BAB II
LANDASAN TEORI

II.1 Kualitas Perangkat Lunak

Definisi kualitas perangkat lunak menurut IEEE terbagi menjadi dua, di antaranya [2]:

1. Sejauh mana sistem, komponen, atau proses dapat memenuhi semua kebutuhan.
2. Sejauh mana sistem, komponen, atau proses dapat memenuhi kebutuhan ataupun harapan pengguna.

Sebuah perangkat lunak dikatakan berkualitas saat banyak pengguna yang menggunakannya, sebaliknya akan dikatakan gagal apabila pengguna meninggalkannya karena beberapa kesalahan yang dimiliki perangkat lunak tersebut. Seperti yang di kutip oleh Wayan Gede Suka Parwita dari buku berjudul "Quality is free: the art of making quality certain" karangan Philip B. Crosby: kesalahan pertama adalah dengan mengasumsikan bahwa kualitas berarti kebaikan atau kemewahan. Crosby dengan jelas mengatakan bahwa "kesesuaian dengan spesifikasi" adalah definisi dari kualitas. Crosby juga menekankan bahwa mendefinisikan kualitas secara jelas sangat penting untuk dapat mengukur dan mengelola konsep kualitas itu sendiri [3]. David Garvin dari Harvard Business School menekankan bahwa kualitas itu kompleks dan memiliki beragam konsep yang dapat dijelaskan dari lima sudut pandang yang berbeda, di antaranya *The transcendental view*, *The user view*, *The manufacturer's view*, *The product view*, dan *The value-based view* [4].

II.2 Domain Perangkat Lunak

Domain perangkat lunak merupakan kategori dari setiap jenis perangkat lunak yang ada. Terdapat tujuh kategori mengenai jenis domain perangkat lunak ini sendiri, diantaranya [4]:

1. *System software*

   System software merupakan sebuah program yang dibuat untuk mendukung program lain untuk dapat digunakan. Perangkat lunak jenis ini misalnya *compilers, editor, file management, operating system, telecommunications processors,* dan lain-lain.

2. *Application software*

   Application software adalah sebuah program yang berdiri sendiri dan digunakan untuk mengatasi kebutuhan bisnis yang spesifik.

3. *Engineering/scientific software*

   Perangkat lunak pada domain ini biasanya ditekankan pada penggunaan algoritma. Penggunaan perangkat lunak ini terdapat pada kebutuhan seperti astronomi, vulkanologi, pabrik, biologi, dan lain sebagainya.

4. *Embedded software*

   Embedded software merupakan perangkat lunak yang ditanam pada suatu sistem. Perangkat lunak ini digunakan dalam mengatur fungsi untuk pengguna maupun untuk dirinya sendiri.

5. *Product-line software*

   Perangkat lunak pada domain *product-line software* dibuat untuk membantu kebutuhan pengguna yang bersifat spesifik yang dapat digunakan oleh pengguna yang berbeda. Contoh dari perangkat lunak pada domain *product-line* software diantaranya untuk keperluan *word processing, multimedia, computer graphic, database management, entertainment,* dan lain sebagainya.

6. *Web application*

   Web application atau biasa disebut *webapps* adalah perangkat lunak yang berbasis *website*. Pada perangkat lunak ini bukan hanya sekedar menampilkan informasi berbentuk teks namun dapat juga berupa gambar.

7. *Artificial intelligence software*

   Perangkat lunak pada domain ini ditekankan pada algoritma untuk dapat menyelesaikan suatu masalah yang kompleks, yang tidak bisa diselesaikan

dengan perhitungan ataupun analisis langsung. Perangkat lunak ini seperti untuk pengenalan pola, jaringan syaraf tiruan, robotik, dan lain-lain.

II.3 Perangkat Lunak Pengolah Animasi Tiga Dimensi

Animasi adalah sekumpulan gambar statis yang digerakkan dalam suatu waktu sehingga menimbulkan efek seolah-olah gambar tersebut dapat bergerak. Awalnya animasi hanya merupakan gambar tangan yang dibuat dalam beberapa lembar kertas dan digerakkan dengan cepat dalam suatu waktu. Namun animasi kini sudah mengalami perkembangan pesat yakni dengan munculnya animasi digital seperti halnya pada film kartun. Animasi terbagi menjadi dua bentuk, yakni animasi dua dimensi dan animasi tiga dimensi. Animasi dua dimensi merupakan animasi yang datar karena hanya terbentuk dari sumbu X dan Y saja. Sedangkan animasi tiga dimensi adalah animasi yang memiliki bentuk hampir sama dengan bentuk asli sebuah benda, pada animasi tiga dimensi ini dapat dilihat dari tiga sudut pandang, yakni sumbu X, Y, dan Z.

Saat ini banyak sekali bermunculan film-film dengan konsep animasi tiga dimensi, karena memiliki gambar yang lebih baik dan tampak lebih nyata seperti bentuk aslinya. Dalam pembangunan animasi tiga dimensi ini tidak terlepas dari perangkat lunak khusus untuk mengolah animasi. Beberapa perangkat lunak pengolah animasi tiga dimensi yang marak di pasaran adalah Blender dan K-3D.

Blender dan K-3D merupakan perangkat lunak pengolah animasi tiga dimensi yang berbasis *open source*. Dengan kata lain siapapun dapat ikut serta membantu pengembangan perangkat lunak ini, karena pihak developer membebaskan pengguna untuk ikut serta mengubah, menghapus, ataupun menambahkan kekurangan yang ada pada perangkat lunak tersebut agar kedepannya dapat menjadi lebih baik.

II.4 Model Kualitas Perangkat Lunak

Dalam keilmuan rekayasa perangkat lunak terdapat beberapa model kualitas perangkat lunak. Pada setiap model ini terdapat beberapa faktor yang menjadi poin-poin utama dalam penilaian kualitas sebuah perangkat lunak. Berikut ini dua model

kualitas perangkat lunak yang dapat digunakan dalam penilaian kualitas sebuah perangkat lunak.

1. Model ISO-9126

   Model ISO-9126 dikenalkan pertama kali pada tahun 1991 sebagai standarisasi kualitas produk perangkat lunak [5]. Standarisasi ini dibuat karena banyaknya model kualitas yang ditawarkan sebagai faktor kualitas perangkat lunak. Dalam dokumen pertama model ISO-9126 terdiri dari dua bagian model kualitas untuk kualitas sebuah produk perangkat lunak, di antaranya [5]:

   1. *Internal and external quality model*.
   2. *Quality in use model*.

   Bagian pertama dari dua kualitas model tersebut menentukan 6 karakteristik yang mereka bagi kedalam 21 sub karakteristik untuk kualitas internal dan kualitas eksternal yang dapat dilihat pada Tabel II-I.

Tabel II-I Faktor Kualitas Internal dan Eksternal

External and Internal Quality	
Functionality	*Suitability* *Accuracy* *Interoperability* *Security* *Functionality Compliance*
Reliability	*Maturity* *Fault Tolerance* *Recoverability* *Reliability Compliance*
Usability	*Understandability* *Learnability* *Operability* *Attractiveness* *Usability Compliance*
Efficiency	*Time Behavior* *Resource Utilization* *Efficiency Compliance*
Maintainability	*Analyzability* *Changeability* *Stability* *Testability* *Maintainability Compliance*
Portability	*Adaptability* *Installability*

External and Internal Quality	
	Co-existence *Replaceability* *Portability Compliance*

Berikut ini merupakan pengertian dari masing-masing faktor kualitas dan sub-faktor yang terdapat pada model *internal dan eksternal* pada ISO-9126, antara lain:

1. *Functionality*

   Faktor ini berhubungan dengan seberapa jauh sebuah perangkat lunak dapat memenuhi kebutuhan penggunanya.

   a. *Suitability*

      Sub-faktor *suitability* berhubungan dengan tingkat kemampuan dan kelayakan dari sebuah perangkat lunak untuk dapat menyediakan fungsionalitas untuk kebutuhan yang spesifik.

   b. *Accuracy*

      Accuracy berhubungan dengan kemampuan sebuah perangkat lunak untuk dapat memberikan hasil yang benar dan sesuai dengan fungsi yang dijalankan.

   c. *Interoperability*

      Sub-faktor *interoperability* menggambarkan tentang kemampuan sebuah perangkat lunak untuk dapat berinteraksi dengan sistem yang lain atau dengan sistem tertentu.

   d. *Security*

      Sub-faktor ini berhubungna dengan keamanan yang dimiliki oleh sebuah perangkat lunak. Keamanan yang dimaksud dapat berupa pemberian hak akses kepada penggunanya.

   e. *Functionality Compliance*

      Functionality compliance menggambarkan tentang kesesuaian antara fungsionalitas yang ada pada sebuah perangkat lunak dengan aturan standar yang berlaku pada perangkat lunak lainnya yang sejenis.

2. *Reliability*

   Faktor *reliability* berhubungan dengan keandalan atau seberapa baik tingkat pelayanan sebuah perankat lunak apabila digunakan dalam kondisi tertentu.

   a. *Maturity*

      Maturity berhubungan dengan kelayakan sebuah perangkat lunak dalam menangani kegagalan atau kesalahan yang terdapat didalamnya.

   b. *Fault Tolerance*

      Fault tolerance merupakan sub-faktor *reliability* yang menggambarkan tentang seberapa jauh sebuah perangkat lunak dapat memberikan toleransi kegagalan atas kesalahan yang terjadi saat digunakan pada kondisi tertentu.

   c. *Recoverability*

      Sub-faktor ini berhubungan dengan kemampuan sebuah perangkat lunak untuk dapat membangun kembali kinerja dan memulihkan data baik secara langsung maupun secara tidak langsung.

   d. *Reliability Compliance*

      Reliability compliance berhubungan dengan kesesuaian antara tingkat pelayanan yang diberikan oleh perangkat lunak dengan aturan standar yang digunakan pada setiap perangkat lunak yang memiliki tipe yang sama.

3. *Usability*

   Usability berhubungna dengan seberapa baik perangkat lunak dapat dipahami, dipelajari, dan digunakan.

   a. *Understandability*

      Sub-faktor ini berhubungan dengan seberapa jauh sebuah perangkat lunak dapat dipahami oleh pengguna baik dari secara konsep logis dan penerapan penggunaan perangkat lunak tersebut.

b. *Learnability*

Learnability menggambarkan tentang sebuah perangkat lunak dapat dipelajari dengan baik oleh penggunanya.

c. *Operability*

Operability berhubungan dengan seberapa jauh perangkat lunak dapat dioperasikan oleh penggunanya.

d. *Attractiveness*

Sub-faktor ini menggambarkan tentang bagaimana sebuah perangkat lunak dapat menarik perhatian bagi penggunanya.

e. *Usability Compliance*

Sub-faktor ini berhubungan dengan kesesuaian antara kegunaan perangkat lunak dengan standar yang digunakan oleh perangkat lunak sejenis lainnya.

4. *Efficiency*

Faktor *efficiency* berhubungan dengan efisiensi dari seberapa besar sumber daya yang digunakan oleh sebuah perangkat lunak.

a. *Time Behaviour*

Time behaviour berhubungan dengan waktu yang dibutuhkan oleh sebuah perangkat lunak untuk menjalankan suatu fungsi atau tujuan tertentu.

b. *Resource Utilization*

Sub-faktor ini menggambarkan tentang jumlah sumber daya yang digunakan oleh perangkat lunak untuk melakukan fungsi atau tujuan tertentu.

c. *Efficiency Compliance*

Efficiency compliance berhubungan dengan kesesuaian antara sumber daya yang digunakan oleh sebuah perangkat lunak dengan standar penggunaan sumber daya pada perangkat lunak lainnya yang sejenis.

5. *Maintainability*

Maintainability menggambarkan tentang pemeliharaan sebuah perangkat lunak, seberapa baik perangkat lunak tersebut dapat dipertahankan.

a. *Analyzability*

Analyzability berhubungan dengan seberapa jauh sebuah perangkat lunak dapat di analisis, hal ini diperlukan untuk analisis kekurangan atau penyebab kegagalan agar dapat diketahui bagian mana yang perlu dimodifikasi.

b. *Changeability*

Changeability berhubungan dengan seberapa baik perangkat lunak dapat diubah, upaya ini diperlukan untuk modifikasi, penghapusan kesalahan atau perubahan lingkungan.

c. *Stability*

Sub-faktor ini berhubungan dengan stabilitas dari sebuah perangkat lunak yang memungkinkan untuk menyimpulkan resiko efek tak terduga yang disebabkan oleh modifikasi.

d. *Testability*

Testability menggambarkan tentang bagaimana perangkat lunak dapat diuji, hal ini untuk menyimpulkan tentang upaya yang diperlukan untuk memvalidasi perangkat lunak dan cakupan pengujian.

e. *Maintainability Compliance*

Hal ini berhubungan dengan kesesuaian antara pemeliharaan yang dilakukan terhadap perangkat lunak dengan standarisasi yang terdapat pada pemeliharaan perangkat lunak lainnya yang sejenis.

6. *Portability*

Portability menggambarkan tentang kemampuan sebuah perangkat lunak untuk dapat berpindah dari sebuah lingkungan atau sistem ke sistem lainnya.

a. *Adaptability*

Sub-faktor ini berhubungan dengan seberapa jauh sebuah perangkat lunak dapat beradaptasi dengan perubahan lingkungan atau sistem yang berbeda.

b. *Installability*

Installability menggambarkan tentang seberapa baik perangkat lunak dapat digunakan dalam lingkungan atau sistem tertentu.

c. *Co-existence*

Co-existence berhubungan dengan bagaimana perangkat lunak dapat berdampingan dengan produk atau perangkat lunak lain pada suatu lingkungan atau sistem yang sama untuk mengetahui tentang dependensi, perilaku, atau efek samping yang ditimbulkan.

d. *Replaceability*

Replaceability berhubungan dengan bagaimana sebuah perangkat lunak dapat menggantikan perangkat lunak lain apakah ada kebergantungan kepada perangkat lunak lain saat perangkat lunak tersebut digunakan.

e. *Portability Compliance*

Portability compliance berhubungan dengan kesesuaian antara perubahan yang dapat dilakukan oleh sebuah perangkat lunak dengan standarisasi *portability* yang terdapat pada perangkat lunak lain yang sejenis.

Sedangkan untuk model *quality in use* pada ISO-9126 terdapat empat faktor yang ada didalamnya seperti yang dapat dilihat pada Gambar II-I, antara lain:

1. *Effectiveness*

   Effectiveness berhubungan dengan kemampuan untuk mencapai tujuan pengguna melalui akurasi dan kelengkapan perangkat lunak.

2. *Productivity*

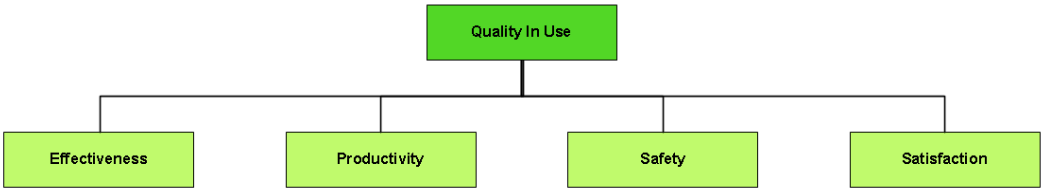
   Productivity merupakan upaya perangkat lunak dalam menghindari kelebihan penggunaan sumber daya untuk mencapai tujuan pengguna.

3. *Safety*

   Safety merupakan kemampuan perangkat lunak untuk dapat mengurangi tingkat kegagalan pada pengguna lain.

4. *Satisfaction*

   Satisfaction merupakan tingkat kepuasan pengguna dalam menggunakan sebuah perangkat lunak.



Gambar II-I Model *Quality In Use* pada ISO-9126

2. Model ISO-25010

   Model ini merupakan bagian dari Software product Quality Requirements and Evaluation (SQuaRE), dimana model ini berkaitan dengan model kualitas perangkat lunak yang merupakan pengembangan dari model sebelumnya. Pada model ini terdapat beberapa sub-karakteristik tambahan dan beberapa sub-karakteristik yang dipindahkan ke karakteristik lain. Berikut ini merupakan karakteristi atau faktor kualitas internal dan eksternal yang terdapat pada model ISO-25010, dapat dilihat pada Tabel II-II.

Tabel II-II Faktor dan Sub-faktor pada Model ISO-25010

Faktor	Sub-faktor
Functional Suitability	1. *Appropriateness* 2. *Accuracy* 3. *Functional Suitability Compliance*
Reliability	1. *Availability* 2. *Fault Tolerance* 3. *Recoverability* 4. *Reliability Compliance*
Performance Efficiency	1. *Time-behaviour* 2. *Resource-utilisation* 3. *Performance Efficiency Compliance*
Operability	1. *Appropriateness Recognisability* 2. *Learnability* 3. *Ease of Use* 4. *Helpfulness* 5. *Attractiveness* 6. *Technical Accessibility* 7. *Operability Compliance*

Security	1. *Confidentiality* 2. *Integrity* 3. *Non-repudiation* 4. *Accountability* 5. *Authenticity* 6. *Security Compliance*
Compatibility	1. *Replaceability* 2. *Co-existence* 3. *Interoperability* 4. *Compatibility Compliance*
Maintainability	1. *Modularity* 2. *Reusability* 3. *Analyzability* 4. *Changeability* 5. *Modification Stability* 6. *Testability* 7. *Maintainability Compliance*
Transferability	1. *Portability* 2. *Adaptability* 3. *Installability* 4. *Transferability Compliance*

Berikut ini merupakan pengertian dari masing-masing faktor dan sub-faktor yang terdapat pada model ISO-25010, antara lain:

1. *Functional Suitability*

   Functional suitability merupakan tingkat dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi yang spesifik.

   a. *Appropriateness*

      Appropriateness merupakan tingkat dimana perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna tertentu.

   b. *Accuracy*

      Accuracy merupakan tingkat dimana perangkat lunak dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan.

   c. *Functional Suitability Compliance*

      Functional suitability compliance merupakan tingkat dimana perangkat lunak memenuhi standar *functional suitability* yang terdapat pada perangkat lunak lainnya yang sejenis.

2. *Reliability*

   Reliability merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik.

   a. *Availability*

      Availability merupakan tingkatan dimana komponen perangkat lunak dapat beroperasi dan siap ketika dibutuhkan untuk digunakan.

   b. *Fault Tolerance*

      Fault tolerance merupakan tingkat dimana sebuah perangkat lunak dapat bertahan pada tingkat kemampuan tertentu terhadap kegagalan atau kesalahan yang terdapat pada perangkat lunak.

   c. *Recoverability*

      Recoverability merupakan tingkat dimana perangkat lunak dapat kembali pada tingkat kemampuan tertentu dan melakukan pengembalian data secara langsung yang disebabkan oleh kegagalan atau kesalahan yang terjadi pada perangkat lunak.

   d. *Reliability Compliance*

      Reliability compliance merupakan tingkat dimana perangkat lunak dapat memenuhi standar ketahanan yang dimiliki oleh perangkat lunak lain sejenis.

3. *Performance Efficiency*

   Performance efficiency merupakan tingkat dimana perangkat lunak dapat memberikan kinerja yang tepat terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu.

   a. *Time-behaviour*

      Time-behaviour merupakan tingkat dimana perangkat lunak dapat memberikan reaksi dan waktu yang dibutuhkan ketika melakukan aksi dari sebuah fungsi pada kondisi tertentu.

   b. *Resource-utilisation*

      Resource-utilisation merupakan tingkat dimana sebuah perangkat lunak menggunakan sejumlah dan beberapa sumber daya ketika

perangkat lunak melakukan aksi dari sebuah fungsi pada kondisi tertentu.

c. *Performance Efficiency Compliance*

Performance efficiency compliance merupakan tingkatan dimana perangkat lunak memenuhi standar yang berhubungan dengan efisiensi kinerja perangkat lunak.

4. *Operability*

Operability merupakan tingkat dimana perangkat lunak dapat dimengerti, dipelajari, digunakan, dan menarik perhatian pengguna ketika digunakan pada kondisi tertentu.

a. *Appropriateness Recognisability*

Appropriateness recognisability merupakan tingkat dimana perangkat lunak memungkinkan untuk dianalisis oleh pengguna apakah perangkat lunak sudah dapat memenuhi kebutuhan mereka.

b. *Learnability*

Learnability merupakan tingkat dimana perangkat lunak mudah untuk dipelajari penggunaannya oleh para pengguna.

c. *Ease of Use*

Ease of use merupakan tingkat dimana sebuah perangkat lunak dapat digunakan dan dioperasikan oleh para penggunanya.

d. *Helpfulness*

Helpfulness merupakan tingkat dimana perangkat lunak dapat memberikan bantuan ketika pengguna membutuhkan panduan.

e. *Attractiveness*

Attractiveness merupakan tingkat dimana perangkat lunak dapat menarik perhatian bagi para penggunanya.

f. *Technical Accessibility*

Technical accessibility merupakan tingkat dimana penggunaan perangkat lunak untuk memenuhi kebutuhan pengguna yang memiliki keterbatasan.

g. *Operability Compliance*

Operability compliance merupakan tingkat dimana perangkat lunak dapat memenuhi standarisasi dan panduan yang berhubungna dengan penggunaan perangkat lunak.

5. *Security*

Security merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman, akses atau penggunaan dari pengguna yang tidak dikenal.

a. *Confidentially*

Confidentially merupakan tingkat dimana perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna, apakah sebagai ancaman atau kejengajaan.

b. *Integrity*

Integrity merupakan tingkat dimana kelengkapan dan ketepatan dari sejumlah asset telah dijaga.

c. *Non-repudiation*

Non-repudiation merupakan tingkatan dimana aksi atau tindakan yang dilakukan telah terbukti sehingga hal tersebut tidak dapat ditolak.

d. *Accountability*

Accountability merupakan tingkat dimana aksi dari sebuah entitas dapat ditelusuri keunikannya terhadap entitas.

e. *Authenticity*

Authenticity merupakan tingkat dimana identitas dari sebuah subjek atau sumber dapat diterima.

f. *Security Compliance*

Security compliance merupakan tingkat dimana perangkat lunak dapat memenuhi standarisasi yang berhubungan dengan keamanan perangkat lunak.

6. *Compatibility*

Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak untuk dapat melakukan pertukaran informasi dan

melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.

a. Replaceability

Replaceability merupakan tingkat dimana perangkat lunak dapat digunakan pada perangkat lunak lain dengan tujuan yang spesifik pada lingkungan yang sama.

b. Co-existence

Co-existence merupakan tingkat dimana perangkat lunak dapat berdampingan dengan perangkat lunak lain dan berbagi sumber daya pada lingkungan yang sama.tanpa dampak yang merugikan.

c. Interoperability

Interoperability merupakan tingkat dimana perangkat lunak dapat beroperasi secara kooperatif dengan satu atau lebih perangkat lunak lain.

d. Compatibility Compliance

Compatibility compliance merupakan tingkat dimana perangkat lunak dapat memenuhi standarisasi terkait *compatibility* yang terdapat pada perangkat lunak sejenis lainnya.

7. Maintainability

Maintainability merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Modifikasi ini termasuk perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan, kebutuhan, dan fungsionalitas yang spesifik.

a. Modularity

Modularity merupakan tingkatan dimana sistem atau program terdiri dari komponen yang berlainan sehingga perubahan terhadap satu komponen minimal memiliki pengaruh terhadap komponen lain.

b. Reusability

Reusability merupakan tingkat dimana sebuah asset dapat digunakan pada lebih dari satu sistem perangkat lunak atau pada pembangunan asset lainnya.

c. *Analyzability*

Analyzability merupakan tingkat dimana perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak atau untuk mengidentifikasi bagian yang dapat dimodifikasi.

d. *Changeability*

Changeability merupakan tingkat dimana perangkat lunak memungkinkan sebuah modifikasi yang spesifik untuk dapat di implementasikan. Hal ini memudahkan perangkat lunak dapat untuk dimodifikasi.

e. *Modification Stability*

Modification stability merupakan tingkat dimana perangkat lunak dapat menghindari efek yang tidak diharapkan dari modifikasi yang dilakukan terhadap perangkat lunak.

f. *Testability*

Testability merupakan tingkat dimana perangkat lunak memungkinkan modifikasi perangkat lunak untuk dilakukan validasi.

g. *Maintainability Compliance*

Maintainability compliance merupakan tingkatan dimana sebuah perangkat lunak dapat memenuhi standarisasi yang berhubungan dengan *maintainability* pada perangkat lunak lain.

8. *Transferability*

Transferability merupakan tingkat dimana perangkat lunak dapat berpindah dari lingkungan yang satu ke lingkungan yang lain.

a. *Portability*

Portability merupakan kemudahan dimana sistem atau komponen dapat berpindah dari lingkungan *hardware* atau perangkat yang satu ke *hardware* atau perangkat lunak lain.

b. *Adaptability*

Adaptability merupakan tingkat dimana perangkat lunak dapat beradaptasi pada spesifikasi lingkungan yang berbeda tanpa

menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.

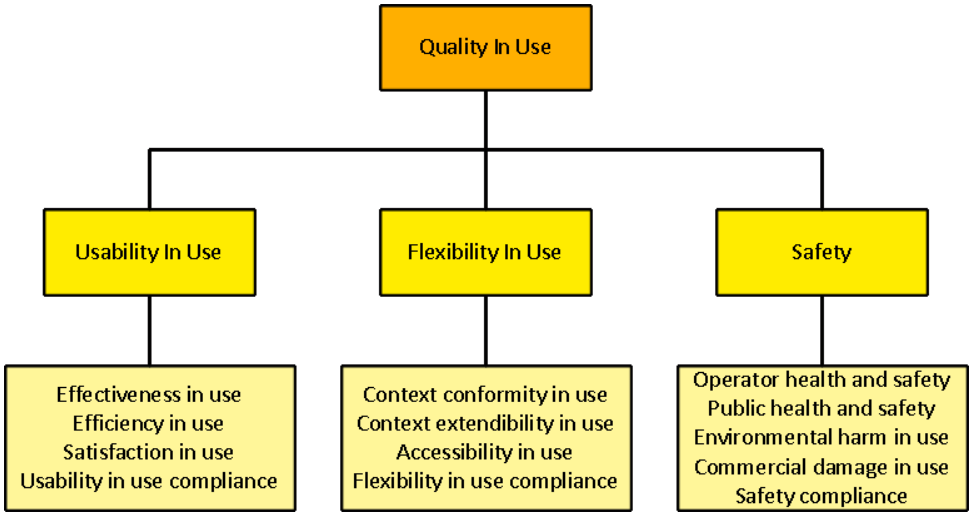
c. *Installability*

Installability merupakan tingkat dimana perangkat lunak dapat dipasang dan dihapus pada lingkungan yang spesifik.

d. *Transferability Compliance*

Transferability compliance merupakan tingkat dimana sebuah perangkat lunak dapat memenui standar yang berhubungan dengan kemampuan perangkat lunak untuk dapat berpindah dari satu lingkungan ke lingkungan lain.

Selain faktor internal dan eksternal yang terdapat pada model ISO-25010, terdapat pula faktor *quality in use* seperti pada model sebelumnya, hanya saja terdapat beberapa perubahan pada model ini yang dapat dilihat pada Gambar II-II.



Gambar II-II *Model Quality In Use* pada ISO-25010

1. *Usability In Use*

Usability in use merupakan tingkatan dimana pengguna dapat mencapai tujuannya dengan penggunaan yang efektif, efisien, dan kepuasan pada penggunaannya.

a. *Effictiveness In Use*

   Effectiveness in use merupakan tingkat dimana pengguna dapat mencapai tujuannya dengan ketepatan dan kesesuaian pengunaan.

b. *Efficiency In Use*

   Efficiency in use merupakan tingkat dimana pengguna dapat mengurangi sejumlah sumber daya untuk efektifitas penggunaan.

c. *Satisfaction In Use*

   Satisfaction in use merupakan tingkatan dimana seorang pengguna mendapatkan kepuasan saat menggunakan perangkat lunak.

d. *Usability In Use Compliance*

   Usability in use compliance berhubungan dengan kesesuaian penggunaan perangkat lunak dengan standarisasi penggunaan yang ada.

2. *Flexibility In Use*

   Flexibility in use merupakan tingkat dimana perangkat lunak dapat digunakan pada berbagai penggunaan.

a. *Context Conformity In Use*

   Context conformity in use merupakan tingkat sejauh mana perangkat lunak digunakan dalam hal penggunaannya.

b. *Context Extendibility In Use*

   Context extendibility in use merupakan tingkat sejauh mana perangkat lunak digunakan dalam hal diluar penggunaannya.

c. *Accessibility In Use*

   Accessibility in use merupakan tingkat penggunaan untuk pengguna dengan keterbatasan kemampuan.

d. *Flexibility In Use Compliance*

   Flexibility in use compliance berhubungan dengan kesesuaian dengan standarisasi terkait *flexibility*.

3. *Safety*

   Safety berhubungan dengan tingkatan penerimaan resiko dari kegagalan terhadap data, perangkat lunak, atau lingkungan yang termasuk dalam konteks penggunaan.

   a. *Operator Health and Safety*

      Operator health and safety merupakan tingkat penerimaan dari resiko kegagalan terhadap pengguna pada konteks penggunaan.

   b. *Public Health and Safety*

      Public health and sfety merupakan tingkat penerimaan dari resiko kegagalan terhadap orang lain pada konteks penggunaan.

   c. *Environmental Harm In Use*

      Environmental harm in use merupakan tingkat penerimaan dari resiko kegagalan yang dialami lingkungan dalam konteks penggunaan.

   d. *Commercial Damage In Use*

      Commercial damage in use merupakan tingkat penerimaan dari resiko kegagalan terhadap kesalahan yang dapat mengarah kepada kerusakan reputasi pada konteks penggunaan.

   e. *Safety Compliance*

      Safety compliance berhubungan dengan tingkatan dimana perangkat lunak dapat memenuhi standar keamanan terhadap penggunaannya.

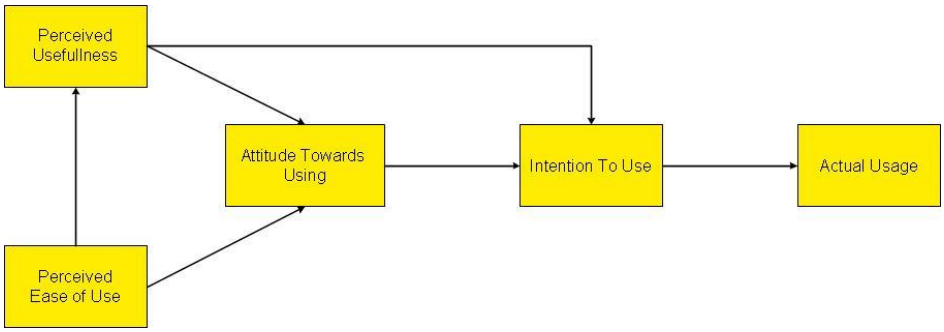
II.5 Model Penerimaan

Model penerimaan merupakan salah satu faktor apakah perangkat lunak yang dibangun dapat diterima oleh para penggunanya atau tidak. Terdapat beberapa teori mengenai model penerimaan ini, diantaranya:

1. *Technology Acceptance Model* (TAM)

Teori ini mengacu pada teori TRA dan telah dirancang untuk memprediksi penerimaan teknologi informasi dan penggunaannya dalam pekerjaan sehari-hari. Tidak seperti TRA, hasil konseptual dari TAM tidak termasuk pembangunan sikap untuk menjelaskan nilai yang lebih baik. TAM2 tidak termasuk TAM dengan memasukkan norma subjektif sebagai prediksi tambahan dari nilai pada kasus

pengaturan wajib. TAM secara luas diterapkan untuk beragam teknologi dan penggunanya [6]. Dalam TAM terdapat 5 bagian utama yang menjadi faktor penerimaan suatu perangkat lunak oleh pengguna, seperti yang terdapat pada Gambar II-III:



Gambar II-III Model Penerimaan TAM (*Technology Acceptance Model*)

1. *Perceived ease of use*

   Perceived ease of use menjelaskan sejauh mana seseorang yakin bahwa menggunakan suatu teknologi akan mempermudah pekerjaannya.

2. *Perceived usefulness*

   Perceived usefulness berhubungan dengan sejauh mana seseorang percaya bahwa dengan menggunakan suatu teknologi akan meningkatkan kinerjanya.

3. *Attitude toward using*

   Attitude toward using berhubungan dengan sikap seseorang untuk menggunakan suatu teknologi yang mereka masih anggap baru, apakah akan ada penerimaan atau penolakan.

4. *Behavioral intention to use*

   Behavioral intention to use menjelaskan tentang perilaku seseorang saat menggunakan teknologi yang mereka anggap baru.

5. *Actual system usage*

   Actual system usage diukur dengan jumlah waktu yang digunakan untuk berinteraksi dengan teknologi dan frekuensi penggunaan suatu teknologi oleh pengguna.

2. *Unified Theory of Acceptance an Use of Technology* (UTAUT2)

Model UTAUT2 diperkenalkan oleh Vankatesh pada tahun 2003 yang merupakan penggabungan dari delapan teori yang ada mengenai *Theory Acceptance Model*. Beberapa variabel yang dapat mempengaruhi penerimaan sebuah perangkat lunak pada model UTAUT2 ini diantaranya [6] [3]:

1. *Performance Expectancy*

   Variabel ini menjelaskan terkait keyakinan seorang pengguna bahwa dengan menggunakan sebuah perangkat lunak akan membantu meningkatkan kinerja dalam pekerjaan mereka.

2. *Effort Expectancy*

   Variabel ini menjelaskan tentang tingkat kemudahan penggunaan suatu perangkat lunak oleh seorang pengguna.

3. *Social Influence*

   Variabel ini menjelaskan dimana seorang pengguna menggunakan sebuah perangkat lunak karena pengaruh orang-orang yang ada disekitarnya.

4. *Facilitating Conditions*

   Variabel ini menjelaskan tingkat dimana seorang pengguna yakin bahwa lingkungan tempatnya bekerja menggunakan perangkat lunak tertentu.

5. *Behavioral Intention*

   Variabel ini berkaitan dengan kesiapan seorang pengguna untuk menggunakan sebuah perangkat lunak yang mereka masih anggap baru.

6. *Use behavior*

   Variabel ini berhubungan dengan perilaku pengguna saat menggunakan suatu perangkat lunak.

II.6 Metode Pengukuran Perangkat Lunak

Proses pengukuran merupakan Untuk melakukan pengukuran faktor kualitas dari sebuah perangkat lunak dapat menggunakan rumus sebagai berikut [7]:

$$Fa = w1c1 + w2c2 + \cdots + wncn \quad \text{(i)}$$

Dimana:

Fa adalah nilai dari total dari faktor a

wi adalah bobot untuk setiap kriteria i

ci adalah nilai untuk kriteria i

Untuk tahapan pengukurannya sendiri dapat menggunakan cara sebagai berikut:

1. Tentukan kriteria yang akan digunakan untuk mengukur suatu faktor
2. Tentukan bobot (w) dari setiap kriteria ($0 \leq w \leq 1$)
3. Tentukan skala dari nilai kriteria (misalnya $0 \leq$ nilai kriteria ≤ 10)
4. Berikan nilai pada setiap kriteria
5. Hitung nilai total dengan menggunakan rumus (i)

II.7 Metode Pengolahan Kuesioner

Kuesioner atau biasa disebut dengan angket merupakan sekumpulan pertanyaan yang diajukan kepada responden dengan tujuan dan maksud tertentu untuk memperoleh data, sehingga data tersebut dapat digunakan atau diolah kembali. Untuk mengolah data dari kuesioner tersebut dapat menggunakan berbagai macam metode pengolahan data, seperti dengan menggunakan skala *likert* dan skala *guttman*.

1. Skala *Likert*

Skala *likert* pertama kali dikembangkan oleh Rensis Likert pada tahun 1932 dalam pengukuran sikap masyarakat. Pengukuran ini dilakukan dengan pemberian skala pada masing-masing jawaban yang telah diajukan berdasarkan tujuan dari pertanyaan atau pernyataan yang diajukan. Salah satu contoh dalam pemberian skala terhadap pertanyaan yang berbentuk pernyataan yang biasa diajukan adalah dengan memberikan jawaban Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), dan Sangat Tidak Setuju (STS). Untuk pembuatan kuesioner dengan menggunakan skala *likert* ini sendiri dapat dibagi ke dalam dua jenis pertanyaan atau pernyataan, yakni yang bersifat positif dan yang bersifat negatif. Dalam penilaiannya sendiri dapat dilihat seperti contoh pada Tabel II-III [8].

Tabel II-III Penilaian Dengan Skala Likert

Jenis Pertanyaan/Pernyataan	Bobot Penilaian				
	SS	S	N	TS	STS
Positif	5	4	3	2	1

Negatif	1	2	3	4	5

2. Skala *Guttman*

Skala *guttman* dikembangkan oleh Louis Guttman. Skala ini biasanya digunakan untuk mendapatkan jawaban yang tegas, contohnya "Ya" atau "Tidak", "Pernah" atau "Tidak Pernah", dan lain-lain. Terkadang untuk memperoleh jawaban dari responden secara memuaskan, setiap jawaban tersebut dibagi lagi ke dalam beberapa skala penilaian, contohnya untuk memberikan jawaba "Ya" dibagi lagi di dalamnya seberapa besar responden menjawab "Ya" dengan sungguh-sungguh dan seberapa besar jawaban "Ya" untuk yang masih ragu-ragu, begitu pun untuk jawaban "Tidak". Untuk penilaiannya sendiri dapat menggunakan pemberian skala nilai seperti pada skala *likert* [8]. Berikut ini merupakan contoh pemberian penilaian dengan menggunakan skala *guttman*.

Tabel II-IV Penilaian Dengan Skala Guttman

Jenis Pertanyaan	**Bobot Penilaian**			
	Ya		**Tidak**	
	Ya (Sungguh-sungguh)	**Ya (Ragu-ragu)**	**Tidak (Ragu-ragu)**	**Tidak (Sungguh-sungguh)**
No. pertanyaan	4	3	2	1

Dalam perhitungan pengolahan data kuesioner ini setiap jawaban yang diperoleh dari responden kemudian dikalikan dengan masing-masing bobot yang telah ditentukan. Kemudian dari total seluruh bobot jawaban yang telah dihitung dibagi dengan total keseluruhan responden yang menjawab untuk mendapatkan nilai rata-rata. Berdasarkan nilai dari rata-rata tersebut maka diperoleh jawaban untuk ditarik sebuah kesimpulan. Berikut ini merupakan rumus dari perhitungan yang dapat digunakan untuk mengolah data kuesioner [8].

$$\bar{x} = \frac{\sum Total}{n}$$

Keterangan:

$\bar{x}$ = rata-rata

$\sum Total$ = jumlah total seluruh nilai setelah dikalikan dengan bobot

n = total responden

Dimana, apabila hasil dari $\bar{x} \geq 3$ maka jawaban dari responden tersebut bersifat positif dan apabila hasil dari $\bar{x} < 3$ maka jawaban dari responden tersebut bersifat negatif.