

Rancang Bangun Kakas Bantu Karakteristik Portabilitas Menggunakan Model Kualitas ISO/IEC 9126

Nama Mahasiswa : Istiningdyah Saptarini
NRP : 5110100057
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Umi Laili Yuhana, S.Kom, M.Sc.
Dosen Pembimbing 2 : Dr. Ir. Siti Rochimah, MT.

ABSTRAK

Proses penjaminan kualitas merupakan salah satu tahap yang penting dalam siklus hidup perangkat lunak. Salah satu cara untuk menjamin kualitas perangkat lunak adalah dengan melakukan evaluasi terhadap perangkat lunak. Evaluasi yang baik dilakukan berdasarkan standar kualitas yang telah baku. Salah satu standar kualitas baku yang dapat digunakan adalah ISO/IEC 9126. ISO/IEC 9126 memiliki enam karakteristik yang mampu mengukur kualitas perangkat lunak baik dari aspek internal maupun eksternal. Portabilitas merupakan salah satu karakteristik pada ISO/IEC 9126 yang mengukur kemampuan perangkat lunak dalam beradaptasi di lingkungan tertentu.

Pada Tugas Akhir ini studi kasus SIAKAD ITS modul penilaian dievaluasi menggunakan model kualitas ISO/IEC 9126. Aspek yang dievaluasi adalah aspek portabilitas. Untuk mendukung proses evaluasi, kakas bantu evaluasi dibangun dengan mengacu pada karakteristik portabilitas ISO/IEC 9126-3. Kakas bantu evaluasi yang dibangun dapat digunakan tidak hanya khusus untuk mengevaluasi SIAKAD ITS modul penilaian tetapi juga untuk mengevaluasi perangkat lunak lain.

Hasil evaluasi SIAKAD ITS modul penilaian menunjukkan hasil yang bagus. Setelah dilakukan penyesuaian kebutuhan dan peninjauan terhadap SIAKAD ITS modul penilaian, dari 12 metrik pengukuran yang ada hanya tiga yang digunakan untuk evaluasi. Evaluasi dilakukan baik secara manual dan dengan menggunakan kakas bantu. Kakas bantu memberikan hasil yang sama dengan evaluasi secara manual.

Kata kunci: ISO/IEC 9126, kakas bantu evaluasi perangkat lunak, model kualitas, sistem informasi akademik.

Design and Implementation ISO/IEC 9126 Quality Model Portability Characteristic Evaluation CASE Tool

Student Name : Istiningdyah Saptarini
Student ID : 5110100057
Major : Informatics Engineering Department - ITS
Advisor 1 : Umi Laili Yuhana, S.Kom, M.Sc.
Advisor 2 : Dr. Ir. Siti Rochimah, MT.

ABSTRACT

Quality assurance process is one of the important stage in the software life cycle. One way to ensure software quality is to evaluate the software. A good evaluation is done based on quality standards. One of the quality standards that can be used is the ISO / IEC 9126. ISO / IEC 9126 has six characteristics that are capable of measuring the quality of software from both internal and external aspects. Portability is one of the characteristics of the ISO / IEC 9126 which measures the software's ability to adapt to a particular environment.

In this final project, case study ITS Academic Information System Assessment Module is evaluated using the ISO/IEC 9126 quality model. Aspect evaluated is the portability aspect. To support the evaluation process, software evaluation tool was built refers to portability characteristic of ISO/IEC 9126-3. Software evaluation tool which was built is able to evaluate not only ITS Academic Information System but also other software.

Evaluation on ITS Academic Information System Assessment Module shows good results. After needs adjustment and a review of ITS Academic Information System Assessment Module, from twelve measurement metrics only three are used for evaluation. The evaluation is done both manually and by using the software evaluation tool. Software evaluation tool gives the same result as the evaluation which is done manually.

Keyword: *academic informatic system, ISO/IEC 9126, software evaluation tools, quality model.*

DAFTAR KODE SUMBER

Kode Sumber 4.1. Fungsi <i>Getter</i> dan <i>Setter</i> Properti <i>idatribut</i>	43
Kode Sumber 4.2. Fungsi <i>insertAtribut</i>	47
Kode Sumber 4.3. Fungsi <i>updateAtribut</i>	47
Kode Sumber 4.4. Fungsi <i>selectAtribut</i>	48
Kode Sumber 4.5. Fungsi <i>selectAtributById</i>	48
Kode Sumber 4.6. Fungsi <i>validate</i>	48
Kode Sumber 4.7. Fungsi <i>hitungMetrik</i>	50
Kode Sumber 4.8. Fungsi <i>hitungSubkarakteristik</i>	50

BAB II

DASAR TEORI

Bab ini membahas teori yang menunjang pengerjaan Tugas Akhir. Teori tersebut antara lain adalah penjaminan kualitas perangkat lunak, model kualitas ISO/IEC 9126-3, serta kerangka kerja pemetaan objek relasional Hibernate.

2.1. Penjaminan Kualitas Perangkat Lunak [2]

Kualitas perangkat lunak menurut IEEE adalah tingkat seberapa jauh sebuah sistem, komponen, atau proses dapat memenuhi kebutuhan yang ditentukan dan ekspektasi pengguna. Agar perangkat lunak dapat memenuhi kebutuhan dan ekspektasi pengguna, maka perlu dilakukan penjaminan kualitas.

Aktivitas penjaminan kualitas meliputi aspek fungsional, manajerial, dan ekonomis dari proses pengembangan dan pemeliharaan perangkat lunak. Sistem penjaminan kualitas mengombinasikan berbagai macam komponen yang ditujukan untuk mencapai tingkat kualitas yang dapat diterima. Komponen-komponen penjaminan kualitas perangkat lunak antara lain sebagai berikut.

- Komponen praprojek, untuk menjamin bahwa ketentuan proyek telah ditetapkan terkait dengan sumber daya, jadwal, biaya, serta untuk menjamin bahwa rencana kualitas dan pengembangan perangkat lunak telah ditentukan.
- Komponen aktivitas penilaian siklus hidup proyek, untuk menjamin kualitas perangkat lunak baik dalam tahap pengembangan maupun tahap pemeliharaan.
- Komponen pencegahan kesalahan dan peningkatan infrastruktur, untuk meminimalkan tingkat kesalahan berdasarkan pengalaman aktivitas penjaminan kualitas suatu organisasi.

- Komponen manajemen kualitas perangkat lunak, untuk mengatur aktivitas pengembangan, pemeliharaan, dan aktivitas manajerial pendukung.
- Komponen standardisasi, sertifikasi, dan penilaian sistem penjaminan kualitas, sebagai media koordinasi sistem kualitas antar organisasi, dan penilaian pencapaian kualitas berdasarkan skala universal.
- Komponen manusia, yang meliputi manajer, tim penguji, dan praktisi di bidang penjaminan kualitas perangkat lunak. Aktor-aktor tersebut berkontribusi sebagai pelaksana dan pendukung proses penjaminan kualitas perangkat lunak.

2.2. Model Kualitas ISO/IEC 9126

The International Organization for Standardization (ISO) dan *the International Electrotechnical Commission* (IEC) telah membentuk standar internasional untuk model kualitas ISO/IEC 9126 yang digunakan untuk mengevaluasi kualitas perangkat lunak. Tujuan ISO/IEC 9126 adalah untuk memberikan standar baku dalam proyek pengembangan perangkat lunak. ISO/IEC 9126 terdiri atas empat bagian, yaitu ISO/IEC 9126-1, ISO/IEC 9126-2, ISO/IEC 9126-3, ISO/IEC 9126-4. ISO/IEC 9126-1 menjelaskan tentang klasifikasi kualitas perangkat lunak sebagai serangkaian karakteristik dan subkarakteristik. ISO/IEC 9126-2 menjelaskan tentang metrik eksternal. Metrik eksternal dapat diukur ketika perangkat lunak sedang beroperasi. Sedangkan ISO/IEC 9126-3 menjelaskan tentang metrik internal. Metrik internal dapat diukur tanpa pengoperasian perangkat lunak. ISO/IEC 9126-4 menjelaskan tentang penggunaan metrik kualitas perangkat lunak untuk tiap karakteristik yang telah dijabarkan di ISO/IEC 9126-1 [3].

Model kualitas ISO/IEC 9126 dibagi menjadi enam karakteristik. ISO/IEC 9126 dapat diaplikasikan oleh siapa saja bergantung pada karakteristik yang diukur. Ada beberapa karakteristik yang dapat diaplikasikan oleh orang yang tidak memiliki latar belakang IT namun ada pula karakteristik yang hanya dapat diaplikasikan oleh orang dengan latar belakang IT [4]. Tiap

karakteristik memiliki beberapa subkarakteristik. Karakteristik-karakteristik tersebut antara lain sebagai berikut.

- Karakteristik fungsionalitas, digunakan untuk memprediksi tingkat kepuasan kebutuhan fungsional yang diimplementasikan. Fungsionalitas memiliki subkarakteristik kesesuaian, akurasi, interoperabilitas, keamanan, dan pemenuhan aspek fungsionalitas.
- Karakteristik keandalan, digunakan untuk memprediksi tingkat keandalan sistem yang dibangun pada proses pengembangan. Keandalan memiliki subkarakteristik kedewasaan, toleransi kesalahan, kemampuan pengembalian kondisi awal (*recoverability*), dan pemenuhan aspek keandalan.
- Karakteristik kebergunaan, digunakan untuk memprediksi tingkat kemudahan pemahaman, pengoperasian, dan kesesuaian sistem. Kebergunaan memiliki subkarakteristik pemahaman, pembelajaran, operasionalitas, daya tarik, dan pemenuhan aspek kebergunaan.
- Karakteristik efisiensi, digunakan untuk mengukur efisiensi perangkat lunak selama pengoperasian. Efisiensi memiliki subkarakteristik perilaku waktu, penggunaan sumber daya, dan pemenuhan aspek efisiensi.
- Karakteristik pemeliharaan, digunakan untuk memprediksi tingkat usaha yang diperlukan untuk memodifikasi perangkat lunak. Pemeliharaan memiliki subkarakteristik analisis, perubahan, stabilitas, pengujian, dan pemenuhan aspek pemeliharaan.
- Karakteristik portabilitas, digunakan untuk memprediksi dampak perangkat lunak yang terjadi pada sistem saat aktivitas pemasangan perangkat lunak dilakukan. Portabilitas memiliki

subkarakteristik adaptasi, pemasangan, keberdampingan (*co-existence*), penggantian (*replaceability*), dan pemenuhan aspek portabilitas [1].

Tabel 2.1 Bobot Subkarakteristik pada Karakteristik Portabilitas [5]

Subkarakteristik	Bobot
Adaptasi	Tinggi
Pemasangan	Rendah
Keberdampingan	Tinggi
Penggantian	Sedang
Pemenuhan aspek portabilitas	tinggi

2.2.1. Karakteristik Portabilitas ISO/IEC 9126-3 [1]

Karakteristik portabilitas merupakan karakteristik yang mencerminkan kemampuan perangkat lunak untuk dapat dipindahkan dari satu lingkungan ke lingkungan yang lain. Karakteristik portabilitas ISO/IEC 9126-3 memiliki lima subkarakteristik. Masing-masing subkarakteristik memiliki bobot yang berbeda. Pembobotan masing-masing subkarakteristik dapat dilihat pada Tabel 2.1. Subkarakteristik pertama, yaitu adaptasi bertujuan untuk mengukur kemampuan perangkat lunak dalam beradaptasi ke berbagai lingkungan tanpa ada batasan dan mampu menghasilkan keluaran yang diharapkan pengguna. Subkarakteristik ini terdiri atas lima metrik pengukuran, yaitu metrik adaptasi sumber data, adaptasi lingkungan perangkat keras, adaptasi lingkungan organisasi, adaptasi lingkungan sistem perangkat lunak, dan kemudahan untuk melakukan pemindahan perangkat lunak. Subkarakteristik pemasangan bertujuan untuk mengukur kemampuan perangkat lunak untuk dapat dipasang pada lingkungan tertentu. Subkarakteristik pemasangan terdiri atas metrik kemudahan percobaan ulang pengaturan, usaha pemasangan, dan fleksibilitas pemasangan. Subkarakteristik keberdampingan bertujuan untuk mengukur kemampuan perangkat lunak untuk dapat berdampingan dengan perangkat lain yang berbagi sumber daya dalam satu

lingkungan. Subkarakteristik keberdampingan hanya terdiri dari satu metrik, yaitu metrik ketersediaan keberdampingan. Subkarakteristik selanjutnya yaitu penggantian. Subkarakteristik penggantian bertujuan untuk mengukur kemampuan perangkat lunak ketika digunakan untuk menggantikan perangkat lunak lain dengan tujuan sama dan dalam lingkungan yang sama. Subkarakteristik ini terdiri atas metrik kelangsungan penggunaan data dan metrik cakupan fitur. Subkarakteristik terakhir yaitu pemenuhan aspek portabilitas. Subkarakteristik ini bertujuan untuk mengukur kemampuan perangkat lunak dalam memenuhi standar atau aturan yang berkaitan dengan aspek portabilitas yang berlaku. Subkarakteristik ini hanya terdiri dari satu metrik, yaitu metrik pemenuhan aspek portabilitas.

2.3. Sistem Informasi Akademik ITS

Sistem Informasi Akademik (SIKAD) ITS Surabaya adalah sistem informasi manajemen mahasiswa yang digunakan untuk menyimpan data akademik institut. SIKAD mampu menyimpan, memproses, dan membantu menghasilkan laporan akademik sesuai dengan kebutuhan kegiatan belajar-mengajar di ITS Surabaya. SIKAD terintegrasi dengan sistem informasi lain seperti SIM Beasiswa, SIM Reporting, SIM Penalaran, SIM SKEM, dsb. untuk membantu proses pengambilan keputusan. Pengguna dari SIKAD antara lain mahasiswa, dosen, staf administrasi, dan pihak manajemen. Setiap pengguna memiliki hak akses masing-masing sesuai dengan tipe pengguna.

SIKAD memiliki beragam fitur untuk masing-masing jenis pengguna. Untuk mahasiswa, fitur yang ada pada SIKAD antara lain melihat biodata, mengisi form rencana studi (FRS), melihat transkrip, jadwal perkuliahan, daftar mata kuliah, nilai, ranking IPK, mengisi kuesioner kuliah dan dosen, serta memperbarui data proses wisuda.

Untuk dosen, fitur yang ada pada SIKAD antara lain melihat biodata anak wali, menyetujui FRS mahasiswa, melihat daftar mata kuliah, hasil penilaian dosen, jadwal mengajar dan

perkuliahan anak wali, nilai, ranking IPK, dan memasukkan nilai mata kuliah.

Sedangkan untuk staf administrasi, fitur yang ada pada SIAKAD adalah mengatur perwalian mahasiswa, melihat dan mengubah data mahasiswa dan dosen, melihat daftar mata kuliah, mengatur jadwal, pra syarat, dan aturan perkuliahan, mengatur pembagian kelas, mata kuliah yang ditawarkan, mencetak absen perkuliahan, melihat jadwal perkuliahan, nilai, daftar perwalian, ranking IPK, melihat status pembayaran spp mahasiswa, dan mencetak transkrip mahasiswa.

Dari fitur-fitur yang telah disebutkan sebelumnya, yang termasuk ke dalam modul penilaian adalah fitur yang berhubungan dengan pelaporan nilai, baik nilai mahasiswa maupun nilai Indeks Prestasi Dosen (IPD), dan memasukkan hasil penilaian dari dosen. Fitur-fitur inilah yang menjadi studi kasus dalam Tugas Akhir ini.

2.4. Kerangka Kerja Pemetaan Objek Relasional Hibernate

Kerangka kerja pemetaan objek relasional adalah kaskas pemetaan yang berfungsi untuk menghubungkan representasi model objek pada kode sumber dengan representasi model objek pada basis data relasional dan sebaliknya. Kerangka kerja pemetaan objek relasional Hibernate merupakan pustaka untuk lingkungan pengembangan berbasis Java. Kerangka kerja Hibernate terbaru yang telah dirilis hingga saat ini adalah Hibernate ORM 4.2.14 [6].

Selain pemetaan model objek, Hibernate juga menangani proses pemanggilan dan pengambilan data. Hibernate menggunakan bahasa *Hibernate Query Language* (HQL) untuk melakukan pengaksesan data. HQL adalah bahasa *query* yang mampu melakukan pengaksesan objek data Hibernate dengan berorientasi pada objek. Hal ini dapat mengurangi waktu pengembangan dibandingkan dengan menggunakan *Structured Query Language* (SQL) biasa. Kerangka kerja Hibernate cocok digunakan untuk perangkat lunak yang memiliki model domain berorientasi objek [7].

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis dan perancangan dari sistem yang akan dibangun. Analisis membahas permasalahan yang diangkat dalam Tugas Akhir. Selanjutnya dibahas mengenai perancangan sistem yang dibuat. Pendekatan yang digunakan dalam perancangan ini adalah pendekatan berorientasi objek. Perancangan direpresentasikan dengan menggunakan diagram UML.

3.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain penentuan parameter pengukuran dari karakteristik portabilitas ISO/IEC 9126-3 dan spesifikasi kebutuhan perangkat lunak.

3.1.1. Penentuan Parameter Pengukuran

Subbab ini membahas analisis metrik-metrik yang ada pada karakteristik portabilitas. Metrik-metrik tersebut dikelompokkan menjadi beberapa subkarakteristik dengan bobot tertentu. Tiap metrik memiliki parameter tertentu yang akan menentukan nilai kualitas dari perangkat lunak yang akan diukur. Parameter disesuaikan agar dapat memenuhi tujuan perhitungan dari masing-masing metrik. Penentuan parameter beserta interpretasi hasil pengukuran dilakukan berdasarkan ISO/IEC 9126-3. Interpretasi hasil pengukuran dan metrik beserta parameter dijelaskan sebagai berikut.

3.1.1.1. Subkarakteristik Adaptasi

Subkarakteristik adaptasi bertujuan untuk memprediksi dampak perangkat lunak pada pengguna yang akan mengadaptasi perangkat lunak ke lingkungan tertentu. Lingkungan tersebut dapat berupa lingkungan perangkat keras, perangkat lunak, maupun lingkungan organisasi. Subkarakteristik ini memiliki bobot tinggi. Subkarakteristik adaptasi terdiri dari lima metrik yang dapat mengukur kemampuan perangkat lunak untuk beradaptasi. Metrik-metrik tersebut antara lain adalah metrik adaptasi sumber data, adaptasi lingkungan perangkat keras, adaptasi lingkungan organisasi,

adaptasi lingkungan perangkat lunak, dan kemudahan untuk melakukan pemindahan perangkat lunak. Tabel 3.1 menjelaskan tentang metrik adaptasi sumber data, sedangkan untuk detail metrik lainnya dapat dilihat pada Lampiran A. Penentuan parameter pada metrik-metrik yang ada dalam subkarakteristik adaptasi dapat dilihat pada Tabel 3.2.

Tabel 3.1 Metrik Adaptasi Struktur Data [4]

Komponen	Deskripsi
Nama Metrik	Adaptasi sumber data
Tujuan Metrik	Mengukur seberapa adaptif perangkat lunak terhadap perubahan data
Metode Pengukuran	Menghitung jumlah sumber data (tabel dalam basis data atau <i>file</i> penyimpanan data lain) yang digunakan dan tidak memiliki batasan setelah perubahan dan membandingkan dengan jumlah seluruh sumber data yang membutuhkan kemampuan adaptasi
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah sumber data yang berhasil beradaptasi B = jumlah sumber data yang membutuhkan kemampuan adaptasi
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Spesifikasi kebutuhan, rancangan sistem, dan laporan peninjauan
Target pemirsa	Pengembang dan pemelihara

Tabel 3.2 Parameter Metrik pada Subkarakteristik Adaptasi

no	Nama metrik	Formula	Parameter A	Parameter B	Interpretasi hasil
1	Adaptasi struktur data	$X = \frac{A}{B}$	Jumlah sumber data yang berhasil beradaptasi (contoh: jumlah tabel pada basis data yang digunakan)	Jumlah sumber data yang membutuhkan kemampuan beradaptasi	x semakin baik apabila nilainya semakin mendekati 1
2	Adaptasi lingkungan perangkat keras	$X = \frac{A}{B}$	Jumlah fitur dari kebutuhan adaptasi lingkungan perangkat keras yang berhasil beradaptasi	Jumlah fitur dengan kebutuhan untuk dapat beradaptasi pada lingkungan perangkat keras tertentu	x semakin baik apabila nilainya semakin mendekati 1
3	Adaptasi lingkungan organisasi	$X = \frac{A}{B}$	Jumlah fitur dari kebutuhan adaptasi lingkungan organisasi yang berhasil beradaptasi	Jumlah fitur dengan kebutuhan untuk dapat beradaptasi pada lingkungan organisasi tertentu	x semakin baik apabila nilainya semakin mendekati 1
4	Adaptasi lingkungan sistem perangkat lunak	$X = \frac{A}{B}$	Jumlah fitur dari kebutuhan adaptasi lingkungan sistem perangkat lunak yang berhasil beradaptasi	Jumlah fitur dengan kebutuhan untuk dapat beradaptasi pada lingkungan sistem perangkat lunak tertentu	x semakin baik apabila nilainya semakin mendekati 1
5	Kemudahan untuk melakukan pemindah-	$X = \frac{A}{B}$	jumlah fitur yang mendukung kemudahan	jumlah fitur dengan kebutuhan kemampuan	x semakin baik apabila nilainya semakin

Tabel 3.2 Parameter Metrik pada Subkarakteristik Adaptasi (Lanjutan)

	an perangkat lunak		adaptasi bagi pengguna	kemudahan beradaptasi	mendekati 1
--	--------------------	--	------------------------	-----------------------	-------------

3.1.1.2. Subkarakteristik Pemasangan

Subkarakteristik pemasangan bertujuan untuk memprediksi dampak perangkat lunak pada pengguna yang akan memasang perangkat lunak pada lingkungan kerja pengguna. Bobot subkarakteristik pemasangan termasuk ke dalam kategori rendah. Subkarakteristik pemasangan terdiri dari metrik kemudahan percobaan ulang pengaturan, metrik usaha pemasangan, dan metrik fleksibilitas pemasangan. Penjelasan detil mengenai metrik-metrik tersebut dapat dilihat pada Lampiran A. Penentuan parameter pada metrik-metrik yang ada dalam subkarakteristik adaptasi dapat dilihat pada Tabel 3.3.

Tabel 3.3 Parameter Metrik pada Subkarakteristik Pemasangan

no	Nama metrik	Formu-la	Parameter A	Parameter B	Interpretasi hasil
1	Kemudahan percobaan ulang pengaturan	$X = \frac{A}{B}$	Jumlah operasi percobaan ulang untuk pengaturan	Jumlah operasi pengaturan yang dibutuhkan	x semakin baik apabila nilainya semakin mendekati 1
2	Usaha pemasangan	$X = \frac{A}{B}$	Jumlah langkah pemasangan yang berjalan secara otomatis	Jumlah langkah pemasangan yang dibutuhkan	x semakin baik apabila nilainya semakin mendekati 1
3	Fleksibilitas pemasangan	$X = \frac{A}{B}$	Jumlah operasi pemasangan yang memiliki kemampuan penyesuaian (dapat diatur oleh pengguna)	Jumlah operasi pemasangan dengan kebutuhan kemampuan penyesuaian	x semakin baik apabila nilainya semakin mendekati 1

3.1.1.3. Subkarakteristik Keberdampingan

Subkarakteristik keberdampingan terdiri dari metrik ketersediaan keberdampingan. Subkarakteristik ini bertujuan untuk memprediksi dampak perangkat lunak pada perangkat lunak lain yang menggunakan sumber daya perangkat keras operasional yang sama. Bobot subkarakteristik keberdampingan termasuk ke dalam kategori tinggi. Penjelasan detail mengenai metrik ketersediaan keberdampingan dapat dilihat pada Lampiran A. Penentuan parameter pada metrik-metrik yang ada dalam subkarakteristik adaptasi dapat dilihat pada Tabel 3.4.

Tabel 3.4 Parameter Metrik pada Subkarakteristik Keberdampingan

no	Nama metrik	Formula	Parameter A	Parameter B	Interpretasi hasil
1	Ketersediaan keberdampingan	$X = \frac{A}{B}$	jumlah entitas perangkat lunak yang dapat berdampingan	jumlah entitas pada lingkungan sistem yang seharusnya dapat berdampingan	x semakin baik apabila nilainya semakin mendekati 1

3.1.1.4. Subkarakteristik Penggantian

Subkarakteristik penggantian terdiri dari metrik kelangsungan penggunaan data dan metrik cakupan fitur. Subkarakteristik ini bertujuan untuk memprediksi dampak perangkat lunak pada pengguna yang mencoba untuk menggunakan perangkat lunak sebagai pengganti perangkat lunak lain pada lingkungan tertentu. Metrik-metrik ini dapat diterapkan pada perangkat lunak yang merupakan pengganti dari perangkat lunak versi sebelumnya. Bobot subkarakteristik penggantian termasuk ke dalam kategori sedang. Penjelasan detail mengenai metrik-metrik tersebut dapat dilihat pada Lampiran A. Penentuan parameter pada metrik-metrik yang ada dalam subkarakteristik adaptasi dapat dilihat pada Tabel 3.5.

Tabel 3.5 Parameter Metrik pada Subkarakteristik Penggantian

no	Nama metrik	Formula	Parameter A	Parameter B	Interpretasi hasil
1	Kelangsungan penggunaan data	$X = \frac{A}{B}$	Jumlah sumber data yang tetap digunakan pada perangkat lunak yang baru	Jumlah sumber data asli yang seharusnya tetap digunakan setelah penggantian perangkat lunak	x semakin baik apabila nilainya semakin mendekati 1
2	Cakupan fungsi	$X = \frac{A}{B}$	Jumlah fitur yang tercakup dalam perangkat lunak baru yang mengeluarkan hasil yang sama dengan fitur pada perangkat lunak lama	Jumlah fitur pada perangkat lunak lama	x semakin baik apabila nilainya semakin mendekati 1

3.1.1.5. Subkarakteristik Pemenuhan Aspek Portabilitas

Subkarakteristik pemenuhan aspek portabilitas hanya terdiri atas satu pengukuran, yaitu pemenuhan aspek portabilitas. Kepatuhan pemenuhan portabilitas mencakup semua aspek portabilitas dalam sistem. Tujuan dari metrik ini adalah menilai kemampuan perangkat lunak dalam memenuhi standar dan aturan pada organisasi pengguna yang berhubungan dengan aspek portabilitas. Bobot subkarakteristik ini termasuk ke dalam kategori tinggi. Penjelasan detail mengenai metrik pemenuhan aspek portabilitas dapat dilihat pada Lampiran A. Penentuan parameter

pada metrik-metrik yang ada dalam subkarakteristik adaptasi dapat dilihat pada Tabel 3.6.

Tabel 3.6 Parameter Metrik pada Subkarakteristik Pemenuhan Aspek Portabilitas

no	Nama metrik	Formula	Parameter A	Parameter B	Interpretasi hasil
1	Pemenuhan aspek portabilitas	$X = \frac{A}{B}$	Bagian dari perangkat lunak yang telah memenuhi aturan/standar/regulasi	Bagian dari perangkat lunak yang harus mengikuti aturan/standar/regulasi tertentu yang berkaitan dengan portabilitas	x semakin baik apabila nilainya semakin mendekati 1

3.1.2. Spesifikasi Kebutuhan Perangkat Lunak

Sistem yang dibuat yaitu berupa kakas bantu. Kakas bantu ini dapat membantu pengguna untuk mengevaluasi karakteristik portabilitas dari perangkat lunak berdasarkan ISO/IEC 9126-3. Kakas bantu dibuat berdasarkan ketentuan yang tercantum pada ISO/IEC 9126-1 dan ISO/IEC 9126-3. Pengguna dapat memasukkan parameter-parameter yang dibutuhkan untuk pengukuran karakteristik portabilitas. Kakas dapat menghitung dan menampilkan nilai portabilitas dari perangkat lunak yang dievaluasi.

3.1.2.1. Aktor

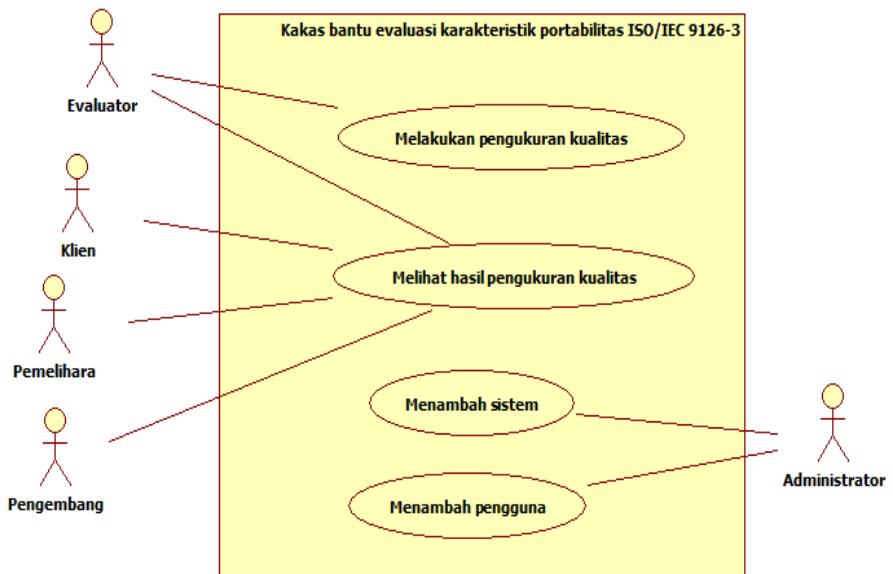
Aktor mendefinisikan pihak-pihak, baik manusia maupun sistem / perangkat lunak lain, yang terlibat dan berinteraksi langsung dengan sistem. Aktor yang terdapat pada sistem ini adalah evaluator, klien, pengembang, pemelihara, dan administrator. Peranan aktor pada sistem dibahas secara lebih rinci pada subbab 3.1.2.2 tentang kasus penggunaan.

3.1.2.2. Kasus Penggunaan

Pada bagian ini dibahas secara rinci masing-masing kasus penggunaan, termasuk didalamnya adalah spesifikasi kasus penggunaan, diagram aktivitas, dan diagram urutan untuk setiap kasus penggunaan. Tugas Akhir ini memiliki empat kasus penggunaan. Gambar untuk semua kasus penggunaan dapat dilihat pada Gambar 3.1.

3.1.2.3. Kasus Penggunaan Menambah Pengguna

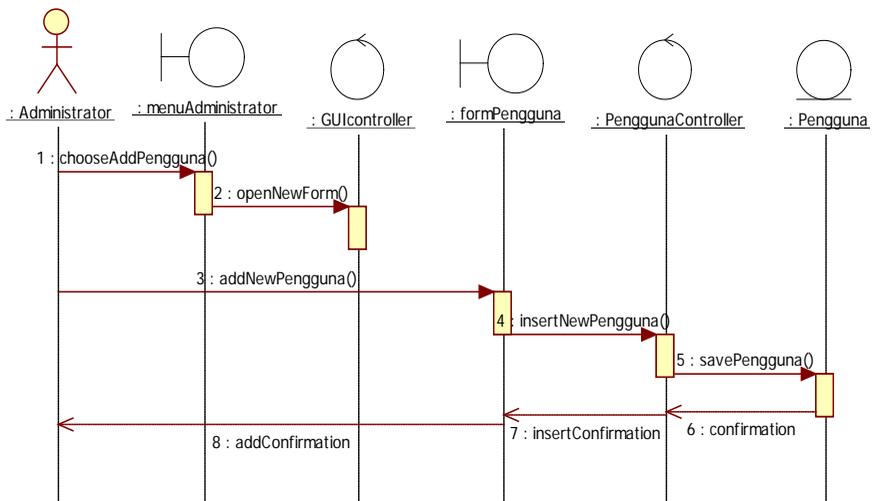
Pada kasus penggunaan ini, administrator dapat menambahkan data pengguna ke dalam basis data. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.7. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini dapat dilihat pada Gambar 3.3 dan Gambar 3.2.



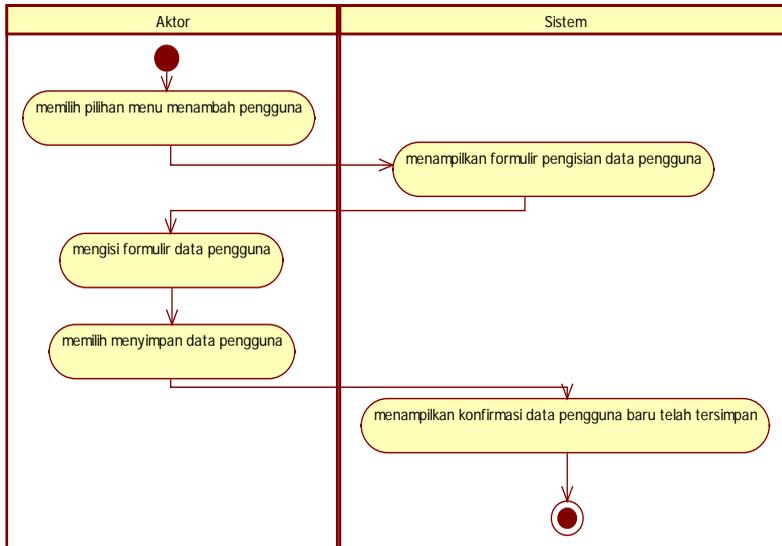
Gambar 3.1 Diagram Kasus Penggunaan

Tabel 3.7 Spesifikasi Kasus Penggunaan Menambah Pengguna

Nama	Menambah pengguna
Kode	UC-0001
Deskripsi	Aktor dapat menambahkan data pengguna baru ke dalam basis data
Tipe	Fungsional
Pemicu	Aktor memilih pilihan menu untuk menambah pengguna
Aktor	Administrator
Kondisi Awal	-
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor memilih pilihan menu untuk menambah pengguna 2. Sistem menampilkan formulir pengisian data pengguna 3. Aktor mengisi formulir data pengguna 4. Aktor memilih untuk menyimpan data pengguna 5. Sistem menampilkan konfirmasi bahwa data pengguna telah tersimpan ke dalam basis data
- Kejadian Alternatif	-
Kondisi Akhir	Data pengguna baru telah tersimpan ke dalam basis data



Gambar 3.2 Diagram Urutan Menambah Pengguna



Gambar 3.3 Diagram Aktivitas Menambah Pengguna

3.1.2.4. Kasus Penggunaan Menambah Sistem

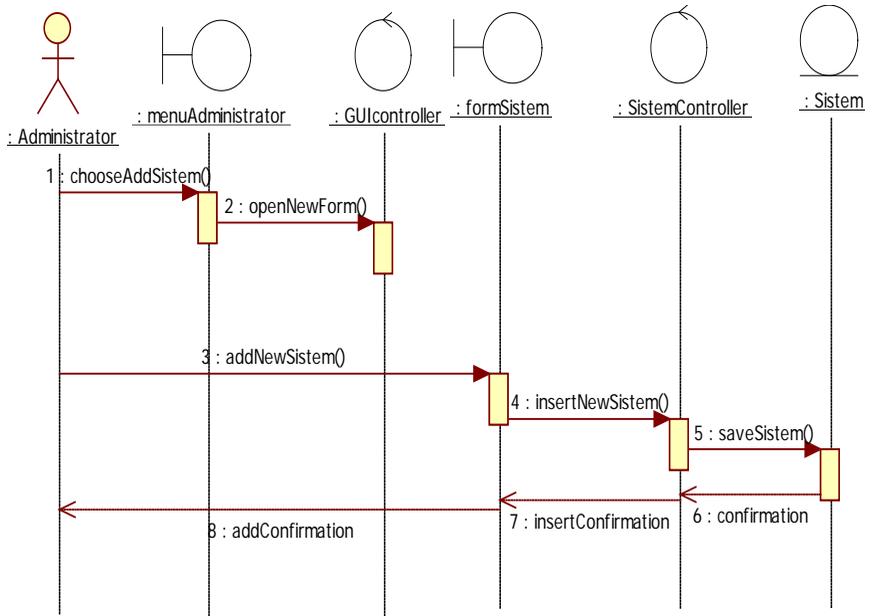
Administrator dapat menambahkan data sistem yang akan dievaluasi ke dalam basis data. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.8. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 3.5 dan Gambar 3.4.

Tabel 3.8 Spesifikasi Kasus Penggunaan Menambah Sistem

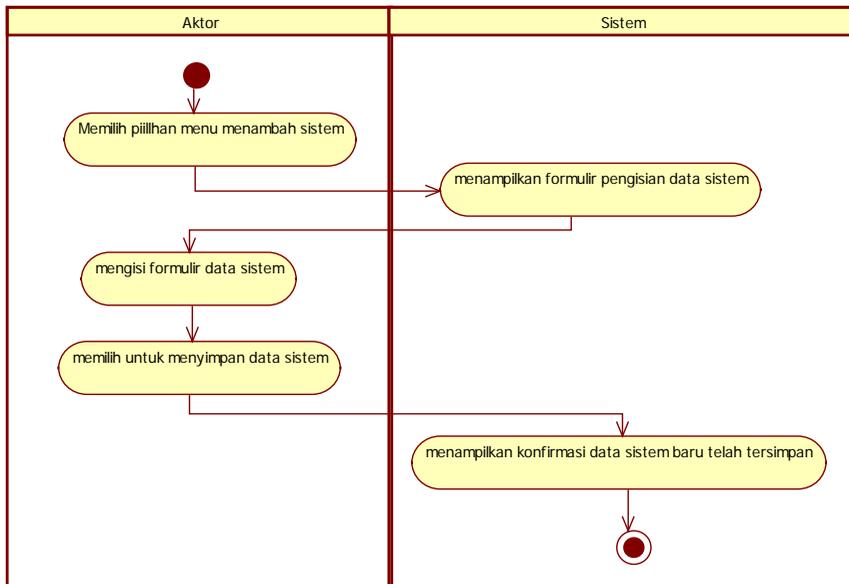
Nama	Menambah sistem
Kode	UC-0002
Deskripsi	Aktor dapat menambahkan data sistem baru ke dalam basis data
Tipe	Fungsional
Pemicu	Aktor memilih pilihan menu untuk menambah sistem
Aktor	Administrator
Kondisi Awal	-
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor memilih pilihan menu untuk menambah sistem 2. Sistem menampilkan formulir pengisian data sistem 3. Aktor mengisi formulir data sistem

Tabel 3.8 Spesifikasi Kasus Penggunaan Menambah Sistem (Lanjutan)

	4. Aktor memilih untuk menyimpan data sistem 5. Sistem menampilkan konfirmasi bahwa data sistem telah tersimpan ke dalam basis data
- Kejadian Alternatif	-
Kondisi Akhir	Data sistem baru tersimpan ke dalam basis data



Gambar 3.4 Diagram Urutan Menambah Sistem



Gambar 3.5 Diagram Aktivitas Menambah Sistem

3.1.2.5. Kasus Penggunaan Melakukan Pengukuran Kualitas

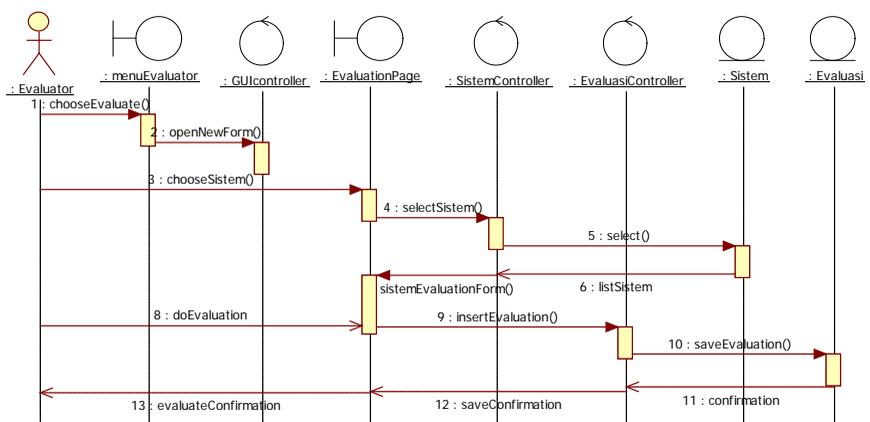
Kasus penggunaan ini dapat dilakukan oleh aktor dengan hak akses evaluator. Pada kasus penggunaan ini, evaluator mula-mula memilih perangkat lunak yang akan dievaluasi dari daftar yang ditampilkan oleh sistem. Kemudian sistem akan menampilkan sejumlah pertanyaan yang harus dijawab oleh evaluator. Setelah itu, sistem akan menampilkan hasil perhitungan berdasarkan metrik yang tercantum ISO/IEC 9126-3. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.9. Diagram aktivitas dari kasus penggunaan ini dapat dilihat pada Gambar 3.7 dan diagram urutan dari kasus penggunaan ini dapat dilihat pada Gambar 3.6.

Tabel 3.9. Spesifikasi Kasus Penggunaan Melakukan Pengukuran Kualitas

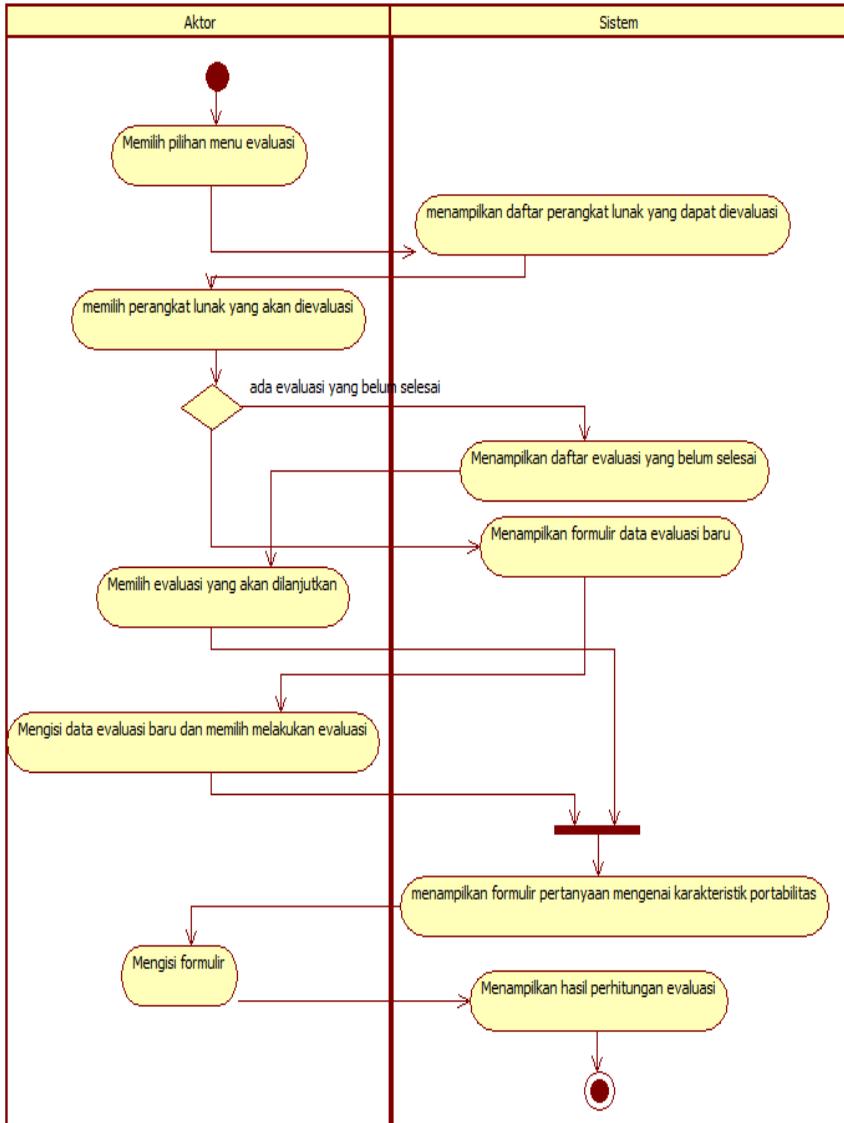
Nama	Melakukan pengukuran kualitas
------	--------------------------------------

Tabel 3.9 Spesifikasi Kasus Penggunaan Melakukan Pengukuran Kualitas (Lanjutan)

Kode	UC-0003
Deskripsi	Sistem dapat menampilkan hasil perhitungan karakteristik portabilitas sesuai dengan parameter inputan aktor.
Tipe	Fungsional
Pemicu	Aktor memilih pilihan menu untuk melakukan evaluasi (pengukuran kualitas)
Aktor	Evaluator
Kondisi Awal	Daftar sistem yang akan dievaluasi sudah ada, Evaluator sudah terdaftar di dalam basis data
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor memilih pilihan menu untuk melakukan evaluasi 2. Sistem menampilkan daftar perangkat lunak yang dapat dievaluasi 3. Aktor memilih perangkat lunak yang akan dievaluasi 4. Sistem menampilkan formulir data evaluasi baru 5. Aktor mengisi formulir data evaluasi dan memilih untuk melakukan evaluasi 6. Sistem menampilkan formulir yang berisi pertanyaan-pertanyaan mengenai karakteristik portabilitas 7. Aktor mengisi formulir 8. Sistem menampilkan hasil perhitungan pada evaluasi
- Kejadian Alternatif	<p>3A Ada evaluasi yang belum selesai dilakukan</p> <p>3B Sistem menampilkan daftar evaluasi yang belum selesai</p> <p>3C Aktor memilih evaluasi yang akan dilanjutkan</p> <p>3D Aliran kembali ke aliran utama langkah nomor 6</p>
Kondisi Akhir	Sistem menampilkan hasil perhitungan dari tiap metrik pada karakteristik portabilitas



Gambar 3.6 Diagram Urutan Melakukan Pengukuran Kualitas



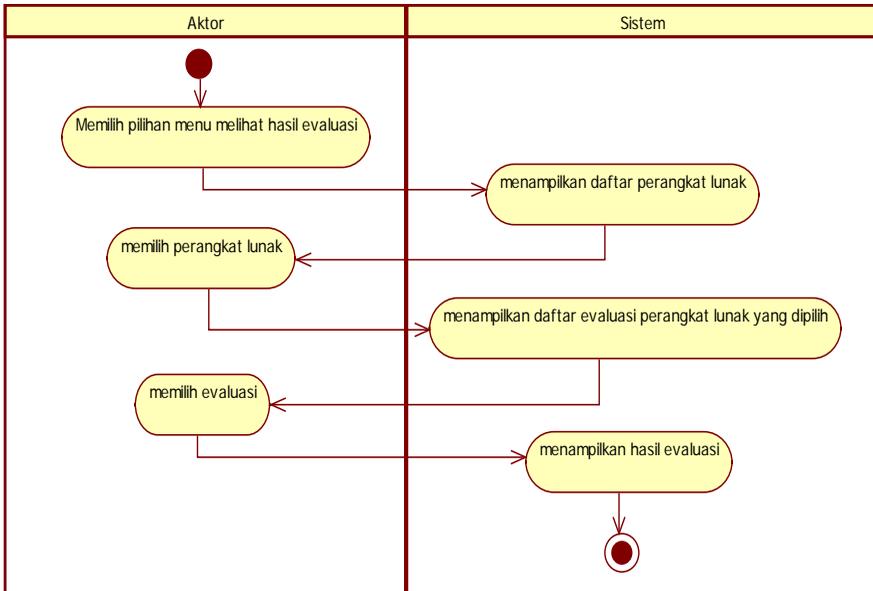
Gambar 3.7 Diagram Aktivitas Melakukan Pengukuran Kualitas

3.1.2.6. Kasus Penggunaan Melihat Hasil Pengukuran Kualitas

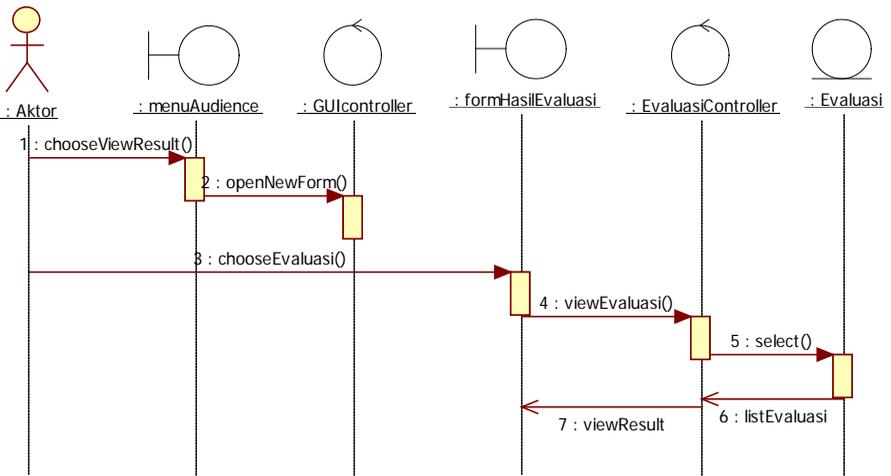
Sistem dapat menampilkan hasil dari pengukuran kualitas yang telah dilakukan sebelumnya. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.10. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini dapat dilihat pada Gambar 3.8 dan Gambar 3.9.

Tabel 3.10 Spesifikasi Kasus Penggunaan Melihat Hasil Pengukuran Kualitas

Nama	Melihat hasil pengukuran kualitas
Kode	UC-0004
Deskripsi	Sistem dapat menampilkan nilai portabilitas dari sistem yang telah diukur (dievaluasi)
Tipe	Fungsional
Pemicu	Aktor memilih pilihan menu untuk melihat hasil evaluasi
Aktor	Klien, pengembang, pemelihara
Kondisi Awal	Daftar sistem sudah ada dan sudah dilakukan evaluasi
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor memilih pilihan menu untuk melihat hasil evaluasi 2. Sistem menampilkan daftar perangkat lunak 3. Aktor memilih perangkat lunak yang akan dilihat nilai portabilitasnya 4. Sistem menampilkan daftar evaluasi yang telah dilakukan terhadap perangkat lunak 5. Aktor memilih evaluasi yang akan dilihat 6. Sistem menampilkan hasil evaluasi
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan hasil perhitungan dari tiap metrik pada karakteristik portabilitas



Gambar 3.8 Diagram Aktivitas Melihat Hasil Evaluasi



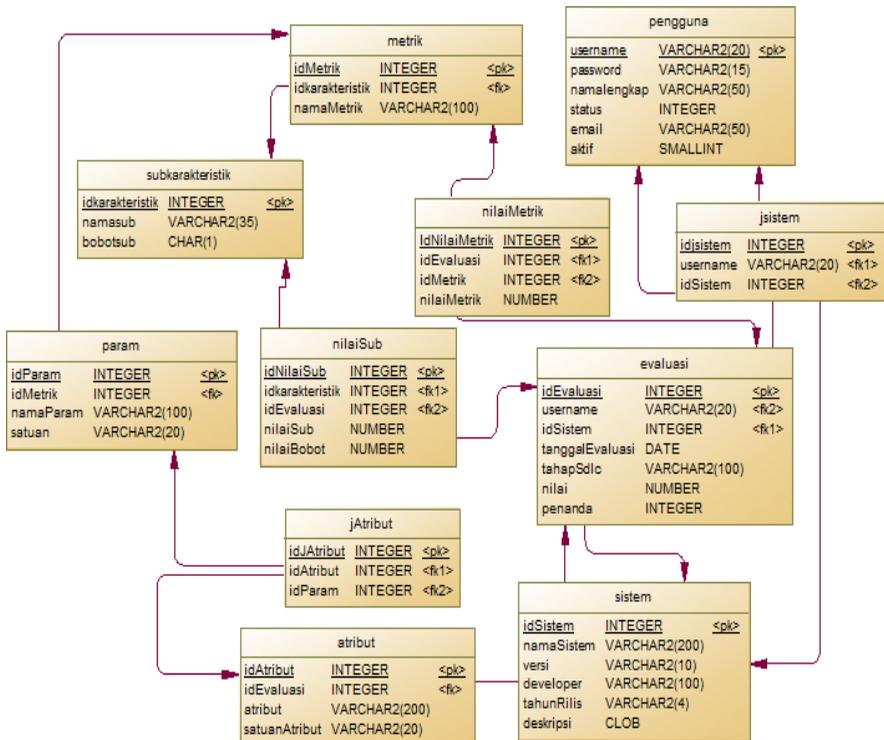
Gambar 3.9 Diagram Urutan Melihat Hasil Evaluasi

3.2. Perancangan Sistem

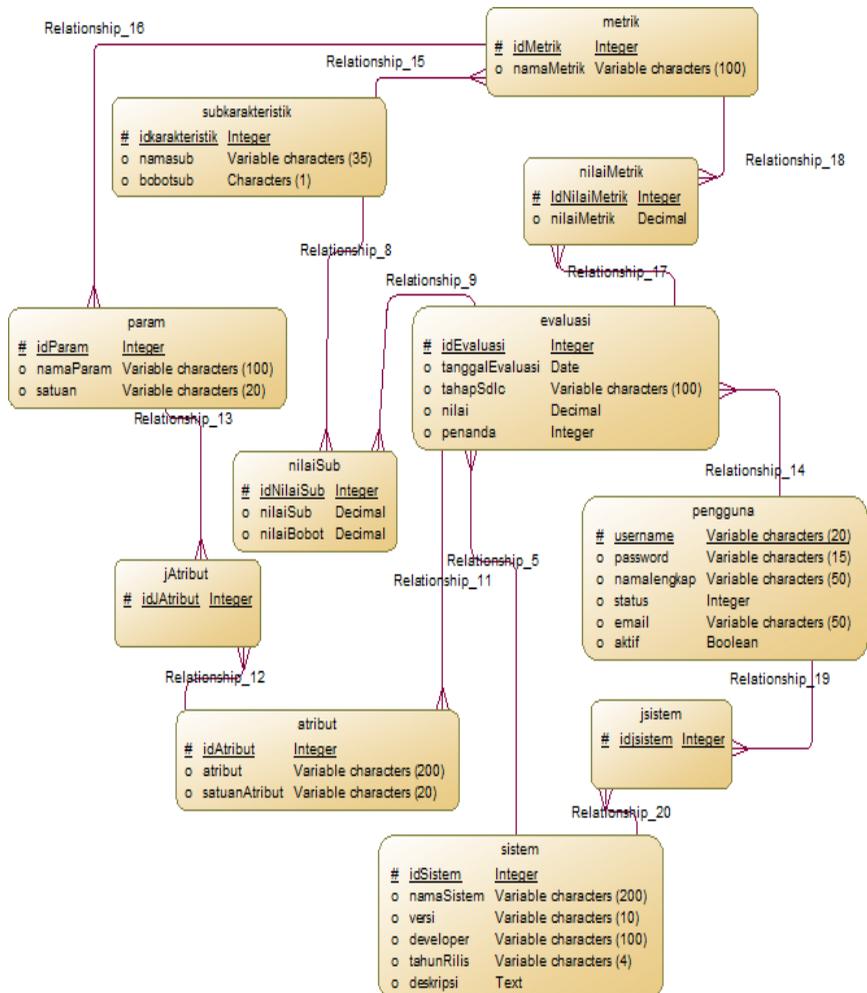
Penjelasan tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan basis data, perancangan diagram kelas, dan perancangan antarmuka.

3.2.1. Perancangan Basis Data

Perancangan basis data berisi rancangan basis data relasional yang digunakan oleh kakas bantu yang dibuat. Rancangan direpresentasikan dengan diagram model data konseptual (CDM) dan diagram model data fisik (PDM). Diagram CDM ditunjukkan pada Gambar 3.11, sedangkan diagram PDM ditunjukkan pada Gambar 3.10.



Gambar 3.10 Diagram Model Data Fisik (PDM)



Gambar 3.11 Diagram Model Data Konseptual (CDM)

Basis data yang digunakan terdiri dari beberapa tabel, antara lain sebagai berikut.

- Tabel pengguna

Tabel ini menyimpan data akun pengguna kakas bantu. Tabel pengguna memiliki hubungan *one-to-many* dengan tabel evaluasi dan tabel jsistem. Satu proses evaluasi ditangani oleh satu akun.

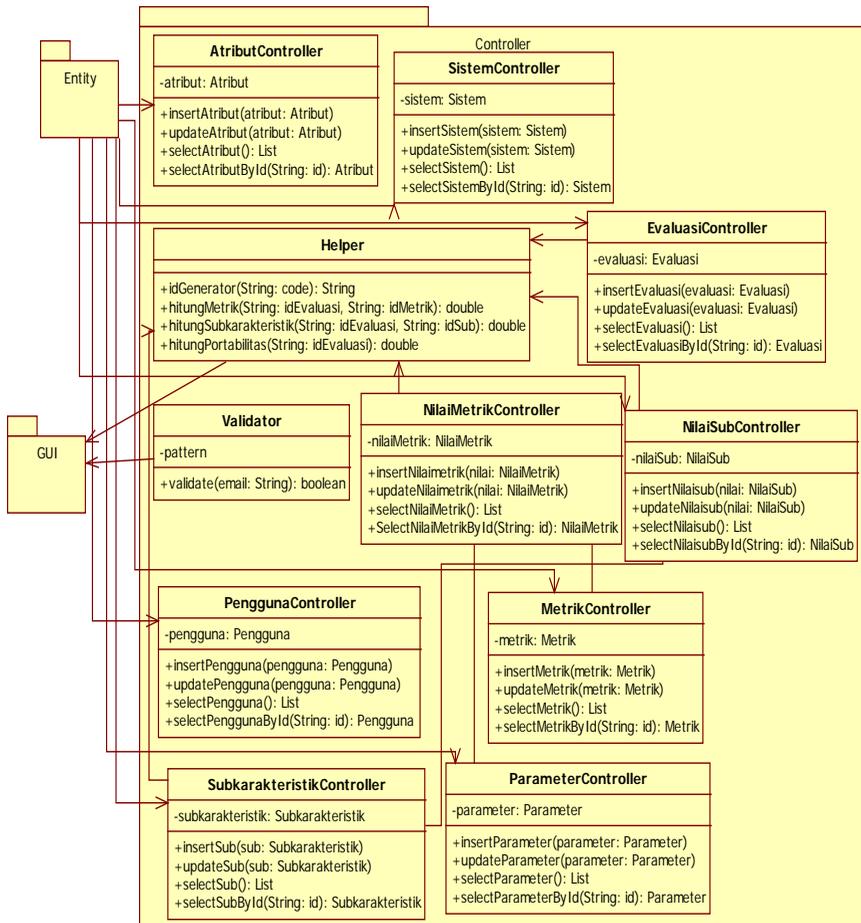
- Tabel jsistem
Tabel ini merupakan tabel *junction* antara tabel pengguna dan tabel evaluasi.
- Tabel subkarakteristik
Tabel ini menyimpan daftar subkarakteristik dari karakteristik portabilitas pada ISO/IEC 9126. Tabel subkarakteristik memiliki hubungan *one-to-many* dengan tabel metrik. Satu subkarakteristik terdiri dari beberapa metrik.
- Tabel metrik
Tabel metrik menyimpan daftar metrik yang ada pada karakteristik portabilitas ISO/IEC 9126. Tabel metrik memiliki hubungan *many-to-one* dengan tabel subkarakteristik dan *one-to-many* dengan tabel param. Satu metrik adalah bagian dari satu subkarakteristik. Satu metrik memiliki beberapa parameter perhitungan.
- Tabel sistem
Tabel sistem adalah tabel untuk menyimpan data sistem yang dievaluasi. Tabel ini memiliki hubungan *one-to-many* dengan tabel evaluasi. Satu sistem dapat mengalami beberapa kali evaluasi.
- Tabel evaluasi
Tabel ini menyimpan informasi proses evaluasi atau pengukuran kualitas. Tabel evaluasi memiliki hubungan *many-to-one* dengan pengguna dan *one-to-many* dengan tabel atribut. Tabel ini juga memiliki hubungan *one-to-many* dengan tabel nilaiMetrik dan tabel nilaiSub. Dalam satu evaluasi terdapat beberapa nilai metrik dan beberapa nilai subkarakteristik.
- Tabel nilaiMetrik

Tabel ini menyimpan hasil nilai pengukuran metrik tertentu. Tabel ini memiliki hubungan *many-to-one* dengan tabel metrik dan tabel evaluasi.

- Tabel nilaiSub
Tabel ini menyimpan hasil nilai subkarakteristik tertentu. Tabel ini memiliki hubungan *many-to-one* dengan tabel evaluasi dan tabel subkarakteristik. Satu baris pada tabel ini merepresentasikan nilai satu subkarakteristik pada satu evaluasi.
- Tabel atribut
Tabel ini menyimpan data atribut dari sistem yang dievaluasi. Atribut tersebut dapat berupa nama fungsi, nama struktur data, nama entitas, ataupun nama komponen.
- Tabel jAtribut
Tabel ini merupakan tabel *junction* antara tabel atribut dan tabel param.
- Tabel param
Tabel ini menyimpan data parameter-parameter apa saja yang digunakan untuk melakukan perhitungan suatu metrik. Tabel ini memiliki hubungan *many-to-one* dengan tabel metrik.

3.2.2. Perancangan Diagram Kelas

Perancangan diagram kelas berisi rancangan dari kelas-kelas yang digunakan untuk membangun sistem. Aktivitas pada kaskas bantu evaluasi ini didominasi oleh komunikasi antara sistem dengan basis data. Sistem secara umum dibagi menjadi tiga paket kelas. Paket pertama yaitu paket Entity. Paket ini mencakupi kelas-kelas yang merepresentasikan entitas tabel pada basis data. Paket kedua yaitu paket Controller. Paket Controller berisi kelas-kelas yang mengatur komunikasi antara sistem dengan basis data. Pada paket ini juga terdapat kelas Helper yang mengatur logika perhitungan evaluasi. Paket terakhir yaitu paket GUI yang berisi kelas-kelas tampilan pada sistem ini. Diagram kelas sistem dapat dilihat pada Gambar 3.12.



Gambar 3.12 Diagram Kelas

3.2.3. Perancangan Antarmuka Pengguna

Bagian ini membahas rancangan tampilan antar muka pada sistem. Sistem memiliki empat antarmuka pengguna utama, yaitu halaman penambahan sistem, halaman penambahan pengguna, halaman evaluasi baru, dan halaman hasil evaluasi.

3.2.3.1. Halaman Penambahan Pengguna

Halaman ini digunakan oleh pengguna pada saat menjalankan kasus penggunaan UC-0001. Pada halaman ini terdapat formulir penambahan pengguna baru. Rancangan halaman penambahan pengguna dapat dilihat pada Gambar 3.13. Penjelasan mengenai atribut-atribut yang ada pada halaman ini dapat dilihat pada Tabel 3.11.

Tabel 3.11 Spesifikasi Atribut Antarmuka Halaman Penambahan Pengguna

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Textbox1	Textbox	Lokasi pengisian nama pengguna	String
2	Textbox2	Textbox	Lokasi pengisian nama lengkap pengguna	String
3	Passwordbox 1	Passwordbox	Lokasi pengisian kata sandi	String
4	Passwordbox 2	Passwordbox	Lokasi pengisian konfirmasi kata sandi	String
5	Combo1	Combobox	Menampilkan status pengguna	String
6	Textbox3	Textbox	Lokasi pengisian alamat <i>email</i> pengguna	String

Pegguna Baru

Nama pengguna (*)

Nama lengkap (*)

Kata sandi (*)

Konfirmasi kata sandi(*)

Status (*)

E-mail

Gambar 3.13 Rancangan Halaman Penambahan Pegguna

3.2.3.2. Halaman Penambahan Sistem

Halaman ini digunakan oleh pengguna pada saat menjalankan kasus penggunaan UC-0002. Halaman ini ditampilkan apabila pengguna memilih untuk menambahkan data sistem yang akan dievaluasi. Rancangan halaman penambahan sistem dapat dilihat pada Gambar 3.14. Penjelasan mengenai atribut-atribut yang ada pada halaman ini dapat dilihat pada Tabel 3.12.

Tabel 3.12 Spesifikasi Atribut Antarmuka Halaman Penambahan Sistem

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Textbox1	Textbox	Lokasi pengisian nama sistem	String
2	Textbox2	Textbox	Lokasi pengisian versi sistem	String
3	Textbox3	Textbox	Lokasi pengisian	Digit

Tabel 3.12 Spesifikasi Atribut Antarmuka Halaman Penambahan Sistem (Lanjutan)

			tahun rilis sistem	
4	Textbox4	Textbox	Lokasi pengisian deskripsi sistem	String

Gambar 3.14 Rancangan Halaman Penambahan Sistem

3.2.3.3. Halaman Evaluasi Baru

Halaman ini digunakan oleh pengguna pada saat menjalankan kasus penggunaan UC-0003. Pada halaman ini pengguna dapat memilih sistem yang akan dievaluasi. Pengguna diharuskan mengisi formulir evaluasi baru apabila ingin memulai melakukan evaluasi. Rancangan halaman evaluasi baru dapat dilihat pada Gambar 3.15. Penjelasan mengenai atribut-atribut yang ada pada halaman ini dapat dilihat pada Tabel 3.13. Setelah pengguna memilih untuk melakukan evaluasi baru, kakas bantu akan menampilkan halaman pengisian atribut. Rancangan halaman pengisian atribut dapat dilihat pada Gambar 3.17. Penjelasan mengenai atribut yang ada pada halaman pengisian atribut dapat dilihat pada Tabel 3.15. Bila atribut sistem sudah diisi oleh pengguna, maka kakas bantu akan menampilkan pertanyaan-pertanyaan yang berkaitan dengan portabilitas. Daftar pertanyaan yang ditampilkan oleh kakas bantu dapat dilihat pada Gambar 3.16, sedangkan contoh rancangan halaman pertanyaan dapat dilihat pada Gambar 3.18. Sedangkan penjelasan mengenai atribut yang ada pada halaman pertanyaan dapat dilihat pada Tabel 3.15.

Kode Sistem	Nama Sistem	Versi	Tahun Rilis	Deskripsi
SIS0000001	Sistem 1	1.0	2014	sistem baru

Sistem (*)	SIS0000001
Tanggal evaluasi	06/25/2014
Tahap SDLC (*) (Siklus Hidup Pengembangan Sistem)	Pemasangan ▼
Evaluator	evaluator
<input type="button" value="Evaluasi"/>	

Gambar 3.15 Rancangan Halaman Evaluasi Baru

Tabel 3.13 Spesifikasi Atribut Antarmuka Halaman Evaluasi Baru

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	TabelSistem	Table	Menampilkan daftar sistem	String
2	Textbox1	Textbox	Lokasi pengisian kode sistem yang akan dievaluasi	String
3	Textbox2	Textbox	Lokasi pengisian tanggal evaluasi	String
4	Combo1	Combo box	Menampilkan tahapan siklus hidup pengembangan perangkat lunak	String
5	Textbox3	Textbox	Lokasi pengisian evaluator	String
6	Button1	Button	Pilihan aksi untuk memulai evaluasi	Event clicked

1. Apakah sumber data pada sistem memiliki kebutuhan untuk dapat beradaptasi ?
 Ya Tidak (sumber data dapat dipindahkan)

2. Apakah sistem memiliki kebutuhan untuk dapat beradaptasi pada berbagai lingkungan perangkat keras?
 Ya Tidak batasan perangkat keras (processor, graphic card, dll) dan/atau fasilitas jaringan

3. Apakah sistem memiliki kebutuhan untuk dapat beradaptasi pada berbagai lingkungan organisasi?
 Ya Tidak (infrastruktur organisasi)

4. Apakah sistem memiliki kebutuhan untuk dapat beradaptasi pada berbagai lingkungan perangkat lunak?
 Ya Tidak (Sistem operasi, perangkat lunak jaringan (ex:browser), perangkat lunak lain yang berkaitan)

5. Apakah sistem memiliki kebutuhan untuk memudahkan pengguna dalam melakukan pemindahan?
 Ya Tidak (Sistem dapat dipindahkan oleh pengguna)

6. Apakah pengguna perlu melakukan proses instalasi sebelum dapat menggunakan sistem?
 Ya Tidak

7. Apakah proses instalasi sistem terdiri dari beberapa langkah operasi?
 Ya Tidak

8. Apakah sistem memiliki kebutuhan untuk pengguna melakukan modifikasi pengaturan saat proses instalasi?
 Ya Tidak

9. Apakah sistem berjalan berdampingan dengan sistem lain?
 Ya Tidak

10. Apakah sistem versi ini merupakan pembaruan atau pengganti dari sistem versi sebelumnya?
 Ya Tidak

11. Apakah sistem memiliki batasan terhadap standar atau regulasi tertentu yang berkaitan dengan portabilitas?
 Ya Tidak

Gambar 3.16 Halaman Daftar Pertanyaan

<p>Kebutuhan fungsional, nonfungsional/ fitur-fitur sistem</p> <p><input type="text"/></p> <p><input type="button" value="Tambah"/></p> <p><input type="button" value="Hapus"/></p>		<p>Sumber data (tabel dalam basis data atau file penyimpanan data lain) yang digunakan sistem (ex: namaTabel, namaFile.txt)</p> <p><input type="text"/></p> <p><input type="button" value="Tambah"/></p> <p><input type="button" value="Hapus"/></p>	
<input type="button" value="Berikutnya"/>			

Gambar 3.17 Halaman Pengisian Atribut

Tabel 3.14 Spesifikasi Atribut Antarmuka Halaman Pengisian Atribut

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Textbox1	Textbox	Lokasi pengisian fitur sistem	String
2	Textbox2	Textbox	Lokasi pengisian sumber data sistem	String
3	ButtonTambah1	Button	Pilihan aksi untuk menambah fitur sistem	Event clicked
4	ButtonTambah2	Button	Pilihan aksi untuk menambah sumber data sistem	Event clicked
5	ButtonHapus1	Button	Pilihan aksi untuk menghapus fitur sistem	Event clicked
6	ButtonHapus2	Button	Pilihan aksi untuk menghapus sumber data sistem	Event clicked
7	ButtonNext	Button	Pilihan aksi untuk menyimpan data	Event clicked
8	TabelFitur	Table	Menampilkan daftar fitur	String
9	TabelSumberData	Table	Menampilkan daftar sumber data	String

Tabel 3.15 Spesifikasi Atribut Antarmuka Halaman Pertanyaan

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Textbox1	Textbox	Lokasi pengisian parameter	String
2	ButtonTambah1	Button	Pilihan aksi untuk menambah data ke TabelParamB	Event clicked
3	ButtonTambah2	Button	Pilihan aksi untuk menambah data ke TabelParamA	Event clicked
4	ButtonHapus1	Button	Pilihan aksi untuk menghapus data	Event clicked

Tabel 3.15. Spesifikasi Atribut Antarmuka Halaman
Pertanyaan (Lanjutan)

			dari TabelParamB	
5	ButtonHapus2	Button	Pilihan aksi untuk menghapus data dari TabelParamA	Event clicked
6	ButtonBack	Button	Pilihan aksi untuk kembali ke halaman pertanyaan	Event clicked
7	ButtonSimpan	Button	Pilihan aksi untuk menyimpan data	Event clicked
8	TabelParamB	Table	Menampilkan daftar parameter	String
9	TabelParamA	Table	Menampilkan daftar parameter	String

Kebutuhan operasi pengaturan saat proses instalasi (ex: lokasi pemasangan, memori yang digunakan)

Operasi

Operasi pengaturan yang dapat diulang

Operasi

6/12

Kembali Simpan

Gambar 3.18 Halaman Pertanyaan

3.2.3.4. Halaman Hasil Evaluasi

Halaman ini digunakan oleh pengguna pada saat menjalankan kasus penggunaan UC-0004. Pada halaman ini sistem menampilkan hasil evaluasi sistem yang telah dipilih. Rancangan halaman evaluasi baru dapat dilihat pada Gambar 3.19. Penjelasan mengenai atribut-atribut yang ada pada halaman ini dapat dilihat pada Tabel 3.16.

Tabel 3.16 Spesifikasi Atribut Antarmuka Halaman Hasil Evaluasi

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/Keluaran
1	TabelMetrik	Table	Menampilkan nilai masing-masing metrik	String
2	TabelSubkarakteristik	Table	Menampilkan nilai masing-masing subkarakteristik	String

Sistem	Sistem 1	Nilai Subkarakteristik		
Versi	1.0	Subkarakteristik	Nilai	Nilai Bobot
Kode Evaluasi	EVA0000001	Adaptasi	0.78	2.3
Tanggal Evaluasi	2014-06-25	Pemasangan	0.75	0.75
Tahap Evaluasi	Pemasangan	Keberdampingan	N/A	N/A
		Penggantian	N/A	N/A
Nilai Metrik		Bobot		
		Adaptasi sumber data	1.0	Sumber data sistem berhasil beradaptasi dengan baik
		Adaptasi lingkungan perangkat keras	0.50	Masih ada fitur sistem yang belum dapat beradaptasi pada lingkungan perangkat keras tertentu
		Adaptasi lingkungan organisasi	N/A	Tidak diperhitungkan
		Adaptasi lingkungan sistem perangkat lunak	0.83	Masih ada fitur sistem yang belum dapat beradaptasi pada lingkungan perangkat lunak tertentu
		Kemudahan untuk melakukan	N/A	Tidak diperhitungkan
Keterangan: N/A = aspek pengukuran tidak diperhitungkan nilai semakin baik apabila semakin mendekati 1				

Gambar 3.19 Rancangan Halaman Hasil Evaluasi

BAB IV IMPLEMENTASI

Bab ini membahas tentang implementasi dari perancangan sistem. Implementasi sistem mencakup proses implementasi metrik pengukuran karakteristik portabilitas ISO/IEC 9126 ke dalam sistem. Bahasa pemrograman yang digunakan adalah bahasa pemrograman Java.

4.1. Paket `HibernateClass`

Paket ini berisi kelas-kelas yang merepresentasikan setiap tabel pada basis data yang digunakan. Kelas yang termasuk ke dalam paket `HibernateClass` antara lain sebagai berikut.

4.1.1. Kelas `Atribut`

Kelas ini adalah representasi tabel `Atribut` pada basis data. Kelas ini merepresentasikan atribut dari perangkat lunak yang dievaluasi. Kelas `Atribut` memiliki beberapa properti, antara lain adalah `idatribut`, `atribut`, `satuan`, `jatribut`, dan `evaluasi`. Kelas ini mengimplementasi fungsi-fungsi `getter` dan `setter` bagi tiap properti. Contoh fungsi `getter` dan `setter` dapat dilihat pada Kode Sumber 4.1.

1	<code>public String getIdatribut() {</code>
2	<code> return this.idatribut;</code>
3	<code>}</code>
4	
5	<code>public void setIdatribut(String idatribut) {</code>
6	<code> this.idatribut = idatribut;</code>
7	<code>}</code>

Kode Sumber 4.1. Fungsi *Getter* dan *Setter* Properti `idatribut`

4.1.2. Kelas `Evaluasi`

Kelas ini merupakan representasi tabel `Evaluasi` pada basis data. Kelas `Evaluasi` merepresentasikan evaluasi perangkat lunak. Kelas ini memiliki beberapa properti, antara lain adalah

idevaluasi, tanggalevaluasi, tahapsdlc, nilai, pengguna, sistem, penanda, nilaiMetrik, atribut, dan nilaiSub. Kelas ini mengimplementasi fungsi-fungsi *getter* dan *setter* bagi tiap properti.

4.1.3. Kelas Metrik

Kelas Metrik adalah representasi tabel Metrik pada basis data. Kelas ini merepresentasikan tiap metrik pada karakteristik portabilitas ISO/IEC 9126. Kelas Metrik memiliki beberapa properti, antara lain adalah idmetrik, namametrik, subkarakteristik, param, dan nilaiMetrik. Kelas ini mengimplementasi fungsi-fungsi *getter* dan *setter* bagi tiap properti.

4.1.4. Kelas Nilaimetrik

Kelas Nilaimetrik adalah representasi tabel Nilaimetrik pada basis data. Kelas ini menyimpan hasil perhitungan tiap metrik pada perangkat lunak yang dievaluasi. Kelas Nilaimetrik memiliki beberapa properti, antara lain adalah idnilaimetrik, nilaimetrik, metrik, dan evaluasi. Kelas ini mengimplementasi fungsi-fungsi *getter* dan *setter* bagi tiap properti.

4.1.5. Kelas Nilaisub

Kelas Nilaisub merupakan representasi tabel Nilaisub pada basis data. Kelas ini merepresentasikan hasil perhitungan tiap subkarakteristik pada perangkat lunak yang dievaluasi. Kelas Nilaisub memiliki beberapa properti, antara lain adalah idnilaisub, nilaisub, subkarakteristik, dan evaluasi. Kelas ini mengimplementasi fungsi-fungsi *getter* dan *setter* bagi tiap properti.

4.1.6. Kelas Param

Kelas Param merupakan representasi tabel Param pada basis data. Kelas ini merepresentasikan parameter perhitungan tiap

metrik pada karakteristik portabilitas ISO/IEC 9126. Kelas Param memiliki beberapa properti, antara lain adalah *idparam*, *namaparam*, *satuan*, *jatribut*, dan *metrik*. Kelas ini mengimplementasi fungsi-fungsi *getter* dan *setter* bagi tiap properti.

4.1.7. Kelas Pengguna

Kelas Pengguna adalah representasi tabel Pengguna pada basis data. Kelas ini merepresentasikan pengguna kakas bantu evaluasi. Kelas Pengguna memiliki beberapa properti, antara lain adalah *username*, *password*, *status*, *email*, *namalengkap*, *evaluasi*, *jsistem*, dan *aktif*. Kelas ini mengimplementasi fungsi-fungsi *getter* dan *setter* bagi tiap properti

4.1.8. Kelas Sistem

Kelas Sistem adalah representasi tabel Sistem pada basis data. Kelas ini merepresentasikan sistem yang dievaluasi. Kelas Sistem memiliki beberapa properti, antara lain adalah *idsistem*, *namasistem*, *versi*, *deskripsi*, *tahunrilis*, dan *evaluasi*. Kelas ini mengimplementasi fungsi-fungsi *getter* dan *setter* bagi tiap properti.

4.1.9. Kelas Subkarakteristik

Kelas Subkarakteristik merupakan representasi tabel Subkarakteristik pada basis data. Kelas Subkarakteristik merepresentasikan tiap subkarakteristik dari karakteristik portabilitas ISO/IEC 9126. Kelas Subkarakteristik memiliki beberapa properti, antara lain adalah *idkarakteristik*, *namasub*, *bobotsub*, *metrik*, dan *nilaiSub*. Kelas ini mengimplementasi fungsi-fungsi *getter* dan *setter* bagi tiap properti.

4.2. Paket HibernateController

Paket ini berisi kelas-kelas yang menghubungkan kakas bantu dengan basis data. Paket ini juga berisi logika perhitungan metrik-metrik dan subkarakteristik-subkarakteristik yang ada pada karakteristik portabilitas ISO/IEC 9126. Kelas-kelas yang termasuk ke dalam paket HibernateController antara lain adalah sebagai berikut.

4.2.1. Kelas AtributController

Kelas ini mengatur komunikasi antara kakas bantu dengan tabel Atribut pada basis data. Kelas ini mengimplementasi fungsi-fungsi yang memanfaatkan kerangka kerja Hibernate dalam melakukan komunikasi. Fungsi yang ada pada kelas ini antara lain adalah fungsi insertAtribut, updateAtribut, selectAtribut, dan selectAtributById. Fungsi insertAtribut berfungsi untuk menyimpan data atribut ke dalam basis data. Fungsi insertAtribut dapat dilihat pada Kode Sumber 4.2. Fungsi updateAtribut berfungsi untuk memperbarui data atribut pada basis data. Fungsi updateAtribut dapat dilihat pada Kode Sumber 4.3. Fungsi selectAtribut berfungsi untuk mengambil seluruh data atribut pada tabel basis data. Fungsi selectAtribut dapat dilihat pada Kode Sumber 4.4. Sedangkan fungsi selectAtributById berfungsi untuk mengambil data atribut tertentu sesuai dengan kode ID dari atribut tersebut. Fungsi selectAtributById dapat dilihat pada Kode Sumber 4.5.

1	String errorMess ="";
2	org.hibernate.Transaction tx = null;
3	try{
4	tx = session.beginTransaction();
5	Query q=session.createQuery("from Evaluasi where idevaluasi=:id");
6	q.setParameter("id", idEval);
7	Evaluasi evaluasi = (Evaluasi)q.list().get(0);
8	atribut.setEvaluasi(evaluasi);
9	evaluasi.getAtribut().add(atribut);...

```

10    ... session.update(evaluasi);
11    session.save(atribut);
12    session.flush();
13    tx.commit();
14    errorMess = "Insertion success";
15    }catch(RuntimeException e){
16        try
17        {
18            tx.rollback();
19            errorMess = "Insertion failed, transaction rolled back";
20        }catch(RuntimeException rbe){
21            errorMess = "Couldn't roll back transaction";
22        }
23        throw e;
24    }finally{
25        if(session!=null) session.close();
26    }

```

Kode Sumber 4.2. Fungsi insertAtribut

```

1    String errorMess = "";
2    org.hibernate.Transaction tx = null;
3    try{
4        tx = session.beginTransaction();
5        Query q = session.createQuery("from Atribut where idatribut = :id ");
6        q.setParameter("id", idAtribut);
7        Atribut atribut = (Atribut)q.list().get(0);
8        atribut.setAtribut(atributBaru);
9        session.update(atribut);
10    session.flush();
11    tx.commit();
12    errorMess = "Data updated";
13    }catch(RuntimeException e){
14        try{
15            tx.rollback();
16            errorMess = "Update failed, transaction rolled back";
17        }catch(RuntimeException rbe) {
18            errorMess = "Couldn't roll back transaction";}
19        throw e;
20    }finally{
21        if(session!=null) session.close();}

```

Kode Sumber 4.3. Fungsi updateAtribut

1	<code>public List<Atribut> selectAtribut(){</code>
2	<code> org.hibernate.Transaction tx = session.beginTransaction();</code>
3	<code> Query query = session.createQuery("from Atribut");</code>
4	<code> List<Atribut> listAtribut = query.list();</code>
5	<code> return listAtribut;</code>
6	<code>}</code>

Kode Sumber 4.4. Fungsi `selectAtribut`

1	<code>public Atribut selectAtributById(String id){</code>
2	<code> Atribut atribut;</code>
3	<code> org.hibernate.Transaction tx = session.beginTransaction();</code>
4	<code> Query query = session.createQuery("from Atribut where idatribut=</code>
5	<code> :name");</code>
6	<code> query.setParameter("name", id);</code>
7	<code> List<Atribut> listAtribut = query.list();</code>
8	<code> if(listAtribut.isEmpty()) return null;</code>
9	<code> atribut = (Atribut) listAtribut.get(0);</code>
10	<code> return atribut;</code>
11	<code>}</code>

Kode Sumber 4.5. Fungsi `selectAtributById`

4.2.2. Kelas `EmailValidator`

Kelas ini mengimplementasi fungsi-fungsi untuk validasi *String*. Kelas `EmailValidator` memiliki atribut yang berisi pola *regex* untuk validasi format tertentu. Fungsi yang ada pada kelas ini adalah `validate` yang ditunjukkan oleh Kode Sumber 4.6.

1	<code>public boolean validate(final String input) {</code>
2	<code> matcher = pattern.matcher(input);</code>
3	<code> return matcher.matches();</code>
4	<code>}</code>

Kode Sumber 4.6. Fungsi `validate`

4.2.3. Kelas `EvaluasiController`

Kelas ini mengatur komunikasi antara kakas bantu dengan tabel `Evaluasi` pada basis data. Kelas ini mengimplementasi fungsi-fungsi yang memanfaatkan kerangka kerja Hibernate dalam

melakukan komunikasi. Fungsi-fungsi dalam kelas ini digunakan untuk menambahkan data evaluasi baru ke dalam basis data, melakukan pembaruan data evaluasi, mengambil semua data evaluasi yang tersimpan dalam basis data, dan mengambil data evaluasi berdasarkan kode penyimpanan.

4.2.4. Kelas Helper

Kelas ini merupakan kelas yang mengimplementasi fungsi perhitungan baik metrik, subkarakteristik, maupun karakteristik berdasarkan ISO/IEC 9126. Fungsi yang ada pada kelas ini antara lain adalah fungsi `hitungMetrik`, dan fungsi `hitungSubkarakteristik`. Fungsi `hitungMetrik` merupakan fungsi untuk menghitung nilai dari metrik sesuai dengan parameter yang dimasukkan. Kode Sumber 4.7 menunjukkan fungsi `hitungMetrik`. Fungsi `hitungSubkarakteristik` adalah fungsi untuk menghitung nilai dari subkarakteristik sesuai dengan metrik-metrik yang termasuk di dalamnya. Fungsi `hitungSubkarakteristik` dapat dilihat pada Kode Sumber 4.8.

1	<code>public double hitungMetrik(String idevaluasi, String satuan, String</code>
2	<code>idmetrik, String idparamA, String idparamB)</code>
3	<code>{ AtributController ac = new AtributController();</code>
4	<code> JatributController jc = new JatributController();</code>
5	<code> List<Atribut> la = ac.selectAtributByEvalSatuan(idevaluasi,</code>
6	<code> satuan);</code>
7	<code> List<Jatribut> ljB = jc.selectJatributByParam(idparamB);</code>
8	<code> List<Jatribut> ljA = jc.selectJatributByParam(idparamA);</code>
9	<code> int countA=0;</code>
10	<code> int countB=0;</code>
11	<code> for(int a=0;a<la.size();a++){</code>
12	<code> for(int b=0;b<ljB.size();b++){</code>
13	<code> if(ljB.get(b).getAtribut().getIdatribut().equals(la.get(a).getIdatribut())){</code>
14	<code> countB++;</code>
15	<code> }</code>
16	<code> }</code>
17	<code> for(int c=0;c<ljA.size();c++){</code>
18	<code> if(ljB.get(c).getAtribut().getIdatribut().equals(la.get(c).getIdatribut()))...</code>

```

19  ...{
20      countA++;
21  }
22  }
23  }
24  double nilaiMetrik = (double) countA/countB;
25  return nilaiMetrik;
    }

```

Kode Sumber 4.7. Fungsi hitungMetrik

```

1  public double hitungSubkarakteristik(String idevaluasi, String
2  subkarakteristik){
3      NilaiMetrikController nmc = new NilaiMetrikController();
4      List<Nilaimetrik> lnm =
5      nmc.selectNilaimetrikByEvalSub(idevaluasi, subkarakteristik);
6      int count =0;
7      double akum =0;
8      for(int a=0;a<lnm.size();a++){
9          if(lnm.get(a).getNilaimetrik()!=-1.0){
10             akum = akum+lnm.get(a).getNilaimetrik();
11             count++;
12         }
13     }
14     double nilaiSub;
15     if(count==0)
16     {nilaiSub=-1.0;}
17     else nilaiSub = akum/count;
18     return nilaiSub;
19 }

```

Kode Sumber 4.8. Fungsi hitungSubkarakteristik

4.2.5. Kelas **MetrikController**

Kelas ini mengatur komunikasi antara kakas bantu dengan tabel **Metrik** pada basis data. Kelas ini mengimplementasi fungsi-fungsi yang memanfaatkan kerangka kerja Hibernate dalam melakukan komunikasi. Fungsi-fungsi dalam kelas ini digunakan untuk mengambil semua data metrik yang tersimpan dalam basis data, dan mengambil data metrik berdasarkan kode penyimpanan atau nama metrik.

4.2.6. Kelas NilaiMetrikController

Kelas ini mengatur komunikasi antara kakas bantu dengan tabel `NilaiMetrik` pada basis data. Kelas ini mengimplementasi fungsi-fungsi yang memanfaatkan kerangka kerja Hibernate dalam melakukan komunikasi. Fungsi-fungsi dalam kelas ini digunakan untuk menambahkan data hasil perhitungan metrik ke dalam basis data, melakukan pembaruan data, dan mengambil data nilai metrik pada satu evaluasi.

4.2.7. Kelas NilaiSubController

Kelas ini mengatur komunikasi antara kakas bantu dengan tabel `NilaiSubkarakteristik` pada basis data. Kelas ini mengimplementasi fungsi-fungsi yang memanfaatkan kerangka kerja Hibernate dalam melakukan komunikasi. Fungsi-fungsi dalam kelas ini digunakan untuk menambahkan data hasil perhitungan tiap subkarakteristik ke dalam basis data, melakukan pembaruan data, dan mengambil data nilai subkarakteristik pada satu evaluasi.

4.2.8. Kelas ParamController

Kelas ini mengatur komunikasi antara kakas bantu dengan tabel `Param` pada basis data. Kelas ini mengimplementasi fungsi-fungsi yang memanfaatkan kerangka kerja Hibernate dalam melakukan komunikasi. Fungsi-fungsi dalam kelas ini digunakan untuk mengambil semua data parameter perhitungan metrik yang tersimpan dalam basis data, dan mengambil data parameter berdasarkan kode penyimpanan.

4.2.9. Kelas PenggunaController

Kelas ini mengatur komunikasi antara kakas bantu dengan tabel `Pengguna` pada basis data. Kelas ini mengimplementasi fungsi-fungsi yang memanfaatkan kerangka kerja Hibernate dalam melakukan komunikasi. Fungsi-fungsi dalam kelas ini digunakan untuk menambahkan data pengguna baru ke dalam basis data, melakukan pembaruan data pengguna, mengambil semua data pengguna yang tersimpan dalam basis data, validasi nama pengguna

dan kata sandi, dan mengambil data pengguna berdasarkan nama pengguna.

4.2.10. Kelas `SystemController`

Kelas ini mengatur komunikasi antara kakas bantu dengan tabel `System` pada basis data. Kelas ini mengimplementasi fungsi-fungsi yang memanfaatkan kerangka kerja Hibernate dalam melakukan komunikasi. Fungsi-fungsi dalam kelas ini digunakan untuk menambahkan data sistem baru ke dalam basis data, melakukan pembaruan data sistem, mengambil semua data sistem yang tersimpan dalam basis data, dan mengambil data sistem berdasarkan kode penyimpanan.

4.2.11. Kelas `SubkarakteristikController`

Kelas ini mengatur komunikasi antara kakas bantu dengan tabel `Subkarakteristik` pada basis data. Kelas ini mengimplementasi fungsi-fungsi yang memanfaatkan kerangka kerja Hibernate dalam melakukan komunikasi. Fungsi-fungsi dalam kelas ini digunakan untuk mengambil semua data subkarakteristik yang tersimpan dalam basis data dan mengambil data subkarakteristik berdasarkan kode penyimpanan atau nama subkarakteristik.

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada kakas bantu yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan kegunaan sistem. Pengujian fungsionalitas mengacu pada kasus penggunaan yang ada pada bab tiga.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan kakas sebagai berikut.

Prosesor	: Intel Core i7 3632QM CPU @ 2.20GHz
Memori	: 12.00 GB
Jenis Device	: Laptop
Sistem Operasi	: Microsoft Windows 8 Basic 64 bit
JDK	: JDK 7

5.2. Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan terhadap kakas bantu. Pengujian yang dilakukan adalah pengujian kebutuhan fungsionalitas dan pengujian kebergunaan kakas bantu. Pengujian fungsionalitas menggunakan metode kotak hitam (*black box*). Metode ini menekankan pada kesesuaian hasil keluaran kakas bantu.

Pengujian terhadap hasil pengukuran kualitas dilakukan dengan membandingkan hasil pengukuran kualitas sistem dengan hasil pengukuran kualitas manual yang dilakukan oleh pengembang kakas bantu. Pengembang kakas bantu melakukan evaluasi terhadap SIAKAD ITS Modul Penilaian dan melakukan perhitungan nilai karakteristik portabilitas sesuai dengan ISO/IEC 9126. Hasil yang didapatkan secara manual tersebut kemudian dibandingkan dengan hasil sistem.

5.2.1. Pengujian Fungsionalitas

Pengujian fungsionalitas kakas bantu dilakukan dengan mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.1.2. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

5.2.1.1. Pengujian Fitur Menambah Pengguna

Pengujian fitur menambah pengguna dilakukan dengan memasukkan data pengguna baru dan menyimpan ke dalam basis data. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pengujian Fitur Menambah Pengguna

ID	UJ.UC-0001
Referensi Kasus Penggunaan	UC-0001
Nama	Pengujian fitur menambah pengguna
Tujuan Pengujian	Menguji fitur menambah pengguna
Skenario	Administrator menambahkan data pengguna baru sesuai kebutuhan
Kondisi Awal	Data uji telah siap dan pengguna sudah masuk ke dalam antarmuka menambah pengguna baru
Data Uji	Data pengembang kakas bantu
Langkah Pengujian	Pengguna mengisi formulir data pengguna baru
Hasil Yang Diharapkan	Data pengguna baru telah tersimpan dan ditampilkan pada antarmuka tabel pengguna
Hasil Yang Didapat	Antarmuka tabel pengguna menampilkan data pengguna yang telah disimpan
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan hasil evaluasi dapat dilihat pada Gambar 5.1

Nama Pengg...	Nama Lengkap	Status	E-mail	
developer		Administrator		aktif
evaluator	evaluator	Evaluator		aktif
klien	klien	Klien		aktif
istiningdyah	Istiningdyah Saptarini	Evaluator	istiningdyah10 @mhs.if.its.ac.id	aktif

Gambar 5.1 Hasil Uji Menambah Pengguna

5.2.1.2. Pengujian Fitur Menambah Sistem

Pengujian fitur menambah sistem dilakukan dengan memasukkan data sistem baru dan menyimpan ke dalam basis data. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.2.

Tabel 5.2 Pengujian Fitur Menambah Sistem

ID	UJ.UC-0002
Referensi Kasus Penggunaan	UC-0002
Nama	Pengujian fitur menambah sistem
Tujuan Pengujian	Menguji fitur menambah sistem
Skenario	Sistem yang menjadi studi kasus adalah SIAKAD ITS modul penilaian
Kondisi Awal	Data uji telah siap dan pengguna sudah masuk ke dalam antarmuka menambah sistem baru
Data Uji	Data SIAKAD ITS modul penilaian
Langkah Pengujian	Pengguna mengisi formulir data sistem baru
Hasil Yang Diharapkan	Data sistem baru telah tersimpan dan ditampilkan pada antarmuka tabel sistem
Hasil Yang Didapat	Antarmuka tabel sistem menampilkan data sistem yang telah disimpan
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan hasil evaluasi dapat dilihat pada Gambar 5.2

Kode Sistem	Nama Sistem	Versi	Tahun Rilis	Deskripsi
SIS0000001	Sistem 1	1.0	2014	sistem baru
SIS0000002	SIAKAD ITS modul penilaian	-	2009	Sistem Informasi Akademik ITS modul penilaian

Gambar 5.2 Hasil Uji Menambah Sistem

5.2.1.3. Pengujian Fitur Melakukan Pengukuran Kualitas

Pengujian fitur melakukan pengukuran kualitas dilakukan dengan melakukan evaluasi terhadap SIAKAD ITS Modul Penilaian. Data masukan kaku bantu diperoleh dengan melakukan penggalan kebutuhan pengguna serta peninjauan SIAKAD ITS Modul Penilaian secara langsung. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.3.

Tabel 5.3 Pengujian Fitur Melakukan Pengukuran Kualitas

ID	UJ.UC-0003
Referensi Kasus Penggunaan	UC-0003
Nama	Pengujian fitur melakukan pengukuran kualitas
Tujuan Pengujian	Menguji fitur melakukan pengukuran kualitas
Skenario	Sistem yang menjadi studi kasus adalah SIAKAD ITS modul penilaian
Kondisi Awal	Data sistem yang diuji telah tersimpan dalam basis data dan telah dipilih
Data Uji	Data uji diperoleh dari hasil penggalan kebutuhan dan peninjauan sistem
Langkah Pengujian	Pengguna mengisi formulir pengujian sesuai dengan data uji
Hasil Yang Diharapkan	Hasil perhitungan metrik dan subkarakteristik yang sesuai dengan sistem yang dievaluasi
Hasil Yang Didapat	Nilai metrik dan subkarakteristik dari SIAKAD ITS modul penilaian
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan hasil evaluasi dapat dilihat pada Gambar 5.3

Sistem	SIAKAD ITS modul penilaian	Nilai Subkarakteristik			
Versi	-	Subkarakteristik	Nilai	Nilai Bobot	Bobot
Kode Evaluasi	EVA0000002	Adaptasi	1.0	3.0	3.0
Tanggal Evaluasi	2014-06-25	Pemasangan	N/A	N/A	1.0
Tahap Evaluasi	Pemeliharaan	Kebardampingan	1.0	3.0	3.0
		Penggantian	N/A	N/A	2.0
Nilai Metrik					
		Metrik	Nilai	Keterangan	
		Adaptasi sumber data	N/A	Tidak diperhitungkan	
		Adaptasi lingkungan perangkat keras	N/A	Tidak diperhitungkan	
		Adaptasi lingkungan organisasi	1.0	Sistem dapat beradaptasi pada lingkungan organisasi yang ditentukan	
		Adaptasi lingkungan sistem perangkat lunak	1.0	Sistem dapat beradaptasi pada lingkungan perangkat lunak yang ditentukan	
		Kemudahan untuk melakukan	N/A	Tidak diperhitungkan	
		Kemudahan percobaan ulang	N/A	Tidak diperhitungkan	
Keterangan:					
N/A = aspek pengukuran tidak diperhitungkan					
nilai semakin baik apabila semakin mendekati 1					

Gambar 5.3. Tampilan Hasil Pengujian

5.2.1.4. Pengujian Fitur Melihat Hasil Pengukuran Kualitas

Pengujian fitur melihat hasil pengukuran kualitas dilakukan dengan melakukan pengukuran kualitas terhadap SIAKAD ITS Modul Penilaian terlebih dahulu. Hasil pengukuran kemudian disimpan ke dalam basis data. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.4.

Tabel 5.4 Pengujian Fitur Melihat Hasil Pengukuran Kualitas

ID	UJ.UC-0004
Referensi Kasus Penggunaan	UC-0004
Nama	Pengujian fitur melihat hasil pengukuran kualitas
Tujuan Pengujian	Menguji fitur melihat hasil pengukuran kualitas

Tabel 5.4 Pengujian Fitur Melihat Hasil Pengukuran Kualitas (Lanjutan)

Skenario	Sistem yang menjadi studi kasus adalah SIAKAD ITS modul penilaian
Kondisi Awal	Pengukuran kualitas sistem sudah dilakukan dan data telah tersimpan dalam basis data
Data Uji	Data uji telah tersimpan di dalam basis data
Langkah Pengujian	Pengguna memilih untuk menampilkan hasil pengukuran kualitas sistem
Hasil Yang Diharapkan	Hasil perhitungan metrik dan subkarakteristik dari sistem yang telah dievaluasi sebelumnya
Hasil Yang Didapat	Nilai metrik dan subkarakteristik dari SIAKAD ITS modul penilaian
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan hasil evaluasi dapat dilihat pada Gambar 5.4

Sistem	SIAKAD ITS modul penilaian	Nilai Subkarakteristik			
Versi	-	Subkarakteristik	Nilai	Nilai Bobot	Bobot
Kode Evaluasi	EVA0000002	Adaptasi	1.0	3.0	3.0
Tanggal Evaluasi	2014-06-25	Pemasangan	N/A	N/A	1.0
Tahap Evaluasi	Pemeliharaan	Keberdampingan	1.0	3.0	3.0
		Penggantian	N/A	N/A	2.0
Nilai Metrik					
		Metrik	Nilai	Keterangan	
		Adaptasi sumber data	N/A	Tidak diperhitungkan	
		Adaptasi lingkungan perangkat keras	N/A	Tidak diperhitungkan	
		Adaptasi lingkungan organisasi	1.0	Sistem dapat beradaptasi pada lingkungan organisasi yang ditentukan	
		Adaptasi lingkungan sistem perangkat lunak	1.0	Sistem dapat beradaptasi pada lingkungan perangkat lunak yang ditentukan	
		Kemudahan untuk melakukan	N/A	Tidak diperhitungkan	
		Kemudahan percobaan ulang	N/A	Tidak diperhitungkan	
Keterangan:					
	N/A = aspek pengukuran tidak diperhitungkan				
	nilai semakin baik apabila semakin mendekati 1				

Gambar 5.4 Hasil Uji Melihat Hasil Pengukuran Kualitas

5.2.2. Pengujian Kebergunaan

Pengujian kebergunaan kakas bantu dilakukan dengan metode kuesioner. Terdapat 10 responden yang terdiri dari pengembang berbagai sistem. Responden menguji kakas bantu

dengan mengukur kualitas sistem tertentu. Kuesioner dapat dilihat pada Lampiran B. Sedangkan hasil pengujian dijelaskan pada subbab 5.3.3.

5.3. Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional dan evaluasi hasil perhitungan metrik dan subkarakteristik.

5.3.1. Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.5. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dilakukan dan kakas bantu dapat berjalan dengan baik. Sehingga bisa ditarik disimpulkan bahwa fungsionalitas dari kakas bantu telah dapat bekerja sesuai dengan yang diharapkan.

Tabel 5.5 Rangkuman Hasil Pengujian

ID	Nama	Hasil
UJ.UC-0001	Pengujian fitur menambah pengguna	Berhasil
UJ.UC-0002	Pengujian fitur menambah sistem	Berhasil
UJ.UC-0003	Pengujian fitur melakukan pengukuran kualitas	Berhasil
UJ.UC-0004	Pengujian fitur melihat hasil pengukuran kualitas	Berhasil

5.3.2. Evaluasi Hasil Perhitungan

Untuk mengukur kesesuaian keluaran kakas bantu maka dilakukan perbandingan antara hasil perhitungan metrik kualitas menggunakan kakas bantu dengan hasil perhitungan metrik secara manual. Sistem yang diukur adalah SIAKAD ITS Modul Penilaian. Berdasarkan penggalian data kebutuhan dan peninjauan, ada tiga metrik yang digunakan untuk mengukur kualitas SIAKAD ITS Modul Penilaian. Hal tersebut dikarenakan adanya penyesuaian

antara kondisi sistem dengan tujuan masing-masing metrik yang ada pada karakteristik portabilitas ISO/IEC 9126.

Metrik pertama yang digunakan adalah metrik adaptasi lingkungan organisasi. SIAKAD ITS Modul Penilaian memiliki kebutuhan untuk dapat beradaptasi di berbagai struktur lingkungan organisasi seperti administrasi, ketua jurusan, dosen, dan mahasiswa. Berdasarkan hasil peninjauan, 17 fitur yang ada pada SIAKAD ITS Modul Penilaian mampu beradaptasi dengan baik pada lingkungan organisasi yang telah ditentukan.

Metrik kedua adalah metrik adaptasi lingkungan sistem perangkat lunak. SIAKAD ITS Modul Penilaian merupakan aplikasi berbasis *web* yang hanya bisa diakses menggunakan *browser*. SIAKAD ITS Modul Penilaian memiliki kebutuhan untuk dapat diakses oleh *browser* yang umum digunakan seperti Mozilla Firefox, Google Chrome, dan Internet Explorer. Berdasarkan hasil peninjauan, 17 fitur yang ada pada SIAKAD ITS Modul Penilaian mampu diakses oleh *browser* yang telah ditentukan.

Metrik ketiga adalah metrik ketersediaan keberdampingan. SIAKAD ITS Modul Penilaian mengakses sumber daya operasional dalam hal ini server basis data yang sama dengan SIAKAD ITS modul lain. SIAKAD ITS secara keseluruhan menggunakan tiga server basis data. SIAKAD ITS Modul Penilaian berhasil berdampingan dengan modul lain sehingga pengukuran metrik ini menghasilkan nilai yang bagus.

Hasil perhitungan metrik kualitas dengan menggunakan kakas bantu dapat dilihat pada Gambar 5.5. Sedangkan hasil perhitungan metrik kualitas secara manual dapat dilihat pada Tabel 5.5. Perbandingan antara hasil perhitungan dapat dilihat pada Tabel 5.6. Untuk hasil perhitungan subkarakteristik menggunakan kakas bantu dapat dilihat pada Gambar 5.6. Kolom bobot menunjukkan nilai maksimal dari subkarakteristik. Hasil perhitungan subkarakteristik secara manual dapat dilihat pada Tabel 5.8. Sedangkan perbandingan hasil perhitungan subkarakteristik dapat dilihat pada Tabel 5.9.

Metrik	Nilai	Keterangan
Adaptasi lingkungan organisasi	1.0	Sistem dapat beradaptasi pada lingkungan organisasi yang ditentukan
Adaptasi lingkungan sistem perangkat lunak	1.0	Sistem dapat beradaptasi pada lingkungan perangkat lunak yang ditentukan
Ketersediaan keberdampingan	1.0	Sistem dapat berdampingan dengan sistem lain dengan baik

Gambar 5.5 Hasil Pengukuran Kualitas dengan Kakas Bantu

Tabel 5.6 Hasil Pengukuran Kualitas secara Manual

Metrik yang diuji	Formula	Parameter A	Parameter B	Hasil
Adaptasi lingkungan organisasi	$X = \frac{A}{B}$	17	17	1
Adaptasi lingkungan sistem perangkat lunak	$X = \frac{A}{B}$	17	17	1
Ketersediaan keberdampingan	$X = \frac{A}{B}$	3	3	1

Tabel 5.7 Daftar Perbandingan Hasil Pengujian

Metrik yang diuji	Hasil pengukuran kakas bantu	Hasil pengukuran manual
Adaptasi lingkungan organisasi	1.0	1
Adaptasi lingkungan sistem perangkat lunak	1.0	1
Ketersediaan keberdampingan	1.0	1

Subkarakteristik	Nilai	Nilai Bobot	Bobot
Adaptasi	1.0	3.0	3.0
Keberdampingan	1.0	3.0	3.0

Gambar 5.6 Hasil Pengukuran Subkarakteristik dengan Kakas Bantu

Tabel 5.8 Hasil Pengukuran Subkarakteristik secara Manual

Subkarakteristik	Metrik yang digunakan	Nilai subkarakteristik	Bobot	Nilai bobot
Adaptasi	2	1	3	3
Keberdampingan	1	1	3	3

Tabel 5.9 Daftar Perbandingan Hasil Pengujian Subkarakteristik

Subkarakteristik	Nilai subkarakteristik menggunakan kakas bantu	Nilai subkarakteristik secara manual	Nilai bobot subkarakteristik menggunakan kakas bantu	Nilai bobot subkarakteristik secara manual
Adaptasi	1.0	1	3.0	3
Keberdampingan	1.0	1	3.0	3

5.3.3. Evaluasi Pengujian Kebergunaan

Dalam pengujian kebergunaan, terdapat beberapa poin yang dinilai. Setiap poin memiliki rentang nilai 1-6 dimana nilai satu adalah nilai terendah dan enam adalah nilai tertinggi. Hasil pengujian kebergunaan dapat dilihat pada Tabel 5.10. Berdasarkan poin-poin yang dinilai dapat dilihat bahwa kakas bantu memiliki nilai kebergunaan sebesar 4,03. Dari beberapa poin yang diuji, kakas bantu memiliki nilai rata-rata tertinggi poin kemudahan kakas bantu dalam melakukan evaluasi yaitu sebesar 4,7. Nilai tersebut menunjukkan bahwa kebergunaan kakas bantu cukup baik karena berada pada di atas 50% dari total nilai.

Tabel 5.10 Hasil Pengujian Kebergunaan

No	Poin yang dinilai	Nilai						Total	Rata-rata
		1	2	3	4	5	6		
1	Kemudahan penggunaan kakas bantu	0	1	1	2	6	0	10	4,3
2	Kemudahan fitur untuk dipahami	0	1	5	3	0	1	10	3,5
3	Kemudahan antarmuka kakas bantu untuk dipahami	0	0	2	4	3	1	10	4,3
4	Kejelasan pesan pada kakas bantu	0	1	2	3	4	0	10	4
5	Daya tarik antarmuka kakas bantu	0	1	5	3	1	0	10	3,4

Tabel 5.10 Hasil Pengujian Kebergunaan (Lanjutan)

No	Poin yang dinilai	Nilai						Total	Rata-rata
		1	2	3	4	5	6		
6	Kemudahan kakas bantu dalam melakukan evaluasi	0	0	0	5	3	2	10	4,7
Total									4,03

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan yang diperoleh selama pengerjaan Tugas Akhir. Selain itu, juga terdapat saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut:

1. Parameter perhitungan tiap metrik pada karakteristik portabilitas ISO/IEC 9126-3 dapat ditentukan dengan melakukan studi literatur secara mendalam dan analisis terhadap ISO/IEC 9126-1 dan ISO/IEC 9126-3 serta referensi terkait.
2. Hasil pengukuran kualitas SIAKAD ITS Modul Penilaian berdasarkan karakteristik portabilitas ISO/IEC 9126-3 menunjukkan hasil yang bagus. Dari beberapa metrik yang ada pada karakteristik portabilitas, setelah dilakukan penyesuaian dengan kebutuhan dan hasil peninjauan sistem ada tiga metrik yang digunakan untuk mengukur kualitas.
3. Kakas bantu yang dibangun dapat membantu pengembang perangkat lunak dalam melakukan proses pengukuran kualitas berdasarkan karakteristik portabilitas dengan model kualitas ISO/IEC 9126-3.

6.2. Saran

Berikut saran-saran untuk pengembangan dan perbaikan yang diharapkan dapat membuat Tugas Akhir ini menjadi lebih baik. Diantaranya adalah sebagai berikut.

1. Tugas Akhir ini hanya memfokuskan penentuan parameter pada karakteristik portabilitas saja. Sedangkan untuk dapat melakukan pengukuran kualitas secara utuh diperlukan

parameter perhitungan untuk karakteristik lain sehingga perlu diadakan penentuan parameter untuk karakteristik lain.

2. Melakukan pengukuran kualitas SIAKAD ITS Modul Penilaian secara menyeluruh, tidak hanya berdasarkan karakteristik portabilitas namun juga berdasarkan karakteristik lain yang ada pada ISO/IEC 9126-3.

LAMPIRAN A. TABEL METRIK

Tabel 7.1 Metrik Adaptasi Lingkungan Perangkat Keras

Komponen	Deskripsi
Nama Metrik	Adaptasi lingkungan perangkat keras
Tujuan Metrik	Mengukur seberapa adaptif perangkat lunak terhadap perubahan lingkungan perangkat keras (perangkat keras dan fasilitas jaringan)
Metode Pengukuran	Menghitung jumlah fitur yang diimplementasi yang mampu mencapai hasil yang diharapkan pada lingkungan perangkat keras yang ditentukan dan membandingkan dengan jumlah kebutuhan fitur dengan kemampuan adaptasi lingkungan perangkat keras
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah fitur yang berhasil mencapai kebutuhan pada lingkungan perangkat keras yang ditentukan B = jumlah kebutuhan fitur dengan kemampuan adaptasi lingkungan perangkat keras
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Spesifikasi kebutuhan, rancangan sistem, dan laporan peninjauan
Target pemirsa	Klien, pengembang, dan pemelihara

Tabel 7.2 Metrik Adaptasi Lingkungan Organisasi

Komponen	Deskripsi
Nama Metrik	Adaptasi lingkungan organisasi
Tujuan Metrik	Mengukur seberapa adaptif perangkat lunak terhadap perubahan organisasi (infrastruktur organisasi)

Metode Pengukuran	Menghitung jumlah fitur yang diimplementasi yang mampu mencapai hasil yang diharapkan pada lingkungan organisasi dan lingkungan bisnis yang ditentukan dan membandingkan dengan jumlah kebutuhan fitur dengan kemampuan adaptasi lingkungan organisasi
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah fitur yang berhasil mencapai kebutuhan pada lingkungan organisasi dan lingkungan bisnis yang ditentukan B = jumlah kebutuhan fitur dengan kemampuan adaptasi lingkungan organisasi
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Spesifikasi kebutuhan, rancangan sistem, dan laporan peninjauan
Target pemirsa	Klien, pengembang, dan pemelihara

Tabel 7.3 Metrik Adaptasi Lingkungan Sistem Perangkat Lunak

Komponen	Deskripsi
Nama Metrik	Adaptasi lingkungan sistem perangkat lunak
Tujuan Metrik	Mengukur seberapa adaptif perangkat lunak terhadap perubahan sistem perangkat lunak (sistem operasi, perangkat lunak jaringan, dan perangkat lunak bantuan)
Metode Pengukuran	Menghitung jumlah fitur yang diimplementasi yang mampu mencapai hasil yang diharapkan pada lingkungan sistem perangkat lunak dan membandingkan dengan jumlah kebutuhan fitur dengan kemampuan adaptasi lingkungan sistem perangkat lunak
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah fitur yang berhasil mencapai kebutuhan pada

	lingkungan organisasi dan lingkungan bisnis yang ditentukan B = jumlah kebutuhan fitur dengan kemampuan adaptasi lingkungan organisasi
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Spesifikasi kebutuhan, rancangan sistem, dan laporan peninjauan
Target pemirsa	Klien, pengembang, dan pemelihara

Tabel 7.4 Metrik Kemudahan untuk Melakukan Pemindahan Perangkat Lunak

Komponen	Deskripsi
Nama Metrik	Kemudahan untuk melakukan pemindahan perangkat lunak
Tujuan Metrik	Mengukur seberapa kecil usaha yang dibutuhkan untuk melakukan pemindahan perangkat lunak
Metode Pengukuran	Menghitung jumlah fitur yang diimplementasi yang mampu mendukung kemudahan adaptasi bagi pengguna dan membandingkan dengan jumlah kebutuhan fitur dengan kemudahan adaptasi
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah fitur yang mendukung kemudahan adaptasi bagi pengguna B = jumlah kebutuhan fitur dengan kemampuan kemudahan adaptasi
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Spesifikasi kebutuhan, rancangan sistem, dan laporan

	peninjauan
Target pemirsa	Klien, pengembang, dan pemelihara

Tabel 7.5 Metrik Kemudahan Percobaan Ulang Pengaturan

Komponen	Deskripsi
Nama Metrik	Kemudahan percobaan ulang pengaturan
Tujuan Metrik	Mengukur seberapa mudah untuk mengulang operasi pengaturan
Metode Pengukuran	Menghitung jumlah operasi percobaan ulang pengaturan dan membandingkan dengan jumlah percobaan pengaturan ulang yang dibutuhkan
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah operasi percobaan ulang untuk pengaturan B = jumlah operasi pengaturan yang dibutuhkan
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Laporan peninjauan
Target pemirsa	Pengembang

Tabel 7.6 Metrik Usaha Pemasangan

Komponen	Deskripsi
Nama Metrik	Usaha pemasangan
Tujuan Metrik	Mengukur seberapa besar usaha yang dibutuhkan untuk memasang perangkat lunak
Metode Pengukuran	Menghitung jumlah langkah pemasangan secara otomatis dan membandingkan dengan jumlah langkah pemasangan yang ditetapkan
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran

	A = jumlah langkah pemasangan secara otomatis B = jumlah langkah pemasangan yang dibutuhkan
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Laporan peninjauan
Target pemirsa	Pengembang

Tabel 7.7 Metrik Fleksibilitas Pemasangan

Komponen	Deskripsi
Nama Metrik	Fleksibilitas pemasangan
Tujuan Metrik	Mengukur seberapa fleksibel kemampuan pemasangan sistem
Metode Pengukuran	Menghitung jumlah operasi pemasangan yang dapat disesuaikan dan membandingkan dengan jumlah kebutuhan operasi pemasangan dengan kemampuan penyesuaian
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah operasi pemasangan yang dapat disesuaikan B = jumlah kebutuhan operasi pemasangan dengan kemampuan penyesuaian
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Spesifikasi kebutuhan, laporan peninjauan
Target pemirsa	Pengembang

Tabel 7.8 Metrik Ketersediaan Keberdampingan

Komponen	Deskripsi
-----------------	------------------

Nama Metrik	Ketersediaan keberdampingan
Tujuan Metrik	Mengukur seberapa fleksibel perangkat lunak ketika berbagi lingkungan sistem dengan perangkat lunak lain tanpa mempengaruhi perangkat lunak lain
Metode Pengukuran	Menghitung jumlah entitas dimana perangkat lunak dapat berdampingan dan membandingkan dengan jumlah entitas pada lingkungan sistem yang seharusnya dapat berdampingan
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah entitas perangkat lunak yang dapat berdampingan B = jumlah entitas pada lingkungan sistem yang seharusnya dapat berdampingan
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Spesifikasi kebutuhan, laporan peninjauan, laporan pengujian
Target pemirsa	Klien, pengembang dan pemelihara

Tabel 7.9 Metrik Kelangsungan Penggunaan Data

Komponen	Deskripsi
Nama Metrik	Kelangsungan penggunaan data
Tujuan Metrik	Mengukur jumlah data asli yang tidak berubah setelah mengalami penggantian perangkat lunak
Metode Pengukuran	Menghitung jumlah data yang tetap digunakan setelah penggantian dan membandingkan dengan jumlah data asli yang seharusnya digunakan setelah penggantian perangkat lunak
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran

	A = jumlah data perangkat lunak yang tetap digunakan setelah penggantian B = jumlah data asli yang seharusnya digunakan setelah penggantian perangkat lunak
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Rancangan sistem, kode sumber, laporan peninjauan, laporan pengujian
Target pemirsa	Klien, pengembang, dan pemelihara

Tabel 7.10 Metrik Cakupan Fitur

Komponen	Deskripsi
Nama Metrik	Cakupan fitur
Tujuan Metrik	Mengukur jumlah fitur yang tidak mengalami perubahan
Metode Pengukuran	Menghitung jumlah fitur yang tercakup dalam perangkat lunak baru yang mengeluarkan hasil yang sama dan membandingkan dengan jumlah fitur pada perangkat lunak lama
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah fitur yang tercakup dalam perangkat lunak baru yang mengeluarkan hasil yang sama B = jumlah fitur pada perangkat lunak lama
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Rancangan sistem, kode sumber, laporan peninjauan, laporan pengujian
Target pemirsa	Klien, pengembang, dan pemelihara

Tabel 7.11 Metrik Pemenuhan Aspek Portabilitas

Komponen	Deskripsi
Nama Metrik	Pemenuhan aspek portabilitas
Tujuan Metrik	Mengukur apakah perangkat lunak sudah memenuhi aspek portabilitas dari aturan, standar, dan persetujuan yang terkait
Metode Pengukuran	Menghitung jumlah komponen yang sudah memenuhi aspek portabilitas dari aturan dan membandingkan dengan jumlah komponen yang seharusnya memenuhi aspek portabilitas dari aturan
Formula	$X = \frac{A}{B}$
Keterangan Formula	X = hasil pengukuran A = jumlah komponen yang sudah memenuhi aspek portabilitas dari aturan B = jumlah komponen yang seharusnya memenuhi aspek portabilitas dari aturan
Interpretasi Hasil	$0 \leq x \leq 1$
Karakteristik Keberhasilan	X disebut berhasil jika nilainya mendekati 1
Skala Pengukuran	Absolut
Sumber Pengukuran	Spesifikasi standar, persetujuan, dan aturan yang berhubungan dengan aspek portabilitas, rancangan sistem, kode sumber, laporan peninjauan
Target pemirsa	Klien dan pengembang

LAMPIRAN B. KUESIONER PENGUJIAN KEBERGUNAAN KAKAS BANTU

Kuesioner Uji Kebergunaan Kakas Bantu Evaluasi Portabilitas
ISO/IEC 9126-3

Nama responden:

Sistem yang diuji:

1. Apakah kakas bantu mudah digunakan?

1	2	3	4	5	6
---	---	---	---	---	---

2. Apakah fitur pada kakas bantu mudah untuk dimengerti?

1	2	3	4	5	6
---	---	---	---	---	---

3. Apakah antarmuka kakas bantu mudah untuk dimengerti?

1	2	3	4	5	6
---	---	---	---	---	---

4. Apakah pesan pemberitahuan yang ada pada kakas bantu cukup jelas?

1	2	3	4	5	6
---	---	---	---	---	---

5. Apakah antarmuka kakas bantu menarik?

1	2	3	4	5	6
---	---	---	---	---	---

6. Apakah kakas bantu memudahkan pengguna dalam proses evaluasi sistem?

1	2	3	4	5	6
---	---	---	---	---	---

BIODATA PENULIS



Penulis, Istiningdyah Saptarini, lahir di kota Palembang pada tanggal 7 Agustus 1992. Penulis adalah anak pertama dari dua bersaudara dan dibesarkan di kota Sidoarjo, Jawa Timur.

Penulis menempuh pendidikan dasar di SDN Wedoro 1. Untuk pendidikan menengah, penulis tempuh di SMPN 1 Sidoarjo. Setelah itu penulis melanjutkan pendidikan di SMAN 5 Surabaya. Pada tahun 2010, penulis memulai pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur.

Di jurusan Teknik Informatika, penulis mengambil bidang minat Rekayasa Perangkat Lunak dan memiliki ketertarikan di bidang *Software Quality Assurance (SQA)*, *Software Maintenance*, dan *Software Evolution*. Penulis juga aktif dalam organisasi kemahasiswaan seperti Himpunan Mahasiswa Teknik Computer (HMTC). Selama kuliah penulis pernah menjadi asisten dosen untuk mata kuliah Matematika Diskrit. Penulis dapat dihubungi melalui alamat email istiningdyah.057@gmail.com.