model ini sesuai digunakan jika permasalahan utama adalah penilaian secara menyeluruh dan mendalam. Model ISO 9126 hampir sama dengan model ISO 25000, hanya saja pada model ISO 25000 menggunakan metrik perangkat lunak pada tingkat penilaian terakhir. Akibatnya waktu penilaian akan lebih lama dan juga kedalaman penilaian akan lebih mendetail. Salah satu keunggulan model Boehm adalah dilibatkannya faktor perangkat keras. Hanya saja pada model ini kriteria yang digunakan masih sedikit. Model FURPS sendiri tidak memperdulikan portability, akan tetapi model ini memisahkan kebutuhan fungsional dan non fungsional. Pada model WebQual tingkat terakhir penilaian adalah menggunakan beberapa skenario untuk pengujian karakteristik. Model yang menggunakan pengukuran secara subjektif antara lain WebQual, McCall, FURPS, Boehm, ServQual, DeLone & McLean sedangkan secara objektif antara lain ISO 9126, SQO-OSS, ISO 25010, ISO 25022. Penggunaan model tersebut disesuaikan dengan kebutuhan yang diperlukan seperti waktu, kedalaman pengukuran kualitas, kompleksitas dan juga fungsi dari perangkat lunak tersebut. Kualitas perangkat lunak diukur menggunakan beberapa faktor, atribut dan juga karakteristik.

## REFERENSI

- [1] R. Baxter, *Software engineering is software engineering*. 2006.
- [2] J. P. Miguel and D. Mauricio, "Sebuah Tinjauan Kualitas Software Model untuk Evaluasi Software Produk," vol. 5, no. 6, pp. 31–53, 2014.
- [3] L. A. A. R. Wayan Gede Suka Parwita and Putri, "Komponen Penilaian Kualitas Perangkat Lunak berdasarkan Software Quality Models," *Semin. Nas. Teknol. Inf. Komun. Terap. 2012 (Semantik 2012)*, vol. 2012, no. Semantik, pp. 89–94, 2012.
- [4] G. Polančič and B. Cegnar, "Complexity metrics for process models – A systematic literature review," *Comput. Stand. Interfaces*, vol. 51, pp. 104–117, 2017, doi: 10.1016/j.csi.2016.12.003.
- [5] G. Taguchi, P. B. Crosby, W. A. Shewhart, M. Joseph, and R. M. Pirsig, "is Quality?"
- [6] T. Wahyudi, F. A. Bachtiar, and M. C. Saputra, "Analisis Pengujian Usability untuk Membandingkan Kemudahan Transaksi pada BukaLapak dengan Lazada," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 5, pp. 2548–964, 2018.
- [7] S. Kasus, P. Badan, and P. Teknologi, "Kualitas Aplikasi Digilib Sebagai Sistem Otomasi Perpustakaan : Universitas Islam Negeri Syarif Hidayatullah Jakarta 1439 H / 2018 M," 2018.
- [8] R. E. Al-Qutaish, "Quality Models in Software Engineering Literature: An Analytical and Comparative Study," *J. Am. Sci.*, vol. 6, no. 3, pp. 166–175, 2010.
- [9] A. Hidayati, S. Kom, A. Retno, and T. Hayati, "Analisa Pengembangan Model Kualitas Berstruktur Hirarki Dengan Kustomisasi Iso 9126 Untuk Evaluasi Aplikasi Perangkat Lunak B2B," *Semin. Nas. Electr. Inform. It's Educ.*, 2009.
- [10] D. P. Kesuma, "Analisis Pengukuran Kualitas Layanan Web Perguruan Tinggi Xyz Menggunakan Servqual," *Semin. Nas. Inform. 2014*, pp. 178–183, 2014.
- [11] B. Behkamal, M. Kahani, and M. K. Akbari, "Customizing ISO 9126 quality model for evaluation of B2B applications," *Inf. Softw. Technol.*, vol. 51, no. 3, pp. 599–609, 2009, doi: 10.1016/j.infsof.2008.08.001.
- [12] . Syaifullah and D. oksa Soemantri, "PENGUKURAN KUALITAS WEBSITE MENGGUNAKAN METODE WEBQUAL 4.0 (Studi Kasus: CV. Zamrud Multimedia Network)," *J. Ilm. Rekayasa dan*

- 
- Manaj. Sist. Inf.*, vol. 2, no. 1, pp. 19–25, 2016.
- [13] A. A. Fitrawan and C. Sri, “Pengukuran Kualitas Perangkat Lunak berdasarkan ISO / IEC 25000 : Systematic Mapping,” *J. Manaj. Inform.*, vol. 04, no. 1, pp. 36–45, 2015.
- [14] S. H. Anwariningsih, “Multi Faktor Kualitas Website,” *J. Gaung Inform.* 4, vol. 1, no. 0271, pp. 17–29, 2011.
-