BAB II

LANDASAN TEORI

II.1 Definisi Pengukuran Perangkat Lunak

Pengukuran merupakan dasar dari setiap disiplin rekayasa dan berlaku juga dalam perekayasaan perangkat lunak. Untuk mengevaluasi performa suatu system atau proses diperlukan suatu mekanisme untuk mengamati dan menentukan tingkat efisiensinya. Melalui pengukuran, maka akan diperoleh tingkat pencapaian di dalam perangkat lunak

Pengukuran menurut IEEE adalah ukuran kualitas dari tingkat dimana sebuah sustem, komponen atau proses memiliki atribut tertentu. Sedangkan mengukur *(measure)* adalah mengindikasikan kuantitastif dari luasan, jumlah, dimensi dan kapasitas.

Untuk setiap pengukuran yang dilakukan dibutuhkan tersedianya suatu ukuran kuantitatif yang disebut metrik. Istilah ukuran, pengukuran dan metrik sering digunakan secara bergantian. Metrik berdasarkan istilah rekayasa perangkat lunak didefinisikan sebagai sebuah ukuran kuantitatif yang dimiliki oleh suatu system, komponen atau proses tertentu dengan *attribute-atribute* yang diberikan. Oleh karena itu, untuk selanjutnya akan digunakan istilah metrik untuk menyebutkan pengukuran dalam pengukuran perangkat lunak [4].

II.2 Software Quality

Menurut Agarwal dkk *Software Quality* merupakan sebagai kesesuaian terhadap kebutuhan performa dan fungsionalitas, standar pengembangan yang terdokumentasi, serta karakteristik implisit dari sebuah perangkat lunak yang dikembangkan secara professional.

Secara umum, definisi Software Quality yang disebutkan oleh adalah *as effective software process applied in a manner that creates a useful product that provides measurable value for those who produce it and those who use it*.

Proses dalam pembuatan sebuah barang dimana kita harus memastikan apakah barang tersebut sudah sesuai yang diharapkan atau belum, pengembangan perangkat lunak atau software juga menuntut hal yang sama. Metode yang dipakai dalam menganalisis kualitas perangkat lunak tersebut tentu saja berbeda dibandingkan dengan metode yang digunakan di pabrik-pabrik misalnya.

Pengujian adalah proses mengeksekusi program secara intensif untuk menemukan kesalahan-kesalahan. Pengujian tidak hanya untuk mendapatkan program yang benar, namun juga memastikan bahwa program tersebut bebas dari kesalahan-kesalahan untuk segala kondisi (Kristanto, 2003). Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan spesifikasi, desain dan pengkodean (Pressman, 1997) [5].

II.3 Domain Perangkat Lunak

Domain perangkat lunak merupakan kategori dari setiap jenis perangkat lunak yang ada. Terdapat tuju kategori mengenai jenis domain perangkat lunak ini sendiri, diantaranya sebagai berikut:

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.4 Perangkat Lunak IDE

IDE (Integrated Development Environment) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak,

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform *(platform-independent)* [6].

- Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows,Linux, Solaris, AIX, HP-UX dan Juga Mac OS X.
- Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta untuk pengembangannya. Semenjak itu konsursium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

Pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.Sampai saat sekarang ini Eclipse sudah mencapai versi 3.6 yang diberinama Helios

II.5 Model Kualitas Perangkat Lunak

Dalam keilmuan 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 model kualitas perangkat lunak yang dapat digunakan dalam penilaian kualitas perangkat lunak.

Model ISO-9126

Model ISO-9126 dikenalkan pertama kali pada tahun 1991 sebagai standarisai kualitas produk perangkat lunak [7]. Standarisasi ini dibuat karena banyaknya model kualitas yang ditawarkan sebagai faktor kualitas perangkat lunak. Dalam dokumen pertama model ISO-9129 terdiri dari empat bagian model kualitas untuk sebuah produk perangkat lunak, di antaranya [7]:

1. Model kualitas
2. Metrik eksternal
3. Mertik internal
4. Kualitas dalam menggunakan metrik

Bagian pertama dari 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.1

Tabel II-1 Faktor Kualitas Internal dan Eksternal

External and Internal Quality	
Faktor	**Sub-Faktor**
1. Functionality	*a. Suitability* *b. Accuracy* *c. Interoperability* *d. Securyty* *e. Functionality Compliance*
2. Reliability	*a. Maturity* *b. Fault Tolerance* *c. Recoverability* *d. Reliabilitu Comliance*
3. Usability	*a. Understandability* *b. Learnability* *c. Operability* *d. Attractiveness* *e. Usability Compliance*
4. Efficiency	*a. Time Behavior* *b. Resource Utilization* *c. Efficiency Compliance*

External and Internal Quality	
Faktor	**Sub-Faktor**
5. Maintainability	*a. Analyzability* *b. Changeability* *c. Modification Stability* *d. Testability* *e. Maintainability Compliance*
6. Pertability	*a. Adaptability* *b. Installability* *c. Co-existence* *d. Replaceability* *e. Portability Comliance*

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

1. *Functionality*

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

   a. Suitability

   Suitability berhubungan dengan tingkat kemampuan dan kelayakan dari sebuah perangkat lunak untuk dapat menyediakan fungsionalitas untuk kebutuhan yang spesifik.

   b. Accuracy

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

   c. Imteroperability

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

   d. Security

   Security berhubungna dengan keamanan yang dimiliki oleh sebuah perangkat lunak. Keamanan yang dimaksud dapat berupa pemberian hak akses kepada penggunanya.

e. *Functional Comliance*

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

2. *Reliability*

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

a. *Maturity*

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

b. *Fault Tolerance*

Faul 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 Comliance*

Reliability comliance merupakan tingkat dimana perangkat lunak dapat memenuhi standar kesalahan yang dimiliki oleh perangkat lunak lain sejenis.

3. *Usability*

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

a. *Understandability*

Understandability 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*

Attractiveness menggambarkan tentang bagaimana sebuah perangkat lunak dapat menarik perhatian bagi penggunanya.

e. *Usability Compliance*

Usability Compliance berhubungan dengan kesesuaian antara kegunaan perangkat lunak dengan standar yang digunakan oleh perangkat lunak sejenis lainnya.

4. *Efficiency*

Efficiency berhubungan dengan efisiensi dari seberapa besar sumber daya yang digunakan oleh sebuah perangkat lunak.

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. *Reource-utilisation*

Reource-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 tingkat dimana perangkat lunak memenuhi standar yang berhubungan dengan efisiensi kinerja perangkat lunak.

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*

      Stability 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*

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

6. *Portablity*

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

a. *Adaptability*

Adaptability 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 sebagai berikut:

1. *Effectiveness*

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

2. *Productivity*

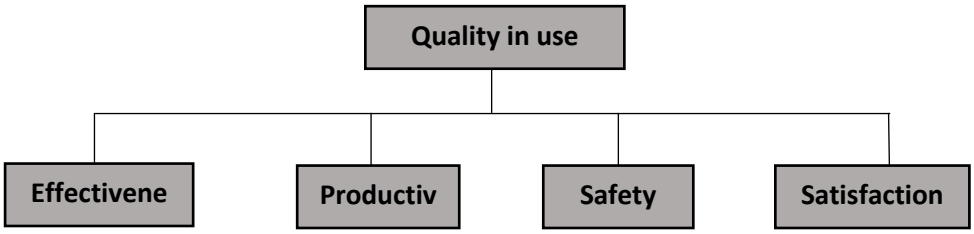
   Productivity merupakan upaya perangkat lunak dalam menghindari kelebihan penggunaan sumber daya untuk mencapai tujuan pengguna.

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-1 Model Quality In Use pada ISO-9126

II.6 Metrik ISO 9126

Implementasi metrik pada perangkat lunak berkaiatan erat dengan estimasi usaha dan biaya kegiatan proyek perangkat lunak. Produktivitas pada sistem dapat diukur dengan menghitung jumlah satuan yang dihasilkan dan membagi nilai ini dengan orang-jam yang yang dibutuhkan untuk menghasilkanya. Estimasi produktivitas biasanya berdasar atas pengukuran beberapa atribut perangkat lunak dan membaginya dengan usaha total yang dibutuhkanya untuk pengembangan, ada dua jenis pengukuran yang telah dipakai:

ISO-9126 adalah ISO 9126-2 adalah representasi kualitas karakteristik dan sub karakteristik ISO 9126 selama fase testing dan operasional. ISO 9126 menyediakan metrik eksternal yang memiliki banyak karakteristik yaitu antara lain fungsionalitas, kehandalan, kebergunaan, efisiensi, maintabilitas, portabilitas. Masing-masing karakteristik memiliki sub karakteristik yang memperjelas makna karakteristik-karakteristik tersebut. Karakteristik efisiensi memiliki sub karakteristik perilaku waktu, utilisasi sumber daya, pemenuhan efisiensi

(*efficiency compliance*) dimana merupakan sub karakteristik yang akan diukur dalam penelitian ini. Berikut ini adalah tabel metrik metrik dalam ISO 9126. [8]

1 *Functionality Metrics*

Metrik fungsionalitas eksternal harus mampu mengukur atribut seperti yang terdapat pada fungsional dari setiap perangkat lunak. Setiap perangkat lunak dapat diamati dari perspektif sebagai berikut:

- Perbedaan antara hasil eksekusi aktual dan persyaratan kualitas spesifikasi, persyasatan kualitas spesifikasi fungsi biasanya digambarkan sebagai persyaratan fungsional.
- Kekurangan dapat dilihat pada saat pengoprasian penggunaan yang tidak disebutkan namun yang bersifat sebagai persyaratan dala spesifikasi,

a. Suitability Metrics

Metrik kesesuaian eksternal harus mampu mengukur atribut seperti terjadinya fungsi yang tidak sesuai atau terjadinya operasi yang tidak sesuai selama pengujian dan user pengoperasian sistem. Dapat dilihat pada Tabel II-2

Tabel II-2 *Suitability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Functional adequacy	X = 1-A/B A = Jumlah fungsi yang terdapat masalah dalam evaluasi B = Jumlah fungsi yang di evaluasi	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
Functional implementation completeness	X = 1-A/B A = Jumlah fungsi yang hilang terdeteksi dalam evaluasi B = Jumlah fungsi yang dijelaskan dalam spesifikasi kebutuhan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Functional implementation coverage	X = 1-A/B A = Jumlah yang dilakasanakan dengan benar atau fungsi yang hilang dalam evaluasi B = jumlah fungsi yang dijelaskan dalam spesifikasi kebutuhan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
Functional specification stability (volatility)	X = 1-A/B A = Jumlah fungsi perubahan setelah memasuki operasi dimulai B = Jumlah fungsi yang dijelaskan dalam spesifikasi kebutuhan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

b. Accuracy Metrics

Metrik akurasi eksternal harus mampu mengukur atribut seperti frekuensi pengguna dalam menghadapi terjadinya hal-hal yang tidak akurat. Dapat dilihat pada Tabel II-3

Tabel II-3 *Accuracy Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Accuracy to expectation	X = A/T A = Jumlah kasus yang dihadapi oleh pengguna dengan perbedaan terhadap hasil yang diharapkan T = Waktu operasi	0 <= X Semakin dekat ke 0 semakin baik	*X= Count/Time* *A= Count* *T= Time*
Computational Accuracy	X = A/T A = Jumlah perhitungan akurasi yang dihadapi oleh pengguna T = Waktu operasi	0 <= X Semakin dekat ke 0 semakin baik	*X= Count/Time* *A= Count* *T= Time*
Precision	X = A/T A = Jumlah hasil yang dihadapi oleh pengguna dengan tingkat presisi yang berbeda dari yang dibutuhkan T = Waktu operasi	0 <= X Semakin dekat ke 0 semakin baik	*X= Count/Time* *A= Count* *T= Time*

c. Interoperability Metrics

Metrik interoperabilitas eksternal harus mampu mengukur atribut seperti jumlah fungsi atau kejadian yang kurang baik yang melibatkan data dan perintah, yang dengan mudah antara produk perangkat lunak dan sistem lainya, produk perangkat lunak lain, atau peralatan yang berhubungan. Dapat dilihat pada Tabel II-4

Tabel II-4 *Interoperability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Data exchangeability (Data format based)	X = A/B A = Jumlah format yang data yang disetujui untuk ditukar dengan perangkat lunak lain atau sistem pengujian bursa data B = Total jumlah format data yang akan diperlukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
Data exchangeability (User's success attempt based)	X = 1-A/B A = Jumlah kasus dimana pengguna gagal untuk pertukanaran data dengan perangkat lunak atau sistem lain B = Jumlah kasus di mana pengguna upaya untuk pertukaran data	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

d. Security Metrics

Metrik keamanan eksternal harus mampu mengukur atribut seperti jumlah fungsi berhubungan dengan masalah keamanan. Dapat dilihat pada Tabel II-5

Tabel II-5 *Security Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Access auditability	X = A/B A = Jumlah "pengguna akses ke sistem data" tercatat dalam database sejarah akses B = Jumlah "pengguna akses ke sistem dan data" dilakukan selama evaluasi	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
Access	X = A/B	0 <= X <= 1	*X=*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
controllability	A = Jumlah yang terdeteksi berbagai jenis operasi ilegal B = Jumlah jenis operasi ilegal seperti dalam spesifikasi	Semakin dekat ke 1.0 semakin baik	*Count/Count* *A= Count* *B= Count*
Data corruption prevention	X = 1 – A/N A = Jumlah kali bahwa peristiwa korupsi data terbesar yang terjadi N = Jumlah uji coba menyebabkan acara korupsi data	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *N= Count*

e. Functionality Compliance Metrics

Sebuah fungsi kepututusan metrik eksternal harus mampu mengukur atribut seperti jumlah fungsi dengan akurat, atau kejadian masalh keputusan, yang merupakan produk software gagal untuk mengetahui standar, konvesi, kontrak atau persyaratan lainya. Dapat dilihat pada Tabel II-6

Tabel II-6 *Functionality Compliance Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Functional compliance	X = 1 - A/B A = Jumlah item kepatuhan fungsi tertentu yang belum dilakasankan selama pengujian B = Total jumlah item kepatuhan fungsi tertentu	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
Interface standard compliance	X = A/B A = Jumlah interface diterapkan dengan benar sebagaimana ditentukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
	B = Total jumlah interface yang membutuhkan pemenuhan		

2 *Reliability Metrics*

Metrik kehandalan eksternal harus mampu mengukur atribut yang berhubungan dengan perilaku dari sistem dengan perangkat lunak merupakan

bagian selama pengujian eksekusi untuk menunjukan sejauh mana keandalan perangkat lunak dalam selama operasi, sistem dan perangkat lunak yang tidak dibedakan dari satu sama lain untuk semua kebanyakan kasus.

a. Maturity Metrics

Matrik ini harusmampu mengukur atribut seperti kebebasan software, kegagalan disebabkan oleh kesalahan yang ada dalam perangkat lunak. Dapat dilihat pada Tabel II-7

Tabel II-7 *Maturity Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Estimated latent fault density	X = {ABS(A1 – A2)}/B (X:perkiraan yang kepadatan kesalahan laten) ABS ()= Absolut A1 = Jumlah kesalahan laten memprediksi dalam produk perangkat lunak A2 = Jumlah sebenarnya terprediksi kesalahan. B = Produk Ukuran	0 <= X Hal ini tergantung pada tahap pengujian. Pada tahap selanjutnya lebih kecil lebih baik.	*X= Count/Size* *A1= Count* *A2= Count* *B = Size*
Failure density against test cases	X = A1/A2 A1 = Jumlah kegagalan terdeteksi A2 = Jumlah kasus uji yang dilakukan	0 <= X Hal ini tergantung pada tahap pengujian. Pada tahap selanjutnya lebih kecil lebih baik.	*X, Y= Count/Size* *A1= Count* *A2= Count* *B= Size*
Fault density	X = A/B A = Jumlah kesalahan yang terdeteksi B = Ukuran Produk	0 <= X Hal ini tergantung pada tahap pengujian. Pada tahap selanjutnya lebih kecil lebih baik.	*X = Count/Size* *A = Count* *B = Size*
Fault removal	a).X = A1/A2 A1= Jumlah kesalahan dikoreksi A2= Jumlah Sebenarnya terdeteksi kesalahan. b).Y = A1/A3 A= Jumlah kesalahan laten memprediksi dalam produk perangkat lunak.	0 <= X <= 1 Semakin dekat 1.0 semakin lebih baik sebagai kesalahan yang lebih sedikit 0 <= Y Semakin dekat 1.0 semakin lebih baik sebagai kesalahan yang lebih sedikit	*X= Count/Count* *Y= Count/Count* *A1=Count* *A2=Count* *A3=Count*
Mean time between failures (MTBF)	a).X = T1/A b).Y = T2 / A T1=Waktu operasi T2=Jumlah interval waktu yang berturut-turuk mengalami kegagalan	0 <= X, Y Semakin lama semakin baik, seperti waktu yang lebih lama dapat diharapkan anatara kegagalan	*X =Time / Count* *Y =Time /Count* *A = Count* *T1 = Time*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
	A=Jumlah sebenarnya terdeteksi kegagalan,(kegagalan terjadi selama waktu operasi).		*T2 = Time*
Test coverage (Specified operation scenario testing coverage)	X = A / B A=Jumlah kasus uji bener-bener dilakukan mewakili skenario operasi selama pengujiaan. B=Junlah kasus uji yang akan dilakukan untuk menutupi kebutuhan	0 <= X <= 1 Semakin mendekati 1.0 semakin lebih baik	*X= Count/Count* *A= Count* *B= Count*
Test maturity	X = A / B A=Jumlah kasus uji lulus selama pengujian atau operasi B=Jumlah kasus uji yang akan dilakukan untuk menutupi kebutuhan	0 <= X <= 1 Semakin mendekati 1.0 semakin lebih baik	*X= Count/Count* *A= Count* *B= Count*

b. Fault Tolerance Metrics

Sebuah kesalahan toleransi metrik eksternal harus berhubungan dengan kamampuan software, mempertahankan tingkat kinerja yang ditentukan dalam kasus kesalahan operasi atau pelanggaran antarmuka yang ditentukan. Dapat dilihat pada Tabel II-8

Tabel II-8 *Fault Tolerance Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Breakdown avoidance	X = 1-A/B A = Jumlah kerusakan B = Jumlah kegagalan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
Failure avoidance	X = A/B A = Jumlah kejadian kegagalan kritis dan serius terhadap kasus uji dari pola kesalahan B=Jumlah kasus uji dieksekusi dari pola kesalahan.	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
Incorrect operation avoidance	X = A/B A = Jumlah kejadian kegagalan kritis dan serius terhadap kasus uji dari pola kesalahan B=Jumlah kasus uji	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
	dieksekusi dari pola kesalahan.		

c. Recoverability Metrics

Metrik pemulihan eksternal harus mampu mengukur atribut seperti perangkat lunak dengan sistem yang mampu membangun kembali tingkat yang memadai atas kinerja dan memulihkan data yang terkena dampak langsung dalam kasus kegagalan. Dapat dilihat pada Tabel II-9

Tabel II-9 *Recoverability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Availability	a).X={To/(To+Tr)} To=Waktu untuk operasi Tr=Wakto untuk meperbaiki A1=Total kasus yang tersedia pengguna software pengguna berhasil ketika pengguna mencoba untuk menggunakan b).Y=A1/A2 A2=Jumlah kasus upaya pengguna untuk	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik, karena pengguna dapat menggunakan perangkat lunak untuk lebih banyak waktu 0 <= Y <= 1 Semakin dekat ke 1.0 semakin baik	*X= Time/Time* *To = Time* *Tr = Time* *Y= Count/Count* *A1= Count* *A2= Count*
	menggunakan perangkat lunak selama waktu pengamatan		
Mean down time	X=T/N T= Jumlah down time N=Jumlah kerusakan yang diamati kasus terburuk atau distribusi down time harus diukur.	0 <X Semakin dekat ke 1.0 semakin baik	*X= Size/Count* *T= Time* *N=Count*
Mean recovery time	X=Sum(T)/N T=Waktu untuk pemulihan sistem perangkat lunak pada setiap kesempatan B=Jumlah kasus yang diamati sistem software masuk recovery	0 <X Semakin dekat ke 1.0 semakin baik	*X= Time/Count* *T= Time* *B= Count*
Restartability	X=A/B A=Jumlah restart yang bertemu untuk waktu yang dibutuhkan selama pengujian atau operasi pengguna B=Total jumlah restart	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik, karena pengguna dapat me restart dengn mudah.	*X=Count/Count* *A=Count* *B=Count*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
	selama pengujian atau operasi pengguna		
Restorability	X=A/B A=Jumlah kasus restorasi berhasil dilakukan B=Jumlah kasus restorasi diuji sesuai kebutuhan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik, karena produk lebih mampu untuk mengembangkan dalam kasus tertentu.	*X=Count/Count* *A=Count* *B=Count*
Restore effectiveness	X=A/B A=Jumlah kasus berhasil dipublikasikan memenuhi terget mengembalikan waktu B=Jumlah kasus yang dilakukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik, karena proses restorasi di produk lebih efektif.	*X=Count/Count* *A=Count* *B=Count*

d. Reliability Compliance Metrics

Sebuah kehandalan kepatuhan metrik eksternal harus mampu mengukur atribut seperti jumlah fungsi dengan, atau kejadian masalah keputusan, di mana produk perangkat lunak gagal untuk mematuhi standar, konvensi atau peraturan yang berkaitan dengan kehandalan suatu perangkat lunak. Dapat dilihat pada Tabel II-10

Tabel II-10 *Reliability Compliance Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Reliability compliance	X = 1-A/B A = Jumlah item kepuasan keandalan ditentukan yang belum dilaksanakan selama pengujian B = Total jumlah item kepatuhan keandalan ditentukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

3 *Usability Metrics*

Metrik ini mengukur sejauh mana perangkat lunak dapat dipahami, dipelajari,dioperasikan, menarik dan sesuai dengan peraturan kegunaan dan pedoman.

Banyak metrik *usability* eksternal diuji oleh pengguan mencoba untuk menggunakan fungsi yang terdapat pada perangkat lunak. Hasil akan

dipengaruhi oleh kemampuan pengguna dan karakteristik perangkat lunak. Kareana software dievaluasi dijalankan dalam kondisi eksplisit ditentukan oleh sampel pengguna yang mewakili kolompok pengguna yang di diindentifikasikan. Untuk hasil yang bisa diandalkan sampel pengguna setidaknya harus memnguasai perangkat lunak tersebut, meskipun informasi yang berguna dapat diperoleh dari kelompok-kelompok yang lebih kecil. Pengguna harus melakukan tes tanpa petunjuk atau bantuan dari luar.

a. Understandability Metrics

Pengguna harus dapat memilih produk perangkat lunak, yang cocok untuk digunakan. Sebuah metrik harus dapat menilai apakah penguna baru dapat memahami, apakah software tersebut cocok, dan bagaimana hal itu dapat digunakan untuk tugas-tugas tertentu. Dapat dilihat pada Tabel II-11

Tabel II-11 *Understandability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Completeness of description	X = A/B A =Jumlah fungsi (jenis fungsi) dipahami B = Total jumlah (jenis fungsi)	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.	*X= Count/Count* *A= Count* *B= Count*
Demonstration accessibility	X = A/B A =Jumlah demonstrasi/tutorial yaang pengguan berhasil mengaksesnya B =Jumlah demonstrasi/tutorial yang tersedia	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.	*X= Count/Count* *A= Count* *B= Count*
Demonstration accessibility in use	X = A/B A =Jumlah kasus di mana pengguna berhasil melihat demonstrasi ketika pengguna mencoba untuk meliha demonstrasi B =Jumlah kasus di mana pengguna mencoba untuk melihat demonstrasi	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.	*X= Count/Count* *A= Count* *B= Count*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
	selama periode pengamatan		
Demonstration effectiveness	X = A/B A =Jumlah fungsi yang di operasikan berhasil B =Jumlah demonstrasi/tutorial yang di akses	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.	*X= Count/Count* *A= Count* *B= Count*
Evident functions	X = A/B A =Jumlah fungsi (jenis fungsi) yang didentifikasi oleh pengguna B =Total jumlah fungsi yang sebenarnya	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.	*X= Count/Count* *A= Count* *B= Count*
Function understand-ability	X = A/B A =Jumlah fungsi antarmuka yang tujuanya dengan benar dijalankan oleh pengguna B =Jumlah fungsi yang tersedia dari antarmuka	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.	*X= Count/Count* *A= Count* *B= Count*
Understandable input and output	X = A/B A =Jumlah input dan output data item yang pengguna berhasil mengerti B =Jumlah item data input dan output yang tersedia dari antarmuka.	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.	*X= Count/Count* *A= Count* *B= Count*

b. Learnability Metrics

Sebuah *learnability* metrik eksternal harus dapat menilai seberapa lama pengguna mengambil untuk belajar sebagai menggunakan fungsi tertentu, dan efektivitas sistem bantuan dan dokumentasi. Dapat dilihat pada Tabel II-12

Tabel II-12 *Learnability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Ease of function learning	T= Berarti waktu yang dibutuhkan untuk pembelajaran menggunakan fungsi dengan benar	0 < T Semakin pendek semakin baik	*T = Time*
Ease of learning to perform a task	T=Jumlah waktu operasi pengguna sampau waktu dicapai	0 < T Semakin pendek semakin a=baik	*T = Time*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
in use	untuk melakukan tugas tertentu dalam waktu singkat		
Effectiveness of the user documentation and/or help system	X = A/B A = Jumlah tugas yang berhasil diselesaikan setelah mengakses bantuan online atau dokumentasi B =Jumlah tugas diuji	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

c. Operability Metrics

Sebuah *operability* metrik eksternal harus dapat menilai apakah pengguna dapat mengoprasikan dan mengendalikan perangkat lunak. Metrik pengoperasian dapat dikategorikan oleh prinsip-prinsip dialog dalam ISO-92126-10. Dapat dilihat pada Tabel II-13

Tabel II-13 *Operability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Operational consistency in use	a).X = 1-A/B A = Jumlah waktu yang dibutuhkan B = Jumlah waktu yang ditentukan b).Y=N/UOT N= Jumlah pesan atau fungsi yang menemukan data yang tidak konsisten dengan akspektasi pengguna yang UOT=Pengguna waktu operasi selama sistem berjalan.	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik 0 <= Y Semakin dekat ke 0.0 semakin baik	*X= Count/Count* *A= Count* *B= Count* *UOT= Count/Count* *N= Count* *Y= Count*

d. Attractiveness Metrics

Eksternal metrik *attractiveness* metrik harus bisa menilai penamplan perangkat lunak. Dan akan dipengaruhi oleh fakto-faktor seperti desain layar dan warna. Hal ini sangat penting untuk produk konsumen. Dapat dilihat pada Tabel II-14

Tabel II-14 *Attractivanes Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Attractive interaction	Kuesioner untuk menilai daya tarik dari antarmuka pengguan, setelah pengguna mengunakan.	Tergantung pada metode kuesioner scoringnya	*Count*
Interface appearance customisability	X = A/B A = Jumlah elemen antarmuka disesuaikan dalam penampilan untuk kepuasan pengguna B = Jumlah antarmuka pengguna yang ingin menyesuaikan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

e. Usability Compliance Metrics

Sebuash *Usability compliance* metrik eksternal harus mampu menilai kepatuhan terhadap standar, konvensi, panduan gaya atau peraturan yang berkaitan dengan kegunaan. Dapat dilihat pada Tabel II-15

Tabel II-15 *Usability Compliance Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Usability compliance	X = 1-A/B A = Jumlah item kepuasan kegunan terentu yang belum dilaksanakan selama pengujian B = Total jumlah item kepuasan kegunaan tertentu	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

4 *Efficiency Metrics*

Efficiency metrik eksternal harus mampu mengukur atribut seperti konsumsi waktu dan sumber daya perilaku pemanfaatan sistem komputer termasuk perangkat lunak selama pengujian operasi.

Disarankan bahwa maksimal dan distribusi waktu yang diuji untuk banyak kasus pengujian atau operasi, karena ukuran yang dipengaruhi kuat dan tergantung pada kondisi penggunaan, seperti beban pengolahan data, frekuensi penggunaan, jumlah yang mengandung situs dan sebagainya. Oleh

karena itu, metrik efisiensi dapat mencakup rasio nilai aktual yang di ukur dengan fluktual kesalahan dengan nilai yang dirancang dengan memungkinkan berbagai kesalahan, yang dubutuhkan oleh spesifikasi.

a Time Behaviour Metrics

Time behaviour metrik eksternal harus mampu mengukur atribut seperti perilaku saat sistem komputer termasuk perangkat lunak selama pengujian atau pengoperasian. Dapat dilihat pada Tabel II-16

Tabel II-16 *Time Behaviour Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Response time	T = (Waktu untuk mendapatkan hasil)-(saat masuk perintah selesai)	0 < T Semakin cepat semakin baik.	*T = Time*
Response time (Mean time to response)	Tmean = (Ti) ∑/ N, (untuk i = 1 sampai N) TXmean = diperlukan waktu respon rata-rata Ti = waktu respon untuk evaluasi ke-i (shot) N = jumlah evaluasi (tembakan sampel)	0 <= X Semakin dekat ke 1.0 dan kurang dari 1.0 lebih baik	*Tmean= Time* *TXmean= Time* *Ti= Time* *N= Count* *X= Time/ Time*
Response time (Worst case response time ratio)	X = Tmax / Rmax Tmax = MAX (Ti) (untuk i = 1 sampai N) Rmax = diperlukan waktu respons maksimum MAX (Ti) = waktu respons maksimum antara evaluasi N = jumlah evaluasi (tembakan sampel) Ti = waktu respon untuk evaluasi ke-i (shot)	0 <= X Semakin ke 1 dan kurang dari 1 akan lebih baik	*Tmax= Time* *Rmax=Time* *Ti= Time* *N= Count* *X= Time/ Time*

b Resource Utilisation Metrics

Resource Utilisation metrik eksternal harus mampu mengukur atribut seperti sumber daya sistem perangkat lunak selama pengujian atau pengoperasian. Dapat dilihat pada Tabel II-17

Tabel II-17 *Resource Utilisation Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
I/O devices utilisation	X = A/B A = Waktu I/O perangkat diduduki B = Ditentukan waktu yang dirancang untuk menempati	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*A= Time* *B= Time* *X= Time/Time*
I/O loading limits	X = Amax/Rmax Amax = MAX (Ai), (untuk i = 1 sampai N) Rmax = maksimum yang diperlukan I / O pesan MAX (Ai) = Jumlah maksimum pesan I / O dari 1 untuk evaluasi ke-i. N = jumlah evaluasi.	0 <= X Semakin kecil semakin baik	*Amax = Count* *Rmax = Count* *Ai = Count* *N= Count* *X = Count/ Count*
I/O related errors	X = A/T A= Jumlah pesan peringatan atau kegagalan sistem T= Penggunaan waktu operasi selam pengamatan pengguna	0 <= X Semakin kecil semakin baik	*A = Count* *T = Time* *X = Count/ Time*
Mean I/O fulfillment ratio	X = Amean/Rmean Amean = ∑ (Ai) / N Rmean = diperlukan rata jumlah pesan I /O Ai = jumlah pesan kesalahan I / O untuk i th evaluasi N = jumlah evaluasi	0 <= X Semakin kecil semakin baik	*Amean = Count* *Rmean = Count* *Ai = Count* *N= Count* *X = Count/ Count*
User waiting time of I/O devices utilisation	T= Waktu yang dihabiskan untu menugguakhir dari perangkat I/O operasi	0 < T Semakin kecil semakin baik	*T = Time*

c Efficiency Compliance Metrics

Efficiency Compliance metrik eksternal harus mampu mengukur atribut seperti jumlah fungsi, atau kejadian masalah kepatuhan, yang merupakan produk software gagal untuk mematuhi standar, konvensi atau peraturan yang berkaitan dengan efisiensi. Dapat dilihat pada Tabel II-18

Tabel II-18 *Efficiency Compliance Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Efficiency Compliance	X = 1-A/B (X:Rasio item kepatuhan berkaitan denga efisiensi) A = Jumlah efisiensi item kepatuahan ditentukan yang belum dilaksanakan selama pengujian B = Total jumlah item kepatuhan efisiensi dintentukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

5 *Maintainability Metrics*

Metrik *maintainability* eksternal harus mampu mengukur atribut seperti pengolahan atau perawatan perangkat lunak, pengguna atau sistem termasuk perangkat lunak, ketika perangkat lunak dipertahankan atau diubah selama pengujian atau pemeliharaan.

a Analysability Metrics

Sebuah *analisability* metrik eksternal harus mampu mengukur atribut seperti pemeliharaan atau usaha atau menghabisan sumber daya ketika mencoba untuk mengetahui kekurangan atau penyebab kegagalan atau bagian untuk pengguna. Dapat dilihat pada Tabel II-19

Tabel II-19 *Analysability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Audit trail capability	X = A/B A = Jumlah data yang sebenarnya direkam selama operasi	0 <= X Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
	B = Jumlah data direncanakan akan cukup direkam untuk memantau status perangkat lunak selama operasi		
Diagnostic function support	X = 1-A/B A = Jumlah kegagalan yang pemelihara dapat diagnos e (menggunakan fungsi diagnostik) untuk memahami penyebab -	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
	kapal hubungan efek B = Total jumlah kegagalan terdaftar		

b Changeability Metrics

Metrik *changeability* aksternal harus mampu mengukur atribut seperti upaya pemelihara atau pengguna dengan mengukur perilaku pengelola, pengguna atau sistem termasuk perangkat lunak ketika mencoba untuk menerapkan modifikasi tertentu. Dapat dilihat pada Tabel II-20

Tabel II-20 *Changeability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Change cycle efficiency	Average Time : Tav = Sum(Tu) / N Tu= Trc – Tsn Tsn = Waktu di mana pengguna selesai untuk mengirim permintaan untuk pemeliharaan untuk pemasok dengan laporan masalah Trc = Waktu di mana pengguna menerima rilis versi revisi (atau laporan status) N = Jumlah versi revisi	0<Tav Semakin pendek lebih baik kecuali dari jumlah versi revisi itu besar	*Tu= Time* *Trc,* *Tsn = Time* *N= Count* *Tav= Time*
Change implementation elapse time	Average Time : Tav = Sum(Tm) / N Tm=Tout – Tin Tout = Waktu di mana penyebab kegagalan dihapus dengan mengubah perangkat	0<Tav Semakin pendek lebih baik kecuali dari jumlah versi revisi itu besar	*Tm= Time* *Tin,* *Tout = Time* *Tav= Time*
	lunak (atau status dilaporkan kembali ke pengguna) Tin = Waktu di mana penyebab kegagalan yang menemukan N = Jumlah kegagalan terdaftar dan dihapus		
Modification complexity	T = Sum (A / B) / N A = Kerja dihabiskan	0 < T Semakin dekat ke 1.0	*A= Time* *B= Size*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
	untuk mengubah B = Ukuran perubahan software N = Jumlah perubahan	semakin baik	*N= Count* *T= Time*
Parameterised modifiability	X=1- A / B A = Jumlah kasus yang pemelihara gagal untuk mengubah perangkat lunak dengan menggunakan parameter B = Jumlah kasus yang maintainer mencoba untuk mengubah perangkat lunak dengan menggunakan parameter	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*A= Count* *B= Count* *X= Count/ Count*
Software change control capability	X= A / B A = Jumlah data log perubahan benar-benar dicatat B = Jumlah data perubahan log rencananya akan direkam cukup untuk melacak perubahan perangkat lunak	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*A= Count* *B= Count* *X= Count/ Count*

c Stability Metrics

Sebuah *Stability Metrics* eksternal harus mampu mengukur atribut yang berhubungan dengan perilaku yang tak terdiga dari sistem perangkat lunak ketika perangkat lunak diuji atau dioperasikan. Dapat dilihat pada Tabel II-21

Tabel II-21 *Stability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Change success ratio	X = Na/Ta Y={(Na/Ta)/(Nb/Nb)} Na = Jumlah kasus yang pengguna pertemuan kegagalan selama operasi setelah software berubah Nb = Jumlah kasus yang pengguna pertemuan kegagalan selama operasi sebelum perangkat lunak berubah Waktu ta = Operasi selama periode observasi ditentukan setelah perangkat lunak berubah Tb waktu = Operasi selama periode pengamatan ditentukan sebelum perangkat lunak berubah	0 <= X,Y Semakin dekat ke 1.0 semakin baik	*Na, Nb= Count* *Ta,Tb=Time* *X= Count/Time* *Y=[(Count/Time) / (Count/Time)]*
Modification impact localisation (Emerging failure after change)	X=A/N A = Jumlah kegagalan muncul setelah kegagalan diselesaikan oleh perubahan selama periode tertentu N = Jumlah kegagalan diselesaikan	0<=X Semakin dekat ke 0 semakin baik	*X= Count/Court* *A= Count* *N= Count*

d Testability Metrics

Sebuah *Testability Metrics* eksternal harus mampu mengukur atribut seperti upaya pemelihara atau pengguna dengan mengukur perilaku pengelola, pengguna atau sistem termasuk perangkat lunak ketika mencoba untuk diuji. Dapat dilihat pada Tabel II-22

Tabel II-22 *Testability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Availability of built-in test function	X = A/B A = Jumlah kasus dimana pengelola dapat	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
	mengunakan sesuia buit-in fungsi tes B =Jumlah kasus peluang uji		

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Re-test efficiency	X = Sum(T)/N T = Waktu yang dihabiskan untuk menguji untuk memastikan apakah melaporkan kegagalan itu diselesaikan atau tidak N = Jumlah kegagalan diselesaikan	0 <= X Semakin dekat ke 1.0 semakin baik	*X= Time/Count* *T= Time* *N= Count*
Test restartability	X = A/B A = Jumlah kasus di mana pengelola dapat menghentikan sebentar dan restart melaksanakan uji coba pada titik-titik yang diinginkan untuk memeriksa langkah demi langkah B = Jumlah kasus jeda melaksanakan uji coba	0 <= X Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

e Maintainability Compliance Metrics

Sebuah *Maintainability Compliance Metrics* eksternal harus mampu mengukur atribut seperti fungsi atau kejadian masalah kepuasan, dimana produk peranglat lunak gagal untuk memenuhi standar yang diperlukan, konvensi atau peraturan yang berkaitan dengan pemeliharaan. Dapat dilihat pada Tabel II-23

Tabel II-23 *Maintainability Compliance Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Maintainability compliance	X = 1-A/B A = Jumlah item kepatuhan rawatan ditentukan yang belum dilaksanakan selama pengujian	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
	B = Total jumlah item kepatuhan rawatan ditentukan		

6 *Portability Metrics*

Portability Metrics ekternal harus mampu mengukru atribut seperti operasi atau sistem selama pengoperasian berlangsung.

a Adaptability Metrics

Adaptability Metrics harus mampu mengukur atribut seperti perilaku sistem atau pengguna yang mencoba untuk beradaptasi pada perangkat lunak untuk lingkungan tertentu yang berbeda, ketika pengguna harus menrapkan prosedur dari sebelumnya yang disediakan oleh perangkat lunak. Dapat dilihat pada Tabel II-24

Tabel II-24 *Adaptability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Adaptability of data structures	X = A/B A = Jumlah data yang beroperasi dan tetapi tidak diamati karena operasi tidak lengkap disebabkan oleh keterbatasan adaptasi B = Jumlah data yang diharapkan dapat beroperasi dalam lingkungan yang perangkat lunak disesuaikan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
Hardware environmental adaptability	X = 1-A/B A = Jumlah fungsi ional operas yang tugas tidak selesai atau tidak cukup menghasilkan untuk memenuhi memadai tingkat s selama co mbined pengujian operasi dengan hardware lingkungan B = Total jumlah fungsi yang diuji	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

b Installability Metrics

Installability Metrics eksternal harus mampu mengukur atribut seperti perilaku sistem atau pengguna yang mencoba untuk mengintal perangkat lunak dalam lingkungann tertentu. Dapat dilihat pada Tabel II-25

Tabel II-25 *Installability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Ease of installation	X = A/B A = Jumlah kasus yang pengguna berhasil di chang ing menginstal operasi untuk /	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
	kenyamanan nya B = Total jumlah kasus yang pengguna mencoba untuk mengubah menginstal operasi untuk / kenyamanan nya		
Ease of Setup Re-try	X = 1-A/B A = Jumlah kasus di mana pengguna gagal di coba selama set - up operasi B = Total jumlah kasus di mana pengguna upaya untuk kembali mencoba setup selama set - up operasi	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

c Co-existence Metrics

Co-existence Metrics eksternal harus mampu mengukur atribut seperti perilaku sistem atau pengguna yang mencoba untuk menggunakan perangkat lunak dengan perangkat lunak independen lain dalam lingkungan yang sama berbagi sumber daya umum. Dapat dilihat pada Tabel II-26

Tabel II-26 *Co-existence Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Available co-existence	X = A/T A = Jumlah hambatan atau kegagalan tak terduga yang pengguna pertemuan selama operasi bersamaan dengan software lain T = Waktu durasi bersamaan operasi perangkat lunak lain	0 <= X Semakin dekat ke 1.0 semakin baik	*X= Count/Time* *A= Count* *T= Time*

d Replaceability Metrics

Replaceability Metrics eksternal harus mampu mengukur atribut seperti perilaku sistem atau pengguna yang mencoba untuk menggunakan perangkat lunak di tempat software ditentukan lain dalam lingkungan perangkat lunak. Dapat dilihat pada Tabel II-27

Tabel II-27 *Replaceability Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Continued use of data	X = A/B A = jumlah data yang digunakan dalam perangkat lunak lain yang akan diganti dan menegaskan bahwa mereka mampu terus menerus digunakan B = jumlah data yang digunakan dalam perangkat lunak lain untuk diganti dan rencananya akan terus digunakan kembali	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
Function inclusiveness	X = A/B A = jumlah fungsi yang menghasilkan hasil yang sama seperti sebelumnya diproduksi dan di mana perubahan tidak akan en diperlukan B = jumlah fungsi diuji yang mirip dengan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*
	fungsi yang disediakan oleh perangkat lunak lain yang akan diganti		
User support functional consistency	X = 1-A1/A2 A = Jumlah fungsi baru yang ditemukan pengguna unacceptabl y konsisten dengan ekspektasi pengguna B = Jumlah fungsi baru	0 <= X Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A1= Count* *A2= Count*

e Portability Compliance Metrics

Portability Compliance Metrics eksternal harus mampu mengukur atribut seperti jumlah fungsi dengan, atau kejadian masalah kepatuhan, di mana produk perangkat lunak gagal untuk mematuhi standar yang diperlukan, konvensi atau peraturan yang berkaitan dengan portabilitas. Dapat dilihat pada Tabel II-28

Tabel II-28 *Portability Compliance Metrics*

Nama Metrik	Pengukuran	Interaksi Nilai Yang Diukur	Jenis Ukuran
Portability compliance	X = 1-A/B A = Jumlah portabilitas item kepatuhan ditentukan yang belum dilaksanakan selama pengujian B = Total jumlah portabilitas item kepatuhan ditentukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik	*X= Count/Count* *A= Count* *B= Count*

Dengan rentang penilaianan, 0 - 0,33 Tidak Bagus/Rendah, 0,34 - 0,66 Bagus/Menengah, 0,67 - 1 Sangat Bagus/Tinggi [8].

II.7 Rata-rata atau Rata-rata Hitung

Untuk keperluan ini, dan perhitungan selanjutnya, akan di gunakan simbul-simbul. Nilai-nilai *data kuantitatif* akan dinyatakan dengan x_1, x_2....., x_n**,** dan apabila delam kumpulan data itu terdapat n buah nilai. Simbul n juga akan dipakai untuk menyatakan ukuran sampel, yaitu banyak data atau obyek yang diteliti dalam sampel. Simbol N dipakai untuk menyatakan ukuran populasi, yakni banyak anggota terdapat dalam populasi.

Jika ada lima nilai ujian dari lima orang mahasiswa untuk mata kuliah statistika berbentuk : 70, 69, 45, 80, dan 56, makan dalam simbul ditulis: $x_1 = 70$, $x_2 = 69$, $x_3 = 45$, $x_4 = 80$, dan $x_5 = 56$.

Rata-rata, atau lengkapnya rata-rata hitung, untuk data kuantitatif yang terdapat dalam sebuah sampel dihitung dengan jalan membagi jumlah nilai data oleh banyak data.

Simbul rata-rata untuk sampel ialah x (baca: eks garis) sedangkan rata-rata untuk populasi dipakai simbul μ adalah parameter untuk menyatakan rata-rata. Rumus untuk rata-rata x adalah [9]:

$$\chi = \frac{x_1 + x_2 + \cdots + x_n}{n} \text{ atau } \chi = \frac{\sum_{i=1}^{n} x_i}{n}$$

Atau lebih sederhana lagi ditulis :

$$x = \frac{\sum x_i}{\mathrm{n}} \ldots\ldots\ldots (1)$$

Keterangan :

$\sum x_1$ = Nilai

n = Jumlah Nilai

II.8 *Rank Order Centroid* (ROC)

ROC didasarkan pada tingkat kepentingan atau prioritas dari kriteria. Menurut Jeffreys dan Ceckfield dalam Afiefah Rahma (2013), teknik ROC memberikan bobot pada setiap kriteria sesuai dengan ranking yang dinilai berdasarkan tingkat prioritas. Biasanya dibentuk dengan peryataan “Kriteia 1 lebih penting dari kriteria 2, yang lebih penting dari kriteria 3” dan seterusnya hingga kriteria ke n, ditulis. Untuk menentukan bobotnya, diberikan aturan yang sama yaitu dimana merupakan bobot untuk kriteia. Atau dapat dijelaskan sebagai berikut :[10].

Jika

$C_{r1} \geq C_{r2} \geq C_{r3} \geq \ldots\ldots \geq C_m$

Maka

$W_1 \geq W_2 \geq W_3 \geq \ldots\ldots \geq W_m$

Selanjutnya, jika k merupakan banyaknya kriteria, makan

$$W1 = \frac{1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{\mathrm{k}}}{\mathrm{k}}$$

$$W2 = \frac{0 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}}{k}$$

$$W3 = \frac{0 + 0 + \frac{1}{3} + \cdots + \frac{1}{k}}{k}$$

$$Wk = \frac{1 + 0 + 0 + \cdots + \frac{1}{k}}{k}$$

Secara umum pembobotan ROC dapat dirumuskan sebagai berituk :

$$\mathrm{Wk} = 1/k \sum_{i=1}^{k} \left(\frac{1}{i}\right) \quad \text{.....(2).}$$

Dimana :

- Wk = Normalisasi rasio perkiraan skala bobot tujuan
- i = Total jumlah tujuan
- k= Ranking dari i tujuan

II.9 Metode Pembobotan *Simple Additive Weighting* (SAW)

Simple Additive Weigthing (SAW) sering juga dikenal istilah metode penjumlahan terbobot, merupakan sebuah metode yang cukup terkenal dan sering digunakan dalam *Multiple Attribute Decision Making* (MADM). Setiap alternative dikalikan dengan atribut untuk memperoleh nilai. Proses perhitungan dapat dinyatakan dalam rumus (1) [11].

$$Pi = \sum_{j=1}^{M} w_j . m_{ij} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Keterangan

Pi = Nilai SAW

j = Batas bawah perhitungan

M = Batas atas perhitungan

wj = Bobot kriteria j

mij = Nilai alternatif i pada kriteria j

Banyak pendapat yang menyatakan bahwa metode SAW hanya bisa digunakan ketika kriteria keputusan yang ada dinyatakan dalam satu bentuk. Namun jika setiapelemen yang ada dinormalisasi, maka metode SAW dapat digunakan pada semua jenis kriteria maupun alternatif. Dalam kasus tersebut, maka bentuk perhitungan akan menjadi seperti rumus (2) [11]:

$$Pi = \sum_{j=1}^{M} w_j.(m_{ij})normal \dots\dots\dots\dots\dots\dots\dots\dots (3)$$

Keterangan :

Pi = Nilai SAW

j = Batas bawah perhitungan

M = Batas atas perhitungan

wj = Bobot kriteria j

(mij)normal = Nilai kriteria dari alternatif pada baris i kolom j yang telah dinormalisasi

Ketika (mij) normal mewakili nilai dari mij yang telah dinormalisasi, alternatifdengan nilai Pi tertinggi akan dipilih sebagai alternatif terbaik. Kriteria yang ada dapat berupa kriteria keuntungan maupun kerugian. Kriteria keuntungan merupakan kriteria dimana semakin besar nilainya maka nilai alternatif akan menjadi semakin baik. Dan kriteria kerugian adalah kriteria dimana semakin kecil nilainya, maka nilai alternatif akan menjadi semakin baik. Jika kriteria merupakan kriteria keuntungan, maka normalisasi dihitung dengan menggunakan rumus (3) [11]:

$$(m_{ij})normal = \frac{(m_{ij})K}{(m_{ij})L} \ldots\ldots\ldots\ldots (4)$$

Keterangan :

(mij) normal = Nilai data baris i kolom j yang telah dinormalisasi

(mij) K = Nilai alternatif i pada kriteria j

(mij) L = Nilai kriteria tertinggi dari alternatif

Namun jika kriteria yang akan dihitung merupakan kriteria kerugian (nilai terendah merupakan nilai yang lebih baik), maka normalisasi dihitung dengan menggunakan rumus (4) [11]:

$$(m_{ij})normal = \frac{(m_{ij})L}{(m_{ij})K} \ldots\ldots\ldots\ldots (5)$$

Keterangan :

(mij) normal = Nilai data baris i kolom j yang telah ternormalisasi

(mij) L = Nilai kriteria terendah dari alternatif

(mij) K = Nilai alternatif i pada kriteria j