BAB II
LANDASAN TEORI

II.1. Kualitas Perangkat Lunak

Pada pengertian yang paling umum kualitas perangkat lunak dapat didefinisikan sebagai suatu proses yang efektif diterapkan dalam arti kata proses perangkat lunak yang menyediakan nilai yang dapat diukur untuk mereka yang menghasilkan dan untuk mereka yang menghasilkannya.

Beberapa organisasi, seperti ISO dan IEEE mencoba untuk membuat standar kualitas software yang mengkombinasikan model dan mengaitkan faktor dan sub-faktor. Beberapa penelitian juga mengusulkan metrik perangkat lunak sebagai alat untuk mengukur *source code* program, arsitektur dan kinerja dari perangkat lunak. Sampai saat ini proses penilaian perangkat luak tetap menjadi masalah terbuka dengan banyak model karena belum jelas dan belum adanya kesepakatan hubungan antara *software quality model* serta hubungan antara model dan metrik.

Pengukuran kualitas tidak hanya berfokus pada hasil akhir tetapi juga proses pada pengukuran kualitasnya sehingga meyakinkan penilaian untuk kualitas dengan metode yang digunakan. Hal ini akan mengarah ke penekanan yang berbeda dari pada berfokus pada kualitas akhir.

II.2. Domain Perangkat Lunak

Domain perangkat lunak merupakan kategori dari setiap jenis perangkat yang ada. Menurut Roger S. Pressman, terdapat tujuh kategori mengenai jenis domain perangkat lunak yaitu[3]:

1. *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* adalah sebuah program yang berdiri sendiri dan digunakan untuk mengatasi kebutuhan bisnis yang spesifik.

3. *Engineering/scientific software* merupakan 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* 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* merupakan 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* 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* merupakan 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 *Web Server*

Server HTTP *Apache* atau Server Web/WWW *Apache* adalah server web yang dapat dijalankan dibanyak system operasi seperti *Unix, BSD, Linux, Windows, Novell Netware* serta platform lainnya yang berguna melayani dan memfungsikan situs web. Protocol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP. *Apache* memiliki fitur-fitur canggih seperti

pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna yang berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah.

Pengguna biasanya melalui aplikasi pengguna seperti peramban web, meminta layanan atas berkas ataupun halaman web yang terdapat pada sebuah server web kemudian server sebagai manajer layanan tersebut akan merespon balik dengan mengirimkan halaman dan berkas-berkas pendukung yang dibutuhkan, atau menolak permintaan tersebut jika halaman yang diminta tidak ada. Saat ini umumnya *Web Server* telah dilengkapi pula dengan mesin penerjemah bahasa yang memungkinkan *Web Server* menyediakan layanan situs web dinamis dengan memanfaatkan pustaka tambahan seperti PHP dan ASP.

Pemanfaatan *Web Server* saat ini tidak terbatas hanya untuk publikasi situs web, pada prakteknya *Web Server* banyak pula digunakan dalam perangkat-perangkat keras lain seperti printer, router, kamera web yang menyediakan layanan HTTP dalam jaringan local yang ditujukan untuk menyediakan perangkat manajemen serta mempermudah peninjauan atas perangkat keras tersebut.

II.4. Model Faktor Kualitas Perangkat Lunak

Kualitas perangkat lunak dapat di nilai melalui ukuran-ukuran dan metode-metode tertentu, serta melalui pengujian-pengujian perangkat lunak. Salah satu tolak ukur kualitas perangkat lunak adalah ISO-9126, yang dibuat oleh *International Organization for Standardization* (ISO) dan *International Electrotechnical Commission* (IEC). ISO-9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk perangkat lunak. Standar ISO-9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer[4].

Berikut adalah model ISO-9126 untuk faktor kualitas *Eksternal* dan *Quality In Use*, yaitu:

II.4.1 Model ISO-9126 *Eksternal*

Dalam model ini dapat digunakan untuk mengukur kualitas suatu perangkat lunak dengan mengukur dari perilaku sistem yang merupakan dari bagiannya. Metrik eksternal hanya dapat digunakan selama tahap pengujian proses siklus hidup dan selama tahap operasional. Berikut penjelesan beberapa faktor dan sub-faktornya yang dari ISO-9126.

II.4.1.1 *Functionality*

Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu. Setiap perangkat lunak dapat diamati sebagai berikut:

II.4.1.1.1 *Suitability*

Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna.

Tabel II-1 *Suitability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
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
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
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

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Functional specification stability (volatility)	X = 1-A/B A = Jumlah fungsi perubahan setelah memasuki operasi B = Jumlah fungsi yang dijelaskan dalam spesifikasi kebutuhan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.1.2 *Accuracy*

Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.

Tabel II-2 *Accuracy*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Accuracy to expectation	X = A/T A = jumlam kasus yang dihadapi oleh pengguna dengan perbedaan terhadap hasil yang diharapkan T = Waktu operasi	0 <= X Semakin dekat ke 0 semakin baik
Computational Accuracy	X = A/T A = Jumlah perhitungan akurasi yang dihadapi oleh pengguna T = Waktu operasi	0 <= X Semakin dekat ke 0 semakin baik
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

II.4.1.1.3 *Interoperability*

Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih dalam sistem tertentu.

Tabel II.3 *Interoperability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Data exchangeability (Data format based)	X = A/B A = jumlam format data yang disetujui untuk ditukar dengan perangkat lunak lain atau sistem pengujian data B = Total jumlah format data yang akan diperlukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik
Data exchangeability (User's success attempt based)	X = 1-A/B A = Jumlah kasus dimana pengguna gagal dalam pengiriman data dengan perangkat lunak atau sistem lain B = Jumlah kasus di mana pengguna berupaya dalam pengiriman data	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.1.4 *Security*

Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup (*hacker*) maupun otorisasi dalam modifikasi data.

Tabel II-4 *Security*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Access auditability	X = A/B A = Jumlah pengguna yang mengakses ke sistem data B = Jumlah "pengguna akses ke sistem dan data" dilakukan selama evaluasi	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik
Access controllability	X = A/B A = Jumlah yang terdeteksi sebagai jenis operasi ilegal B = Jumlah jenis operasi ilegal seperti dalam spesifikasi	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Data corruption prevention	X = 1 – A/N A = Jumlah kehilangan data terbesar yang terjadi N = Jumlah uji coba yang menyebabkan kehilangan data	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.1.5 *Functionality Compliance*

Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku.

Tabel II-5 *Functionality Compliance*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Functional Compliance	X = 1 - A/B A = Jumlah item yang mengikuti fungsi tertentu yang belum dilakasanakan selama pengujian B = Total jumlah item kepatuhan fungsi tertentu	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik
Interface standard Compliance	X = A/B A = Jumlah interface yang diterapkan dengan benar B = Total jumlah interface yang membutuhkan pemenuhan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.2 *Reliability*

Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.

II.4.1.2.1 *Maturity*

Kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak.

Tabel II-6 *Maturity*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Estimated latent fault density	X = {ABS(A1 – A2)}/B (X : perkiraan kesalahan) ABS ()= Absolut A1 = Jumlah kesalahan dalam produk perangkat lunak A2 = Jumlah yang sebenarnya terprediksi dalam kesalahan. B = Produk Ukuran	0 <= X Hal ini tergantung pada tahap pengujian. Pada tahap selanjutnya lebih kecil lebih baik.
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.
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.
Fault removal	a).X = A1/A2 A1= Jumlah kesalahan dikoreksi A2= Jumlah yang sebenarnya terdeteksi dalam kesalahan. b).Y = A1/A3 A= Jumlah kesalahan 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

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Mean time between failures (MTBF)	a).X = T1/A b).Y = T2 / A T1=Waktu operasi T2=Jumlah waktu interval yang berturut-turut dalam mengalami kegagalan A=Jumlah sebenarnya yang terdeteksi dalam kegagalan,(kegagalan terjadi selama waktu operasi).	0 <= X, Y Semakin lama semakin baik, seperti waktu yang lebih lama dapat diharapkan anatara kegagalan
Test coverage (Specified operation scenario testing coverage)	X = A / B A=Jumlah uji kasus selama pengujiaan. B=Junlah uji kasus yang akan dilakukan untuk menutupi kebutuhan	0 <= X <= 1 Semakin mendekati 1.0 semakin lebih baik
Test maturity	X = A / B A=Jumlah uji kasus lulus selama pengujian B=Jumlah kasus uji yang akan dilakukan untuk menutupi kebutuhan	0 <= X <= 1 Semakin mendekati 1.0 semakin lebih baik

II.4.1.2.2 *Fault Tolerance*

Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak.

Tabel II-7 *Fault Tolerance*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Breakdown avoidance	X = 1-A/B A = Jumlah kerusakan B = Jumlah kegagalan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik
Failure avoidance	X = A/B A = Jumlah kegagalan terhadap uji kasus B=Jumlah kasus uji dieksekusi dari pola kesalahan.	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Incorrect operation avoidance	X = A/B A = Jumlah kegagalan terhadap uji kasus dari kesalahan B=Jumlah kasus uji dieksekusi dari kesalahan.	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.2.3 *Recoverability*

Kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan.

Tabel II-8 *Recoverability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Availability	a).X={To/(To+Tr)} To=Waktu untuk operasi Tr=Wakto untuk memperbaiki A1=Total kasus yang tersedia ketika pengguna mencoba untuk menggunakannya b).Y=A1/A2 A2=Jumlah kasus pengguna untuk menggunakan perangkat lunak selama waktu pengamatan	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
Mean down time	X=T/N T= Jumlah down time N=Jumlah kerusakan yang diamati kasus terburuk atau distribusi down time	0 <X Semakin dekat ke 1.0 semakin baik
Mean recovery time	X=Sum(T)/N T=Waktu untuk pemulihan sistem perangkat lunak B=Jumlah kasus yang diamati sistem software masuk recovery	0 <X Semakin dekat ke 1.0 semakin baik

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Restartability	X=A/B A=Jumlah restart yang bertemu dibutuhkan selama pengujian B=Total jumlah restart selama pengujian	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik, karena pengguna dapat me restart dengn mudah.
Restorability	X=A/B A=Jumlah kasus restorasi yang berhasil dilakukan B=Jumlah kasus restorasi diuji sesuai dengan kebutuhan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik, karena produk lebih mampu untuk mengembangkan dalam kasus tertentu.
Restore effectiveness	X=A/B A=Jumlah kasus yang berhasil dipublikasikan untuk mengembalikan waktu B=Jumlah kasus yang dilakukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik, karena proses restorasi di produk lebih efektif.

II.4.1.2.4 *Reliability Compliance*

Sebuah metrik *eksternal* yang mampu mengukur atribut seperti jumlah fungsi dengan kejadian masalah keputusan, di mana produk perangkat lunak gagal untuk mematuhi standar, konvensi atau peraturan yang berkaitan dengan kehandalan suatu perangkat lunak.

Tabel II-9 *Reliability Compliance*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
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

II.4.1.3 *Usability*

Kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu..

Hasil akan dipengaruhi oleh kemampuan pengguna dan karakteristik perangkat lunak. Karena 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.

II.4.1.3.1 *Understandability*

Kemampuan perangkat lunak dalam kemudahan untuk dipahami. Metrik ini dapat menilai apakah penguna baru dapat memahami, apakah software tersebut cocok, dan bagaimana hal itu dapat digunakan untuk tugas-tugas tertentu.

Tabel II-10 *Understandability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
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.
Demonstration accessibility	X = A/B A =Jumlah demonstrasi/tutorial yang berhasil B =Jumlah demonstrasi/tutorial yang tersedia	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.
Demonstration accessibility in use	X = A/B A =Jumlah kasus pengguna yang berhasil ketika pengguna mencoba demonstrasi B =Jumlah kasus pengguna mencoba untuk melihat demonstrasi selama periode pengamatan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.
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.

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
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.
Function understand-ability	X = A/B A =Jumlah fungsi antarmuka yang tujuanya benar B =Jumlah fungsi yang tersedia dari antarmuka	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.
Understandable input and output	X = A/B A =Jumlah input dan output data item yang pengguna yang berhasil dimengerti B =Jumlah item data input dan output yang tersedia dari antarmuka.	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik.

II.4.1.3.2 *Learnability*

Kemampuan perangkat lunak dalam kemudahan untuk dipelajari.

Tabel II-11 *Learnability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Ease of function learning	T= waktu yang dibutuhkan untuk menggunakan fungsi dengan benar	0 < T Semakin pendek semakin baik
Ease of learning to perform a task in use	T=Jumlah waktu operasi pengguna sampai waktu dicapai untuk melakukan tugas tertentu	0 < T Semakin pendek semakin a=baik
Effectiveness of the user documentation and/or help system	X = A/B A = Jumlah tugas yang berhasil diselesaikan B =Jumlah tugas diuji	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.3.3 *Operability*

Kemampuan perangkat lunak dalam kemudahan untuk dioperasikan.

Tabel II-12 *Operability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Error correction	T = Tc – Ts Tc = waktu menyelesaikan koreksi Ts = Waktu mulai koreksi	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.3.4 *Attractiveness*

Metrik *attractiveness* bisa menilai penampilan perangkat lunak. Dan akan dipengaruhi oleh fakto-faktor seperti desain layar dan warna. Hal ini sangat penting untuk produk konsumen.

Tabel II-13 *Attractivenes*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Attractive interaction	Kuesioner untuk menilai daya tarik dari antarmuka pengguan, setelah pengguna mengunakannya	Tergantung pada metode kuesioner scoringnya
Interface appearance customisability	X = A/B A = Jumlah elemen antarmuka B = Jumlah antarmuka pengguna yang ingin menyesuaikan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.3.5 *Usability Compliance*

Metrik *Usability Compliance* mampu menilai kepatuhan terhadap standar, konvensi, panduan gaya atau peraturan yang berkaitan dengan kegunaan.

Tabel II-14 *Usability Compliance*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Usability Compliance	X = 1-A/B A = Jumlah item kepuasan kegunaan terentu yang belum dilaksanakan selama pengujian B = Total jumlah item kepuasan kegunaan tertentu	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.4 *Efficiency*

Metrik *Efficiency* mampu mengukur atribut seperti penggunaan waktu dan sumber daya termasuk pada sistem komputer perangkat lunak selama pengujian.

Disarankan bahwa maksimal dan distribusi waktu yang diuji untuk banyak kasus pengujian atau operasi, karena ukuran yang kuat dipengaruhi tergantung pada kondisi penggunaan, seperti pengolahan data, frekuensi penggunaan, dan sebagainya. Oleh karena itu, metrik ini dapat mencakup rasio nilai aktual yang di ukur dengan fluktual kesalahan dengan nilai yang dirancang dengan memungkinkan berbagai kesalahan yang dibutuhkan secara spesifik..

II.4.1.4.1 *Time Behaviour*

Metrik *time behaviour* mampu mengukur atribut seperti perilaku pada sistem komputer dan termasuk juga perangkat lunak selama pengujian atau pengoperasian.

Tabel II-15 *Time Behaviour*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Response time	T = (Waktu untuk mendapatkan hasil)	0 < T Semakin cepat semakin baik.
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
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

II.4.1.4.2 *Resource Utilisation*

Kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan.

Tabel II-16 *Resource Utilisation*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
I/O devices utilisation	X = A/B A = Waktu I/O dalam perangkat B = Ditentukan waktu yang telah dirancang	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik
I/O loading limits	X = Amax/Rmax Amax = MAX (Ai), (untuk i = 1 sampai N) Rmax = maksimum pesan yang diperlukan I / O MAX (Ai) = Jumlah maksimum pesan I / O dari 1 untuk evaluasi ke-i. N = jumlah evaluasi.	0 <= X Semakin kecil semakin baik
I/O related errors	X = A/T A= Jumlah pesan peringatan atau kegagalan sistem T= Penggunaan waktu operasi selama pengamatan pengguna	0 <= X Semakin kecil semakin baik
Mean I/O fulfillment ratio	X = Amean/Rmean Amean = $\sum$ (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
User waiting time of I/O devices utilisation	T= Waktu yang dihabiskan untuk menuggu akhir dari perangkat I/O operasi	0 < T Semakin kecil semakin baik

II.4.1.4.3 *Efficiency Compliance*

Metrik *Efficiency Compliance* yang 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.

Tabel II-17 *Efficiency Compliance*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Efficiency Compliance	X = 1-A/B (X : item rasio yang berkaitan dengan efisiensi) A = Jumlah efisiensi yang belum dilaksanakan selama pengujian B = Total jumlah item efisiensi yang ditentukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.5 *Maintainability*

Metrik *maintainability* 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.

II.4.1.5.1 *Analysability*

Metrik yang mampu mengukur atribut seperti pemeliharaan usaha atau menghabisan sumber daya ketika mencoba untuk mengetahui kekurangan atau penyebab kegagalan atau bagian untuk pengguna.

Tabel II-18 *Analysability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Audit trail capability	X = A/B A = Jumlah data yang direkam selama operasi B = Jumlah data yang direncanakan untuk memantau status perangkat lunak selama operasi	0 <= X Semakin dekat ke 1.0 semakin baik
Diagnostic function support	X = 1-A/B A = Jumlah kegagalan yang dapat menyebabkan	0 <= X <= 1 Semakin dekat ke 1.0

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
	efek B = Total jumlah kegagalan terdaftar	semakin baik

II.4.1.5.2 *Changeability*

Metrik *changeability* mampu mengukur atribut seperti upaya pemelihara atau pengguna dengan mengukur perilaku pengelola, pengguna atau sistem termasuk perangkat lunak ketika mencoba untuk menerapkan modifikasi tertentu.

Tabel II-19 *Changeability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Change cycle efficiency	Average Time : Tav = Sum(Tu) / N Tu= Trc – Tsn Tsn = Waktu untuk mengirim permintaan dengan laporan masalah Trc = Waktu menerima laporan status n = Jumlah versi revisi	0<Tav Semakin pendek lebih baik kecuali dari jumlah versi revisi itu besar
Change implementation elapse time	Average Time : Tav = Sum(Tm) / N Tm=Tout – Tin Tout = Waktu penyebab kegagalan dengan mengubah perangkat lunak Tin = Waktu penyebab kegagalan yang menemukan N = Jumlah kegagalan terdaftar	0<Tav Semakin pendek lebih baik kecuali dari jumlah versi revisi itu besar
Modification complexity	T = Sum (A / B) / N A = Kerja dihabiskan untuk mengubah B = Ukuran perubahan software N = Jumlah perubahan	0 < T Semakin dekat ke 1.0 semakin baik
Parameterised modifiability	X=1- A / B A = Jumlah kasus gagal untuk mengubah perangkat lunak dengan menggunakan parameter B = Jumlah kasus untuk mengubah perangkat lunak dengan menggunakan parameter	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Software change control capability	X= A / B A = Jumlah data log perubahan B = Jumlah data perubahan log untuk melacak perubahan perangkat lunak	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.5.3 *Stability*

Sebuah metrik yang mampu mengukur atribut yang berhubungan dengan perilaku yang tak terdiga dari sistem perangkat lunak ketika perangkat lunak diuji atau dioperasikan.

Tabel II-20 *Stability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Change success ratio	X = Na/Ta Y={(Na/Ta)/(Nb/Nb)} Na = Jumlah kasus kegagalan selama operasi setelah software berubah Nb = Jumlah kasus 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
Modification impact localisation (Emerging failure after change)	X=A/N A = Jumlah kegagalan muncul selama periode tertentu N = Jumlah kegagalan diselesaikan	0<=X Semakin dekat ke 0 semakin baik

II.4.1.5.4 *Testability*

Sebuah metrik yang mampu mengukur atribut seperti upaya pemelihara atau pengguna dengan mengukur perilaku pengelola, pengguna atau sistem termasuk perangkat lunak ketika mencoba untuk di uji.

Tabel II-21 *Testability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Availability of built-in test function	X = A/B A = Jumlah kasus sesuai buit-in fungsi tes B =Jumlah kasus peluang uji	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik
Re-test efficiency	X = Sum(T)/N T = Waktu habis untuk memastikan kegagalan selesai atau tidak N = Jumlah kegagalan selesai	0 <= X Semakin dekat ke 1.0 semakin baik
Test restartability	X = A/B A = Jumlah kasus berhenti dan restart selama uji coba untuk memeriksa langkah demi langkah B = Jumlah kasus jeda melaksanakan uji coba	0 <= X Semakin dekat ke 1.0 semakin baik

II.4.1.5.5 *Maintainability Compliance*

Sebuah metrik yang mampu mengukur atribut seperti fungsi atau kejadian masalah kepuasan, dimana produk perangkat lunak gagal untuk memenuhi standar yang diperlukan, konvensi atau peraturan yang berkaitan dengan pemeliharaan.

Tabel II-22 *Maintainability compliance*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Maintainability compliance	X = 1-A/B A = Jumlah item kepatuhan yang belum dilaksanakan selama pengujian B = Total jumlah item kepatuhan rawatan ditentukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.6 *Portability*

Metrik *portability* dengan kemampuan mampu mengukur atribut seperti operasi atau sistem selama pengoperasian berlangsung.

II.4.1.6.1 *Adaptability*

Metrik yang 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.

Tabel II-23 *Adaptability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Adaptability of data structures	X = A/B A = Jumlah data yang beroperasi dan tidak lengkap yang disebabkan oleh keterbatasan adaptasi B = Jumlah data yang dapat beroperasi dalam lingkungan perangkat lunak yang disesuaikan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik
Hardware environmental adaptability	X = 1-A/B A = Jumlah fungsional operasi tidak selesai atau tidak cukup selama pengujian operasi dengan hardware B = Total jumlah fungsi yang diuji	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.1.6.2 *Installability*

Metrik yang mampu mengukur atribut seperti perilaku sistem atau pengguna yang mencoba untuk menginstal perangkat lunak dalam lingkungan tertentu.

Tabel II-24 *Installability*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Ease of installation	X = A/B A = Jumlah kasus pengguna untuk menginstal / kenyamanannya B = Total jumlah kasus pengguna dalam instalasi	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik
Ease of Setup Re-try	X = 1-A/B A = Jumlah kasus pengguna gagal selama set -	0 <= X <= 1 Semakin dekat ke

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
	up operasi B = Total jumlah kasus pengguna untuk kembali mencoba setup selama set - up operasi	1.0 semakin baik

II.4.1.6.3 *Co-Existence*

Metrik yang 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.

Tabel II-25 *Co-existence*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Available co-existence	X = A/T A = Jumlah hambatan atau kegagalan tak terduga selama operasi bersamaan dengan software lain T = Waktu durasi bersamaan operasi perangkat lunak lain	0 <= X Semakin dekat ke 1.0 semakin baik

II.4.1.6.4 *Replaceability*

Metrik yang mampu mengukur atribut seperti perilaku sistem atau pengguna yang mencoba untuk menggunakan perangkat lunak di tempat software ditentukan lain dalam lingkungan perangkat lunak.

Tabel II-26 *Replaceability*

Nama Metrik	Pengukuran	Interaksi nilai ukur
Continued use of data	X = A/B A = jumlah data yang digunakan dalam perangkat lunak lain yang akan diganti dan terus menerus digunakan B = jumlah data yang digunakan dalam perangkat lunak lain untuk diganti dan digunakan kembali	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

Nama Metrik	Pengukuran	Interaksi nilai ukur
Function inclusiveness	X = A/B A = jumlah fungsi yang menghasilkan hasil yang sama seperti sebelumnya dan di mana perubahan tidak akan diperlukan B = jumlah fungsi diuji yang mirip dengan fungsi yang disediakan oleh perangkat lunak lain yang akan diganti	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik
User support functional consistency	X = 1-A1/A2 A = Jumlah fungsi baru yang ditemukan pengguna dengan ekspektasi pengguna B = Jumlah fungsi baru	0 <= X Semakin dekat ke 1.0 semakin baik

II.4.1.6.5 *Portability Compliance*

Metrik yang 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.

Tabel II-27 *Portability Compliance*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Portability compliance	X = 1-A/B A = Jumlah portabilitas yang ditentukan dan belum dilaksanakan selama pengujian B = Total jumlah portabilitas yang ditentukan	0 <= X <= 1 Semakin dekat ke 1.0 semakin baik

II.4.2 Model ISO-9126 *Quality In Use*

Ukuran kualitas yang digunakan harus didasarkan pada data yang mencerminkan hasil pengguna berinteraksi dengan suatu perangkat lunak. Hal ini memungkinkan untuk mengumpulkan data tujuan tertentu. Selain itu terdapat pula model *quality in use* pada model ISO-9126 yang akan digunakan dalam penelitian ini yang merupakan salah satu model kualitas berupa pandangan pengguna

terhadap sebuah perangkat lunak. Berikut beberapa faktor yang terdapat dalam model *quality in use* diantaranya sebagai berikut :

II.4.2.1 *Effectiveness*

Menilai apakah tugas yang dilakukan oleh pengguna mencapai tujuan tertentu dengan akurasi dan kelengkapan dalam konteks tertentu yang digunakan.

Tabel II-28 *Effectiveness*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Task Effectiveness	M1 = $\| 1-\Sigma Ai \|_1$ Ai = nilai value proporsional masing-masing komponen yang hilang atau tidak benar dalam output tugas	0 <= M1 <= 1 Semakin dekat ke 1 lebih baik.
Task Completion	X = A / B A = jumlah tugas diselesaikan B = jumlah tugas berusaha	0 <= X <= 1 Semakin dekat ke 1 lebih baik.
Error Frequency	X = A / T A = jumlah kesalahan yang dibuat oleh pengguna T = waktu atau jumlah tugas	0 <= X Semakin dekat ke 0 yang lebih baik.

II.4.2.2 *Productivity*

Menilai sumber daya yang pengguna gunakan dalam konteks tertentu yang digunakan. Sumber daya yang paling umum adalah waktu untuk menyelesaikan tugas, meskipun sumber daya lainyang relevan dapat mencakup upaya pengguna.

Tabel II-29 *Productivity*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Task Time	X = Ta / T Ta = jumlah waktu yang digunakan untuk menyelesaikan tugas T = waktu atau jumlah tugas	0 <= X Semakin kecil semakin baik.
Task Efficiency	X = M1 / T M1 = efektivitas tugas T = tugas waktu	0 <= X Semakin besar lebih baik.

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Economic Productivity	X = M1 / C M1 = efektivitas tugas C = total biaya tugas	0 <= X Semakin besar lebih baik.
Productive Proportion	X = Ta / Tb Ta = waktu produktif = waktu tugas - membantu waktu - waktu error - waktu pencarian Tb = waktu tugas	0 <= X <= 1 Semakin dekat ke 1,0 lebih baik.
Relative User Efficiency	Efisiensi pengguna relatif X = A / B A= Efisiensi tugas atau Seorang pengguna biasa B = Efisiensi tugas atau ahli pengguna	0 <= X <= 1 Semakin dekat ke 1,0 lebih baik.

II.4.2.3 *Safety*

Menilai tingkat resiko terhadap pengguna, bisnis, software atau lingkungan dalam konteks tertentu yang digunakan. Ini termasuk kesehatan dan keselamatan baik pengguna dan mereka yang terkena dampak penggunaan serta konsekuensi fisik atau ekonomi yang tidak di inginkan.

Tabel II-30 *Safety*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
User health and safety	X = 1-A / B A = jumlah pengguna melaporkan RSI (cedera yang diakibatkan dari kegiatan yang dilakukan berulang-ulang) B = Jumlah pengguna	0 <= X <= 1 Semakin dekat ke 1 yang lebih baik.
Safety of people affected by use of the system	X = 1-A / B A = jumlah orang menaruh di hazard B = Jumlah orang yang berpotensi terkena dampak oleh sistem	0 <= X <= 1 Semakin dekat ke 1 yang lebih baik.
Economic damage	X = 1-A / B A = jumlah kejadian kerusakan ekonomi B = Jumlah situasi penggunaan	0 <= X <= 1 Semakin dekat ke 1 yang lebih baik.

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Software Damage	X = 1-A / B A = jumlah kejadian data hilang dalam perangkat lunak B = Jumlah situasi penggunaan	0 <= X <= 1 Semakin dekat ke 1 yang lebih baik.

II.4.2.4 *Satisfaction*

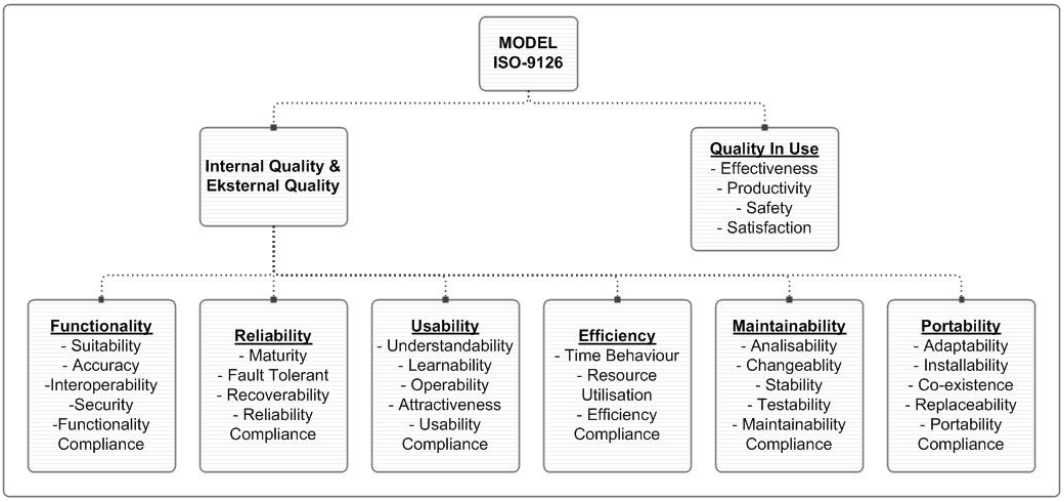
Menilai sikap pengguna terhadap penggunaan produk dalam konteks yang digunakan. Sebagai catatan kepuasan dipengaruhi oleh persepsi pengguna sifat dari produk perangkat lunak dan oleh persepsi pengguna dari efisiensi, produktivitas dan dalam penggunaan.

Tabel II-31 *Satisfaction*

Nama Metrik	Pengukuran	Interpretasi Nilai Ukur
Satisfaction Scale	X = A / B A = memproduksi kuesioner skala psikometrik B = rata populasi	0 < X yang lebih besar lebih baik
Satisfaction Questionnaire	X = Σ (Ai) / n Ai) = menanggapi pertanyaan n = jumlah tanggapan	Bandingkan dengan nilai sebelumnya, atau dengan rata-rata populasi
Discretionary Usage	X = A / B A = berapa kali fungsi perangkat lunak tertentu / aplikasi / sistem yang digunakan B = jumlah kali mereka dimaksudkan untuk digunakan	0 <= X <= 1 dekat ke 1 yang lebih baik.

Berdasarkan gambar 2 dibawah dalam model ISO-9126 juga terdapat internal quality yang dalam faktor dan subfaktor terdapat kesamaan dengan eksternal quality hanya saja terdapat perbedaan dari metrik dan tujuannya dari masing-masing subfaktor.

Gambar-2 Model ISO-9126



II.5 Pengukuran

Pengukuran perangkat lunak adalah faktor utama untuk mengetahui seberapa baik perangkat lunak yang akan dinilai berdasarkan metode yang digunakan. Pengukuran dilakaukan dengan cara mencari nilai kriteria perangkat lunak dan mencari bobot kepentingan berdasarkan metode-metode yang digunakan. Penilaian keseluruhan dapat di peroleh dari hasil nilai kriteria dan nilai bobot. Perhitungan bobot dan nilai kriteria bisa bervariasi tergantung dengan metode yang digunakan[5][6.]

II.6 Kuesioner

Kuesioner merupakan sebuah daftar pertanyaan diajukan kepada responden dengan tujuan dan maksud tertentu untuk memperoleh data, sehingga data tersebut dapat digunakan atau diolah kembali. Dalam penelitian ini kuesioner digunakan untuk mengambil data dari responden yang bertujuan untuk mendapatkan data berupa tanggapan atau pengalaman dari responden dalam menggunakan perangkat lunak *Web Server Apache.*

II.7 Rata-rata (Mean)

Rata-rata atau Mean merupakan ukuran statistik kecenderungan terpusat yang paling sering digunakan. Penghitungan rata-rata dilakukan dengan menjumlahkan

seluruh nilai data suatu kelompok sampel, kemudian dibagi dengan jumlah sampel tersebut. Jadi jika suatu kelompok sampel acak dengan jumlah sampel n, maka bisa dihitung rata-rata dari sampel tersebut dengan rumus sebagai berikut.

Untuk mencari nilai rata-rata digunakan rumus rata-rata (Mean) sebagai berikut [7] :

$$\bar{x} = \frac{\sum_{i=1}^{n} xi}{n} \qquad (2.1)$$

Dimana :

$\bar{x}$ = nilai rata-rata

xi = nilai sample ke-*i*

n = jumlah sample

II.8 Metode Pembobotan

Metode pembobotan *Rank Order Centroid* (ROC) didasarkan pada tingkat kepentingan atau prioritas dari kriteria. Menurut Jeffreys dan Cockfield (2013) teknik ROC memberikan bobot pada setiap kriteria sesuai dengan rangking yang dimulai berdasarkan tingkat prioritas. Untuk menentukan bobotnya diberikan aturan yaitu dimana merupakan bobot untuk kriteria sebagai berikut :

Jika

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

Maka

$$W_1 \geq W_2 \geq W_3 \geq \cdots \geq W_n$$

Selanjutnya, jika k merupakan banyak kriteria, maka

$$W_1 \frac{1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}}{\mathrm{k}}$$

$$W_2 \frac{0 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}}{\mathrm{k}}$$

$$W_3 \frac{0+0+\frac{1}{3}+\cdots+\frac{1}{k}}{\mathrm{k}}$$

$$W_k \frac{0+\cdots+0+\frac{1}{k}}{\mathrm{k}}$$

Secara umum pembobotan ROC dapat dirumuskan sebagai berikut.

$$Wk = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{1}{i}\right) \qquad (2.2)$$

II.9. Metode Keputusan

Metode *Weighted Summation Model* digunakan untuk mengubah optimasi *multi-objective* menjadi bentuk optimasi *single-objective*. Fungsi objective tunggal ini dibentuk dari penjumlahan tiap-tiap fungsi objective *ai* dikalikan dengan koefisien pembobot *wi*. Secara umum untuk metode dapat dinyatakan sebagai berikut [8] :

$$f(x) = \sum_{w=1}^{n} a_i w_i \qquad (2.3)$$

Dimana :

a = nilai kriteria

w = bobot kriteria

n = jumlah data