

BAB II

LANDASAN TEORI

2.1 Dasar Teori

Dalam penelitian ini beberapa teori penunjang yang sesuai dengan topik penelitian untuk dijadikan sebagai dasar pengetahuan yaitu sebagai berikut:

2.1.1 Pengukuran

Menurut Arikunto dalam Wulan (2017) bahwa pengukuran adalah suatu kegiatan yang membandingkan sesuatu dengan satuan ukuran yang ditentukan sehingga bersifat kuantitatif. Pengukuran merupakan pengumpulan data berupa informasi yang relevan dengan tujuan yang telah ditentukan Cangelosi dalam Wulan (2017). Menurut Zainul dan Nasution dalam Wulan (2017) pengukuran memiliki karakteristik yaitu menggunakan angka atau skala dan mengikuti aturan yang telah ditentukan.

Pengukuran merupakan suatu hal yang sangat mendasar bagi setiap disiplin ilmu seperti rekayasa teknik perangkat lunak (*software engineering*). Pengukuran adalah suatu elemen penting dalam proses *engineering*, melalui pengukuran produk atau sistem yang telah dibangun dapat dihitung atributnya secara kuantitatif sehingga mendapatkan pemahaman yang lebih dalam terhadap sebuah sistem. Dengan melakukan pengukuran, dapat memperoleh informasi terkait suatu objek atau produk melalui mekanisme evaluasi yang secara objektif. Hasil dari suatu pengukuran yang dilakukan dapat diolah sehingga dapat mengetahui tingkat pencapaian suatu perangkat lunak (Susanto, dkk., 2015).

2.1.2 Computer Based Test

Menurut Putri & Rahayu (2018) *Computer Based Test* merupakan pengujian dan penilaian belajar dari siswa yang dilakukan secara otomatis menggunakan alat komputer yang telah terhubung dengan jaringan internet. Menurut McKenna dalam Putri & Rahayu (2018) *Computer Based Test* digunakan pada komputer

untuk melakukan tes dan melihat penilaian hasil belajar dari siswa. Pengujian dan penilaian tersebut memiliki standar yang terklasifikasi sebagai berikut:

1. Penilaian diagnostik yaitu tes yang dilakukan untuk melihat pemahaman awal dari siswa terhadap suatu subjek.
2. Tes individu yaitu pengujian yang dilakukan dengan cara memberikan sebuah umpan balik kepada siswa untuk mengetahui perkembangan belajarnya mengenai keterampilan dan pemahaman tentang suatu subjek. Pada tes jenis ini bentuknya berupa pertanyaan objektif (*Objective Test*).
3. Tes sumatif yaitu pengujian yang dilakukan untuk mengukur dan mengevaluasi keberhasilan belajar pada suatu area atau tempat belajar dari siswa tersebut. Tes pada jenis ini juga dalam bentuk tes objektif.

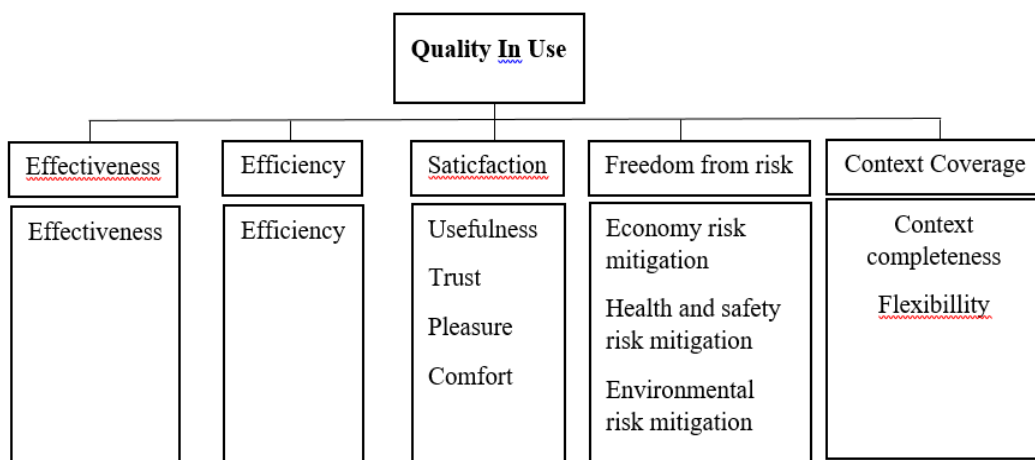
2.1.3 ISO/IEC 25010

Menurut Hengki, dkk., (2018) ISO/IEC 25010 merupakan salah satu dari model pengujian dan evaluasi kualitas perangkat lunak yang termasuk kedalam bagian dari *Software Product Quality Requirements and Evaluation (SQuaRE)*, yang memiliki langkah atau cara pengujian yang berkaitan dengan model kualitas perangkat lunak model ISO 9126. Menurut Joyce, dkk., (2015) pada ISO/IEC 25010 memiliki delapan domain (karakteristik) dan memiliki sub domain tambahan. Beberapa dari sub domainnya dipindahkan ke domain lainnya. ISO/IEC 25010 telah diatur ulang dan dibuat karakteristik dan sub karakteristik baru serta dilakukan peningkatan pemahaman definisi untuk mengatasi keterbatasan pada ISO 9126, karena dianggap kurang lengkap dan kurang jelas. Sehingga didalam ISO/IEC 25010 memiliki karakteristik yang lebih lengkap dan terbaru dibandingkan dengan ISO sebelumnya yaitu ISO 9126.

Dalam ISO/IEC 25010 memiliki 2 model kualitas yaitu *Quality in use models* dan *Product quality models*. Karakteristik yang didefinisikan oleh kedua model ini relevan untuk semua produk perangkat lunak. Karakteristik dan sub karakteristik memberikan istilah yang konsisten untuk menentukan, mengukur dan mengevaluasi kualitas produk sistem dan perangkat lunak.

a. *Quality In Use*

Quality In Use terdiri dari lima karakteristik beberapa di antaranya dibagi lagi menjadi sub karakteristik, yang berhubungan dengan hasil interaksi ketika produk digunakan. *Quality In Use* mendefinisikan lima karakteristik terkait hasil interaksi dengan sistem seperti *effectiveness*, *efficiency*, *satisfaction*, *freedom from risk*, and *context coverage* yang dapat dilihat pada gambar 2.1.



Gambar 2. 1 *Quality In Use Models* (ISO/IEC, 2011)
Sumber: ISO/IEC (2011)

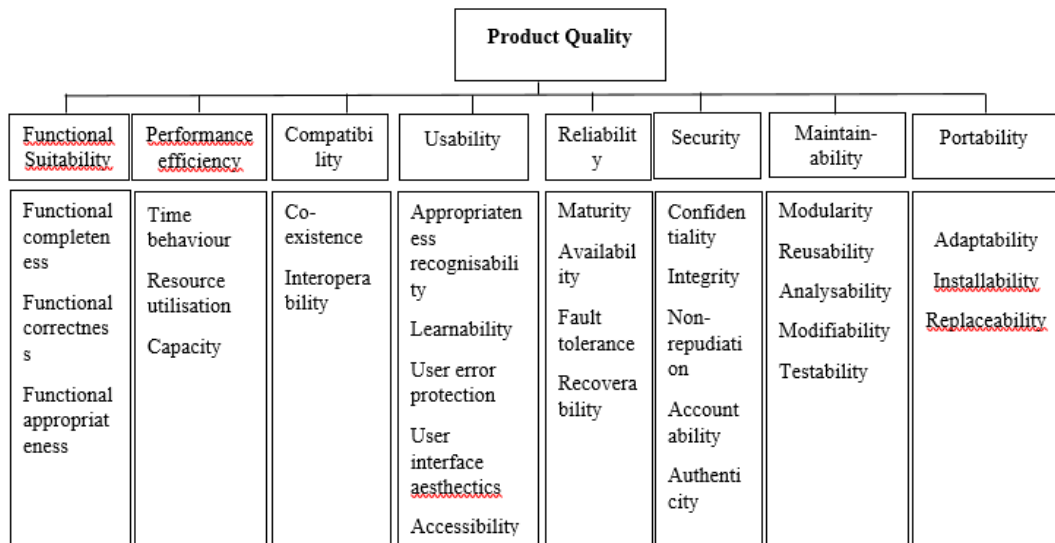
Tabel 2. 1 Karakteristik dan Sub Karakteristik *Quality In Use*

| Karakteristik & sub karakteristik | Definisi |
|-----------------------------------|---|
| 1. <i>Effectiveness</i> | akurasi dan kelengkapan yang digunakan pengguna untuk mencapai tujuan tertentu |
| 2. <i>Efficiency</i> | sumber daya yang dikeluarkan terkait dengan keakuratan dan kelengkapan yang digunakan pengguna untuk mencapai tujuan |
| 3. <i>Satisfaction</i> | sejauh mana kebutuhan pengguna terpenuhi ketika produk atau sistem digunakan dalam konteks penggunaan tertentu |
| 3.1. <i>Usefulness</i> | tingkat kepuasan pengguna dengan pencapaian tujuan pragmatis yang mereka rasakan, termasuk hasilnya penggunaan dan konsekuensi penggunaan |

| | |
|---|--|
| 3.2. <i>Trust</i> | sejauh mana pengguna atau pemangku kepentingan lainnya memiliki keyakinan bahwa produk atau sistem akan berperilaku seperti yang diinginkan |
| 3.3. <i>Pleasure</i> | sejauh mana pengguna memperoleh kesenangan dari memenuhi kebutuhan pribadinya |
| 3.4. <i>Comfort</i> | sejauh mana pengguna puas dengan kenyamanan fisik perangkat lunak |
| 4. <i>Freedoom from risk</i> | sejauh mana produk atau sistem memitigasi potensi risiko terhadap status ekonomi, kehidupan manusia, kesehatan, atau lingkungan Hidup |
| 4.1. <i>Economic risk mitigation</i> | sejauh mana produk atau sistem memitigasi potensi risiko terhadap status keuangan, operasi yang efisien, properti komersial, reputasi, atau sumber daya lain dalam konteks penggunaan yang dimaksudkan |
| 4.2. <i>Health and safety risk mitigation</i> | sejauh mana produk atau sistem mengurangi potensi risiko bagi orang-orang dalam konteks penggunaan yang dimaksudkan |
| 4.3. <i>Environmental risk mitigation</i> | sejauh mana produk atau sistem memitigasi potensi risiko terhadap properti atau lingkungan yang dimaksudkan konteks penggunaan |
| 5. <i>Context Coverage</i> | sejauh mana produk atau sistem dapat digunakan dengan efektifitas, efisiensi, bebas dari risiko dan kepuasan dalam konteks penggunaan yang ditentukan dan dalam konteks di luar yang awalnya secara eksplisit diidentifikasi |
| 5.1. <i>Context Completness</i> | sejauh mana produk atau sistem dapat digunakan dengan efektifitas, efisiensi, bebas dari risiko dan kepuasan dalam semua konteks penggunaan yang ditentukan |
| 5.2. <i>Flexibility</i> | sejauh mana produk atau sistem dapat digunakan dengan efektifitas, efisiensi, bebas dari risiko dan kepuasan dalam konteks di luar yang awalnya ditentukan dalam persyaratan |

b. *Product Quality*

Product Quality terdiri dari delapan karakteristik yang selanjutnya dibagi lagi menjadi subkarakteristik. Penjelasan didalam karakteristik *product quality* dapat dilihat pada gambar 2.2.



Gambar 2. 2 Gambar *Product Quality models* (ISO/IEC, 2011)
Sumber: ISO/IEC (2011)

Tabel 2. 2 Karakteristik dan Sub Karakteristik *Product Quality*

| Karakteristik & Sub Karakteristik | Definisi |
|--|---|
| 1. <i>Functional Suitability</i> | sejauh mana produk atau sistem menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan di bawah kondisi tertentu |
| 1.1. <i>Functional Completeness</i> | sejauh mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan |
| 1.2. <i>Functional Correctness</i> | sejauh mana suatu produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang dibutuhkan |
| 1.3. <i>Functional Appropriateness</i> | sejauh mana fungsi memfasilitasi pencapaian tugas dan tujuan tertentu |
| 2. <i>Performance Efficiency</i> | kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi yang dinyatakan |
| 2.1. <i>Time behavior</i> | sejauh mana respon dan waktu pemrosesan dan tingkat throughput suatu produk atau sistem, kapan menjalankan fungsinya, memenuhi persyaratan |
| 2.2. <i>Resource utilization</i> | sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem, saat menjalankannya fungsi, memenuhi persyaratan |
| 2.3. <i>Capacity</i> | sejauh mana batas maksimum suatu produk atau parameter sistem memenuhi persyaratan |
| 3. <i>Compatibility</i> | sejauh mana suatu produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem |

| | |
|---|---|
| | atau komponen, menjalankan fungsi yang diperlukan, sambil berbagi lingkungan perangkat keras atau perangkat lunak yang sama |
| 3.1. <i>Co-existence</i> | sejauh mana suatu produk dapat menjalankan fungsi yang diperlukan secara efisien sambil berbagi lingkungan yang sama |
| 3.2. <i>Interoperability</i> | sejauh mana dua atau lebih sistem, produk atau komponen dapat bertukar informasi dan menggunakan informasi yang telah dipertukarkan |
| 4. <i>Usability</i> | sejauh mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan yang ditentukan efektivitas, efisiensi dan kepuasan dalam konteks penggunaan tertentu |
| 4.1. <i>Appropriateness recognizability</i> | sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka |
| 4.2. <i>Learnability</i> | sejauh mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan pembelajaran yang ditentukan menggunakan produk atau sistem dengan efektifitas, efisiensi, bebas dari resiko dan kepuasan dalam hal tertentu konteks penggunaan |
| 4.3. <i>Operability</i> | sejauh mana produk atau sistem memiliki atribut yang membuatnya mudah dioperasikan dan dikendalikan |
| 4.4. <i>User error protection</i> | sejauh mana sistem melindungi pengguna dari membuat kesalahan |
| 4.5. <i>User interface aesthetics</i> | sejauh mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna |
| 4.6. <i>Accessibility</i> | sejauh mana produk atau sistem dapat digunakan oleh orang-orang dengan karakteristik dan karakteristik terluas kemampuan untuk mencapai tujuan tertentu dalam konteks penggunaan tertentu |
| 5. <i>Reliability</i> | sejauh mana sistem, produk atau komponen melakukan fungsi tertentu di bawah kondisi tertentu untuk jangka waktu tertentu |
| 5.1. <i>Maturity</i> | sejauh mana sistem, produk, atau komponen memenuhi kebutuhan keandalan dalam operasi normal |
| 5.2. <i>Availability</i> | sejauh mana suatu sistem, produk, atau komponen beroperasi dan dapat diakses bila diperlukan untuk digunakan |
| 5.3. <i>Fault tolerance</i> | sejauh mana sistem, produk atau komponen beroperasi sebagaimana dimaksud meskipun terdapat perangkat keras atau kesalahan perangkat lunak |
| 5.4. <i>Recoverability</i> | sejauh mana, jika terjadi gangguan atau kegagalan, produk atau sistem dapat memulihkan |

| | |
|-----------------------------|---|
| | data secara langsung terpengaruh dan membangun kembali keadaan sistem yang diinginkan |
| 6. <i>Security</i> | sejauh mana suatu produk atau sistem melindungi informasi dan data sehingga orang atau produk lain atau sistem memiliki tingkat akses data yang sesuai dengan jenis dan tingkat otorisasinya |
| 6.1. <i>Confidentiality</i> | sejauh mana produk atau sistem memastikan bahwa data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses |
| 6.2. <i>Integrity</i> | sejauh mana sistem, produk atau komponen mencegah akses tidak sah ke, atau modifikasi, program atau data computer |
| 6.3. <i>Non-repudation</i> | sejauh mana tindakan atau peristiwa dapat dibuktikan telah terjadi, sehingga peristiwa atau tindakan tersebut tidak dapat dilakukan ditolak nanti |
| 6.4. <i>Accountability</i> | sejauh mana tindakan suatu entitas dapat dilacak secara unik ke entitas tersebut |
| 6.5. <i>Authenticity</i> | sejauh mana identitas subjek atau sumber daya dapat dibuktikan sebagai yang diklaim |
| 7. <i>Maintainability</i> | tingkat efektivitas dan efisiensi yang dengannya produk atau sistem dapat dimodifikasi sesuai tujuan pengelola |
| 7.1. <i>Modularity</i> | sejauh mana sistem atau program komputer terdiri dari komponen diskrit sedemikian rupa sehingga berubah menjadi satu komponen berdampak minimal pada komponen lainnya |
| 7.2. <i>Reusability</i> | sejauh mana suatu aset dapat digunakan di lebih dari satu sistem, atau dalam membangun aset lainnya |
| 7.3. <i>Analysability</i> | tingkat efektivitas dan efisiensi yang memungkinkan untuk menilai dampak pada produk atau system |
| 7.4. <i>Modifialibility</i> | sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menimbulkan cacat atau menurunkan kualitas produk yang ada |
| 7.5. <i>Testability</i> | tingkat keefektifan dan efisiensi yang dengannya kriteria pengujian dapat ditetapkan untuk sistem, produk atau komponen dan pengujian dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi |
| 8. <i>Portability</i> | tingkat efektivitas dan efisiensi yang dengannya sistem, produk, atau komponen dapat ditransfer satu perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan lainnya |
| 8.1. <i>Adaptability</i> | sejauh mana produk atau sistem dapat secara efektif dan efisien diadaptasi untuk berbeda atau berkembang perangkat keras, perangkat lunak |

| | |
|----------------------------|--|
| | atau lingkungan operasional atau penggunaan lainnya |
| 8.2. <i>Installability</i> | tingkat efektivitas dan efisiensi yang dengannya produk atau sistem dapat berhasil dipasang dan / atau dihapus instalasinya di lingkungan tertentu |
| 8.3. <i>Replaceability</i> | sejauh mana suatu produk dapat menggantikan produk perangkat lunak tertentu lainnya untuk tujuan yang sama dalam hal yang sama lingkungan Hidup |

Untuk menghitung hasil dari kuesioner ISO/IEC 25010 menggunakan analisis *gap*. Menurut Purnamawati dalam Diana (2015) nilai *gap* merupakan hasil selisih antara nilai kinerja aktual dengan nilai harapan. Semakin tinggi jumlah *gap* tersebut maka semakin rendah kepuasan pengguna yang dirasakan dan jika semakin rendah jumlah *gap* tersebut maka semakin besar kepuasan pengguna yang diperoleh. Untuk menghitung *gap* dilakukan perhitungan terhadap nilai rata-rata tingkat kepuasan dari kinerja dan harapan. Menurut Sambagia, dkk., (2021) perbandingan antara kinerja dan harapan akan menentukan urutan serta dapat digunakan untuk memprioritaskan atribut yang paling mempengaruhi kepuasan pengguna. Nilai rata-rata tingkat kinerja dan harapan dari masing-masing atribut akan dibandingkan dengan nilai rata-rata dari totalnya. Nilai rata-rata kinerja dan harapan setiap atribut dapat dihitung dengan menggunakan persamaan berikut:

$$\bar{x} = \frac{\sum x_i}{n} \quad \dots (2.1)$$

$$\bar{y} = \frac{\sum y_i}{n} \quad \dots (2.2)$$

Keterangan:

\bar{x} = Skor rata-rata kinerja atribut ke-i

\bar{y} = Skor rata-rata harapan atribut ke-i

n = Jumlah responden

Nilai total rata-rata kinerja dan harapan dapat dihitung menggunakan persamaan berikut:

$$\bar{\bar{x}} = \frac{\sum \bar{x}}{k} \quad \dots (2.3)$$

$$\bar{y} = \frac{\sum \bar{y}}{k} \quad \dots (2.4)$$

Keterangan:

\bar{x} = Skor total rata-rata kinerja

\bar{y} = Skor total rata-rata harapan

k = Jumlah atribut

Untuk mencari nilai *gap* dapat menggunakan persamaan berikut ini:

$$Gap = P - E \quad \dots (2.5)$$

Keterangan :

Gap = Kesenjangan

P = Presepsi pelanggan atas kinerja pelayanan aktual

E = Ekspektasi/ harapan

$\sum P - \sum E = 0$: artinya bahwa kenyataan sama dengan harapan pelanggan atau memuaskan pelanggan

Dalam metode ISO/IEC 25010 tahap pertama yaitu menghitung nilai rata-rata dari kinerja dan harapan, setelah itu menghitung nilai total rata-rata kinerja dan harapan kemudian mencari nilai *gap* kesenjangan antara skor tingkat kinerja aktual dan skor tingkat harapan pengguna. Jika hasil dari nilai *gap* tersebut bernilai negatif (-) maka dapat dikatakan kualitas layanan kurang baik artinya bahwa pengguna merasakan ketidakpuasan terhadap layanan tersebut. Tetapi jika hasil perhitungan nilai *gap* sama dengan nol (0) maka kualitas layanan dapat dikatakan baik artinya pengguna merasakan kepuasan terhadap layanan yang diberikan, dan apabila jumlah perhitungan nilai *gap* bernilai positif (+) maka kualitas jasa sangat baik artinya pengguna merasakan sangat puas terhadap layanan yang diberikan.

Menurut Stolzer dalam Jienardy (2017) *gap* positif akan diperoleh apabila nilai persepsi konsumen lebih besar dari nilai harapan. Konsumen dianggap puas terhadap perusahaan. Sedangkan *gap* negative akan diperoleh apabila nilai harapan konsumen lebih besar dari nilai persepsi. Hal ini menunjukkan konsumen kurang atau tidak puas terhadap perusahaan. Nilai harapan yang semakin tinggi dan nilai persepsi yang semakin rendah menunjukkan semakin besar *gap* yang terjadi.

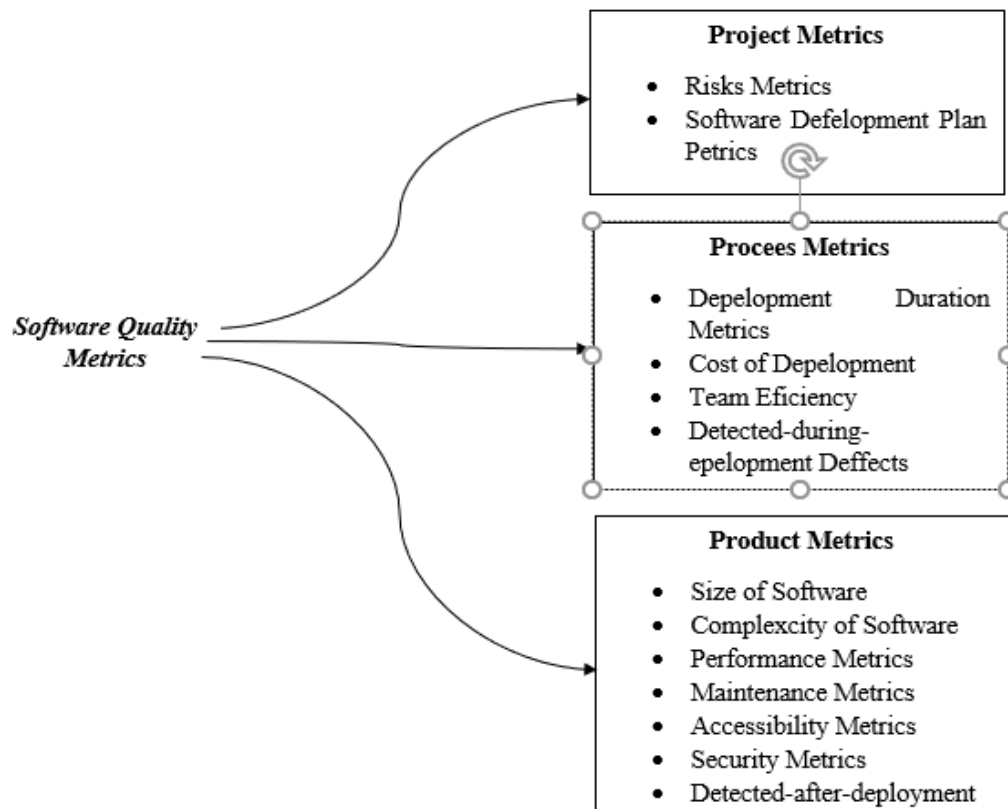
2.1.4 *Software Quality Metrics*

Menurut IEEE dalam Mladenova (2020) *Software Quality Metrics* merupakan fungsi yang inputnya adalah data perangkat lunak dan outputnya berupa nilai numerik tunggal yang dapat diinterpretasikan sebagai sejauh mana perangkat lunak memiliki atribut tertentu yang mempengaruhi kualitasnya. Metrik kualitas perangkat lunak dapat diterapkan untuk layanan dan proses perangkat lunak tidak hanya untuk program perangkat lunak itu sendiri. Tujuan dari metrik dan proses pengukuran adalah untuk menyediakan informasi yang dibutuhkan untuk perbaikan dan prediksi.

Menurut Hani'ah, dkk., (2015) *Software Quality Metrics* atau metrik kualitas perangkat lunak merupakan pengukuran yang berfokus pada penilaian kualitas produk, proses dan proyek. Namun metrik ini lebih erat kaitannya dengan metrik proses dan produk. Metrik produk dalam hal ini yaitu menggambarkan mengenai karakteristik dari suatu produk perangkat lunak tersebut. Pengukuran dapat diterapkan pada proses perangkat lunak agar dapat memperbaikinya secara terus menerus. Pengukuran dapat digunakan di seluruh proyek perangkat lunak untuk membantu dalam estimasi, kontrol kualitas, penilaian produktivitas dan pengendalian proyek.

Tujuan dari *Software Quality Metrics* adalah untuk memahami penyimpangan yang terjadi terkait fungsional yaitu kualitas kinerja yang direncanakan, serta untuk mengidentifikasi situasi yang memerlukan

pengembangan dalam bentuk pencegahan atau tindakan perbaikan untuk mencapai tujuan kualitas.



Gambar 2. 3 Gambar Klasifikasi *Software Metrics* (Mladenova, 2020)

Sumber: Mladenova (2020)

a. *Project Metrics*

Project Metrics atau Metrik Proyek digunakan pada tahap awal proyek untuk mengukur potensi risiko dan membuat rencana pengembangan yang stabil. Jenis metrik ini juga digunakan untuk memantau status dan biaya proyek berdasarkan dari pengalaman dan data yang dikumpulkan dari proyek sebelumnya.

b. *Procees Metrics*

Procees Metrics atau Metrik Proses adalah Jenis metrik yang mengamati proses pengembangan seperti informasi tentang cacat yang ditemukan saat pemograman perangkat lunak pada proyek yang sedang dikembangkan sekarang atau pada proyek sebelumnya, dan memberikan hasil yang dapat

memprediksi masalah di masa depan. Metrik ini juga digunakan ketika evaluasi efisiensi tim diperlukan.

c. *Product Metrics*

Product Metrics atau Metrik Produk adalah metrik yang paling luas dari tiga kategori. Metrik ini mempertimbangkan produk yang dikembangkan dan dilakukan pengujian pada akhir fase pengembangan.

1. *Performance Metrics*

Performance Metrics mengacu pada indikator seperti kecepatan, beban jaringan, beban kerja database. Dalam hal pengembangan web, terkadang metrik yang mengukur kinerja juga dapat mengukur jumlah pengunjung situs web dan perilakunya. Bergantung pada perilaku dan respons pengguna, aksesibilitas dan kegunaan dapat digunakan untuk menentukan apakah perangkat lunak memenuhi persyaratan dan apakah pengguna dapat bekerja dan menyesuaikan diri di dalamnya.

2. *Maintenance Metrics*

Maintenance Metrics terdiri dari *functional metrics*, *periodicity metrics*, *equipment metrics*. Sebagian besar metrik ini diukur setelah pengembangan perangkat lunak dan tidak jarang mereka mengukur hasil dari metrik lain, seperti metrik kinerja. Pemantauan berkala terhadap bagian-bagian penting perangkat lunak dapat mengurangi biaya perbaikan cacat.

3. *Security Metrics*

Security Metrics membantu organisasi untuk memverifikasi bahwa tindakan keamanan mereka sesuai dengan kebijakan, mengidentifikasi kekuatan dan kelemahan keamanan, dan mengidentifikasi tren keamanan. Beberapa metrik paling penting yang harus diukur yaitu:

- a. Waktu rata-rata untuk mendeteksi dan waktu rata-rata untuk merespons

- b. Jumlah pengguna dengan tingkat akses tertinggi
- c. Jumlah serangan setiap bulan

4. *Effectivness Metrics*

Indikator yang dapat menunjukkan apakah tes itu sendiri dirancang dengan baik dan sesuai dapat diukur dengan beberapa tinjauan metrik berikut ini:

- a. Persentase Kasus Uji yang lulus Lulus
- b. Kasus Uji Gagal
- c. Persentase Cacat Tetap
- d. Persentase Cacat yang diterima
- e. Persentase Cacat yang Ditolak
- f. Waktu rata-rata untuk Perbaikan Cacat
- g. Jumlah pengujian yang dijalankan per periode
- h. Uji Efisiensi Desain

2.1.5 Populasi dan Sampel

Sebelum melakukan pengujian perlunya dilakukan proses penentuan sample dari populasi. Besaran ukuran sampel berbanding lurus dengan besaran tingkat toleransi kesalahan (*error tolerance*), adapun beberapa tingkat toleransi kesalahan yaitu 5%, 10% dan 15%. Semakin sedikit jumlah sampel maka semakin besar tingkat kesalahan dan begitu pula sebaliknya. Menurut Sevilla dalam Supriyanto & Iswandiri (2017) salah satu cara perhitungan yang dapat digunakan untuk menentukan jumlah dari sampel adalah dengan Rumus Slovin yang dapat dilihat pada persamaan 2.6.

$$n = \frac{N}{1+Ne^2} \quad \dots(2.6)$$

Keterangan:

n = Jumlah Sampel

N = Jumlah Populasi

e = Batas Toleransi Kesalahan (*error tolerance*).

2.1.6 Pengujian Validitas dan Reliabilitas

Pengujian validitas dan reliabilitas merupakan suatu proses pengujian yang dilakukan untuk mengetahui valid atau tidak valid suatu kuesioner.

2.1.6.1. Pengujian Validitas

Pengujian validitas dalam penelitian ini digunakan untuk menganalisis pertanyaan, apakah pertanyaan memenuhi syarat atau tidak, jika tidak maka pertanyaan tersebut tidak dapat digunakan lagi. Menurut Sugiharto dan Sitinjak dalam Firdiana (2018) syarat yang harus dipenuhi yaitu memiliki kriteria sebagai berikut :

1. Jika pada rhitung bernilai positif, serta $r_{hitung} \geq r_{tabel}$ maka suatu kuesioner dinyatakan valid dan H_0 diterima
2. Jika pada rhitung bernilai negatif, serta $r_{hitung} \leq r_{tabel}$ maka kuesioner dinyatakan tidak valid dan H_0 ditolak

2.1.6.2 Pengujian Reliabilitas

Menurut Janti (2014) Uji reliabilitas digunakan untuk melihat apakah instrumen yang ada dalam kuesioner dapat digunakan lebih dari satu kali, paling tidak oleh responden yang sama akan menghasilkan data yang konsisten. Uji realibilitas merupakan pengujian yang dilakukan untuk memastikan apakah pertanyaan penelitian atau kuesioner yang digunakan dalam proses pengumpulan data reliabel atau tidak. Syarat nilai terkecil yang dianggap memenuhi adalah apabila koefisien dari *cronbach's alpha* (ukuran umum dari konsistensi internal skala) yang dihasilkan adalah = 0,6. Sedangkan jika koefisien yang dihasilkan kurang dari 0,6 maka pertanyaan penelitian tersebut dinyatakan tidak reliabel. Apabila dalam pengujian pertanyaan sudah dinyatakan valid dan reliabel, maka pertanyaan tersebut dapat digunakan sebagai parameter pengukuran dalam proses pengumpulan data.

2.1.7 Skala Likert

Menurut Harun (2018) Skala Likert merupakan skala penelitian yang digunakan untuk mengukur suatu sikap dan pendapat dari responden terhadap sebuah peristiwa baik itu berupa tanggapan positif ataupun negatif.

Dalam menanggapi pertanyaan menggunakan skala likert, responden memberikan tingkat persetujuannya terhadap suatu pertanyaan dengan cara mengisi pertanyaan dengan memilih salah satu dari pilihan yang tersedia. Skala yang dipakai untuk jawaban dari kuesioner pada bagian pertanyaan yang diajukan dapat dilihat pada tabel berikut ini:

Tabel 2. 3 Nilai Pernyataan (Kustijono, dkk., 2018)

| No | Pernyataan | Keterangan |
|----|------------|---------------|
| 1 | 1 | Sangat Kurang |
| 2 | 2 | Kurang Baik |
| 3 | 3 | Baik |
| 4 | 4 | Sangat Baik |

2.2. Tinjauan Pustaka

Tabel 2. 4 Tinjauan Pustaka

| No | Judul Penelitian dan Tahun | Deskripsi | Metode | Hasil |
|----|---|--|---|---|
| 1 | "Pengukuran Kesesuaian Fungsional Dengan Pendekatan Berorientasi Tujuan Pada Sistem Informasi Akademik (Sia) Berdasarkan Model Kualitas ISO/IEC 25010" (2017) | Penelitian ini melakukan pengukuran kualitas sistem informasi akademik menggunakan model pengukuran ISO/IEC 25010 yaitu terkait kesesuaian fungsional dari SIA tersebut. Pengukuran yang dilakukan | ISO/IEC 25010 dan <i>Goal Question Metric</i> | Dari hasil pada penelitian terdapat perbedaan nilai pengukuran dari kedua institusi. Pada siacad IAIN Raden Intan di Lampung masih memerlukan |

| | | | | |
|---|--|---|--|---|
| | | berorientasi pada tujuan dari institusi dipetakan menggunakan <i>Goal Question Metric</i> (GQM) | | perbaikan terutama bagi karyawan karena masih memiliki nilai yang belum mencapai standar kualitas aplikasi <i>web</i> . Sedangkan siakad ITS di Surabaya sudah memiliki nilai sesuai standar. |
| 2 | “Pengukuran Kesesuaian Fungsional Menggunakan Pendekatan Berorientasi Tujuan Pendekatan ISO / IEC 25010 Studi Kasus <i>Website</i> Sinta Purbalingga” (2019) | Penelitian ini melakukan pengukuran kesesuaian fungsional kualitas dari <i>website</i> yaitu <i>website</i> dari Sinta Purbalingga pengukuran dilihat berdasarkan perspektif dari pengelola <i>website</i> tersebut | Pada penelitian ini menggunakan pendekatan berbasis tujuan pada ISO / IEC 25010. | Hasil pengukuran menggunakan ISO 25010 memperoleh hasil 73,6 dari hasil tersebut menunjukkan bahwa <i>website</i> tersebut telah memenuhi standar kualitas yang baik hasil dari rata-rata kualitasnya mencapai 3,68 yang berarti kualitasnya telah memenuhi standar |

| | | | | |
|---|---|--|---------------|--|
| 3 | “Evaluasi Kualitas Perangkat Lunak Dengan ISO/IEC 25010:2011 (Studi Kasus : Aplikasi First Aid Pada <i>Platform Android</i>)” (2018) | Penelitian ini membahas tentang evaluasi dari kualitas aplikasi first aid yang berbasis android, evaluasi dilakukan menggunakan model kualitas dari ISO/IEC 25010:2011 | ISO/IEC 25010 | Hasil dari pengujian diperoleh hasil yang paling berpengaruh dan memperoleh nilai yang paling tinggi yaitu <i>Portability</i> 77,7% sehingga dapat menjadi acuan untuk perbaikan terhadap pemetaan masalah yang terkait dengan kualitas perangkat lunak tersebut |
|---|---|--|---------------|--|

Berdasarkan penelitian terdahulu yang digunakan terdapat persamaan dan perbedaan dari penelitian yang akan dilakukan. Persamaannya adalah salah satu model yang digunakan sama yaitu ISO/IEC 25010 namun pada judul penelitian pertama menggabungkan pendekatan *Goal Question Metric* (GQM) dan ISO/IEC 25010 terkait kesesuaian fungsional dengan menggunakan 2 objek penelitian sekaligus yaitu perguruan tinggi dan menggunakan 3 kategori responden (mahasiswa, dosen, dan karyawan) dari setiap objek penelitiannya, dan pada penelitian yang kedua juga menggunakan pendekatan berbasis pada tujuan berdasarkan model ISO/IEC 25010 dan pada objek penelitiannya adalah sebuah *website*, untuk mengukur kualitas dari kesesuaian fungsionalnya peneliti hanya menggunakan perspektif dari pengelola *website* tersebut yang dijadikan sebagai respondennya. Pada penelitian yang ketiga menggunakan model ISO/IEC 25010

untuk melakukan pengukuran dari delapan karakteristik yang dimiliki ISO/IEC 25010 dan menggunakan responden secara acak.

Perbedaannya dengan penelitian yang dilakukan adalah studi kasus atau objek penelitian yang diteliti serta metode yang digunakan yaitu *Software Quality Metrics* dan ISO/IEC 25010. Salah satu metode yang digunakan yaitu *Software Quality Metrics* merupakan metode yang masih jarang digunakan pada penelitian dengan menggabungkan kedua metode sekaligus yaitu penggabungan metode *Software Quality Metrics* dan ISO/IEC 25010.