

PENILAIAN KUALITAS PERANGKAT LUNAK PADA APLIKASI AKTA NOTARIS FIDUSIA DI CV. FREDAVELOP

Muhlis Sul Aen¹, Alif Finandhita²

Program Studi Teknik Informatika – Universitas Komputer Indonesia

Jl. Dipati Ukur No. 112-116, Bandung

E-mail : muhlissulaen@gmail.com¹, finandhita@gmail.com²

ABSTRAK

CV. Fredavelep merupakan perusahaan yang bergerak dalam bidang IT (Teknik Informatika). CV. Fredavelep memproduksi sebuah *web* aplikasi yang dirancang khusus untuk membantu meningkatkan kinerja notaris dalam pembuatan akta secara praktis, yaitu akta notaris fidusia. Aplikasi akta notaris fidusia yang ada saat ini memiliki penilaian yang kurang baik berdasarkan kuesioner kepuasan pengguna. Hal tersebut menjadi masalah yang dimiliki aplikasi akta notaris fidusia. Salah satu solusi terbaik untuk menilai kualitas sebuah sistem perangkat lunak adalah melalui proses evaluasi untuk mengetahui lebih rinci masalah yang dimiliki oleh *web* aplikasi tersebut.

Salah satu standar yang digunakan untuk mendukung proses evaluasi kualitas perangkat lunak adalah standar ISO-25010. Penilaian standar ISO-25010 dilakukan melalui tahapan pendefinisian kebutuhan perangkat lunak berdasarkan keterkaitan antar karakteristik perangkat lunak tersebut. Dalam melakukan penilaian faktor kualitas digunakan kuesioner yang dibuat berdasarkan karakteristik dari perangkat lunak, dengan pendekatan *Goal Question Metrics* (GQM), yang kemudian digunakan untuk melakukan pembobotan pada tiap faktor dan sub-faktor kualitas. Setelah itu akan ditentukan pengategorian kualitas perangkat lunak dari tahap pengukuran. Penilaian metrik kualitas internal dilakukan menggunakan metrik *Chidamber and Kemerer* (CK). Penilaian struktur *code* dan *class* program dilakukan untuk mengetahui bagian *code* dan *class* mana yang perlu diperbaiki pada perangkat lunak.

Kata kunci : Kualitas Perangkat Lunak, ISO-25010, *Goal Question Metrics* (GQM), *Metrics Chidamber and Kemerer* (CK)

1. PENDAHULUAN

Web adalah perangkat lunak yang bentuknya paling sederhana yang memiliki *hypertext* yang menyajikan informasi dengan menggunakan teks, gambar, animasi, suara, dan lain-lain yang tersimpan dalam sebuah internet [1]. Salah satu

contoh aplikasi *web* untuk pembuatan Akta yaitu Fidusia. Aplikasi Fidusia merupakan sebuah aplikasi berbasis *web* yang digunakan oleh notaris untuk melakukan transaksi perjanjian hutang piutang antara kreditor dan debitor dengan melibatkan penjaminan. Pada aplikasi ini dirancang khusus untuk membantu meningkatkan kinerja Notaris dalam pembuatan Akta secara praktis yang diproduksi oleh CV. Fredavelep.

Aplikasi akta notaris fidusia ini dibangun dengan konsep OOP (*Object Oriented Programing*)[2][3], dan diketahui dari struktur kode dapat mempengaruhi *respond* suatu aplikasi[4]. Masalah yang muncul ketika aplikasi ini baru digunakan setelah *launching*, yaitu terdapat kendala di mana ada keluhan pengguna yaitu *respons* dari aplikasi fidusia yang lambat ketika diakses, sehingga mengakibatkan efek kerja *user* menjadi lebih lambat. Hal itu disebabkan perancangan desain arsitektur *code* dan arsitektur *class* yang tidak tepat, sehingga akan mempengaruhi kualitas dari aplikasi tersebut[5].

Berdasarkan dari hasil penelitian yang dilakukan oleh Boehm dan Jones, kesalahan memang telah ada secara ilmiah pada setiap pengembangan perangkat lunak yang baru saja dilakukan pengembangan dimungkinkan terdapat kesalahan sehingga aplikasi menjadi lebih berat[6]. Guna menunjang kualitas pada aplikasi fidusia, penilaian perangkat lunak dilakukan dari sisi produk, sehingga berdampak pada perbaikan sistem untuk ke depannya.

Untuk mengetahui kualitas dari sebuah perangkat lunak, dapat menggunakan standar ISO/IEC 25010 yang di mana digunakan untuk mengevaluasi sistem perangkat lunak secara spesifik [7] dan untuk penilaian struktur *code* dan *class* menggunakan metrik *Chidamber and Kemerer* (CK) [8].

2. ISI PENELITIAN

2.1.1 Landasan Teori

2.1.1 Kualitas Perangkat Lunak

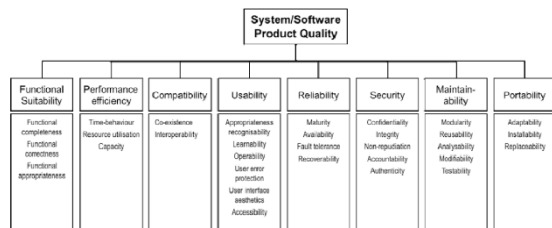
Kualitas perangkat lunak bisa didefinisikan sebagai kesesuaian yang diharapkan pada semua perangkat lunak yang dibangun berkaitan dengan

fungsi perangkat lunak yang diutamakan dan untuk kerja perangkat lunak[9]. Menurut badan standar internasional (IEEE, ISO) ada beberapa sudut pandang untuk menggambarkan kualitas perangkat lunak. Standar ISO menggambarkan tiga pandangan kualitas[10], yaitu :

1. Pandangan pengguna (*user*)
2. Pandangan pengembang (*developer*), dan
3. Pandangan manajer (*manager*)

2.1.2 Standar Kualitas Perangkat Lunak

Standar yang digunakan untuk menilai kualitas perangkat lunak adalah Standar ISO 25010. ISO 25010 adalah salah satu metode standar internasional yang digunakan untuk menilai standar kualitas sebuah perangkat lunak yang dirilis oleh ISO/IEC[11][12]. Standar ISO 25010 dapat digunakan untuk mengevaluasi kualitas produk perangkat lunak. Berikut model *software products quality* dari standar ISO 25010 dapat dilihat pada Gambar 1.



Gambar 1. Diagram Model Kualitas *Product Quality*

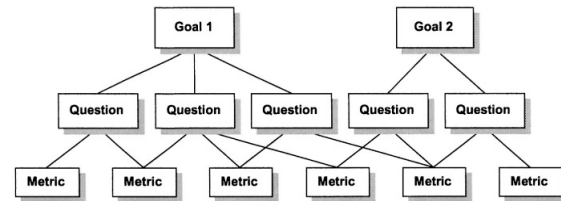
2.1.3 The Goal Question Metrics (GQM)

The Goal Question Metric (GQM) adalah sebuah metode struktur hirarkis di mulai dengan tujuan (menetapkan tujuan pengukuran, objek yang akan diukur, masalah yang akan diukur dan ukuran dari sudut pandang mana yang akan diambil). Tujuan dari GQM adalah untuk membnetuk beberapa pertanyaan, setiap pertanyaan kemudian dibentuk menjadi metrik. Beberapa metrik dapat digunakan untuk pertanyaan-pertanyaan yang berbeda dengan tujuan yang sama. Beberapa metode GQM juga dapat memiliki pertanyaan dan metric yang sama[13].

Model pengukuran GQM terdapat 3 tingkatan, yaitu :

1. *Goal* : Tujuan yang ditetapkan untuk sebuah objek, untuk berbagai alasan, sehubungan dengan berbagai model kualitas.
2. *Question* : Seperangkat pertanyaan yang digunakan untuk menggambarkan cara penilaian atau pencapaian tujuan tertentu.
3. *Metric* : Satu set data terkait dengan setiap pertanyaan untuk menjawab dengan cara kuantitatif.

Berikut adalah struktur hirarki metode GQM seperti terlihat pada Gambar 2.



Gambar 2. Struktur Hirarki Metode GQM

2.1.4 Chidamber and Kemerer Metrics

Metriks dari Chidamber dan Khemerer untuk mengukur *coding* / program dari aplikasi akta notaris fidusia yang berkonsep OOP.

Menurut Chidamber dan Kemerer Metric untuk mengukur dan memberikan penilaian perangkat lunak berbasis *objek oriented* dapat menggunakan enam buah metrik[8][14], yaitu :

1. *Weight Methods per Class* (WMC)

Pengukuran metriks WMC adalah menghitung jumlah method yang diimplementasikan dalam suatu *class*. Untuk mengetahuinya maka dihitung seluruh *method* lokal pada sebuah perangkat lunak

$$\text{Rumus : } m_1 + m_2 + m_3 + \dots + m_h \quad (1)$$

2. *Depth of Inheritance Tree* (DIT)

DIT merupakan penghitungan jalur *inheritance* maksimum dari sebuah *class*. Kedalaman sebuah *class* dalam hirarki pewarisan diukur oleh DIT.

Rumus : Apabila ada pewarisan diberikan nilai 1 dan apabila tidak ada diberikan nilai 0.

3. *Number of Children* (NOC)

Pengukuran NOC adalah menghitung jumlah *children* atau anak kelas

Rumus : Apabila ada anak diberi nilai dan apabila tidak tidak diberi nilai.

4. *Coupling Between Objects* (CBO)

Coupling adalah ketergantungan modul satu dengan modul lainnya, *coupling* yang baik adalah *coupling* yang rendah di mana modul satu dengan modul lainnya tidak saling ketergantungan.

Rumus : Menghitung jumlah *method* lokal yang mengakses ke *method* eksternal

5. *Respon for a Class* (RFC)

Pengukuran RFC yaitu menghitung jumlah *method* lokal yang diimplementasikan ditambah *method* yang dipanggil dalam objek.

Rumus : jumlah *method* lokal + jumlah *method* eksternal.

6. *Lack OF Cohesion in Method* (LCOM)

LCOM adalah metrik untuk menghitung sebuah *Cohesi*. *Cohesi* adalah fungsi yang terkait dalam sebuah modul. *Cohesi* yang baik adalah *cohesi* yang tinggi, semakin tinggi *cohesi* semakin baik perancangannya.. Pengukuran LCOM dilakukan dengan menghitung jumlah koneksi pada *method* [14][15][16].

Rumus Perhitungannya adalah:

$$LCOM^* = (m - \sum (Mu) / a) / (m-1) \quad (2)$$

Dimana:

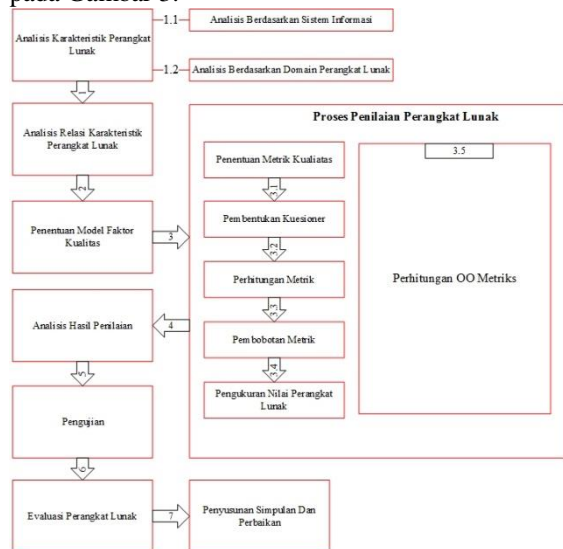
m adalah jumlah *method* dalam *class*.

a adalah jumlah *variable* dalam *class*.

mu adalah jumlah *variable* yang dipanggil.

2.1.5 Tahapan Penilaian Kualitas Perangkat Lunak

Tahapan dalam melakukan penilaian perangkat lunak dengan urutan dan prosedur tertentu yang bersifat tetap dengan menggunakan metode kuantitatif [17]. Berikut adalah alur dari tahapan penilaian kualitas perangkat lunak dapat dilihat pada Gambar 3.



Gambar 3. Tahapan Penilaian Kualitas Perangkat Lunak

2.2 Analisis Sistem

2.2.1 Analisis Karakteristik Perangkat Lunak

Analisis karakteristik perangkat lunak adalah dengan cara menganalisis berdasarkan komponen sistem informasi dan *domain* perangkat lunak.

1. Analisis komponen sistem informasi

Analisis komponen sistem yang ada pada aplikasi akta notaris fidusia. Berikut adalah analisis komponen sistem informasi sebagai berikut[18] :

a. Karakteristik *Inputan*

Karakteristik *inputan* pada aplikasi fidusia yaitu *inputan menu* cabang untuk mengisikan cabang mana saja yang digunakan dalam pembuatan akta, menu kuasa menunjukan pemberi dan penerima kuasa, menu *user* untuk memasukan pengguna aplikasi, menu pendaftaran yang akan digunakan untuk membuat akta notaris

b. Karakteristik *Models*

Karakteristik *Models* merupakan komponen yang berisikan proses koneksi *database* ke aplikasi, koneksi akan memanggil *table* yang dipilih untuk kemudian diarahkan ke dalam *controller*.

c. Karakteristik *Views*

Karakteristik *Views* merupakan keluaran atau *output* dari aplikasi fidusia yaitu sebagai berikut : menampilkan halaman *login*, tampilan *dashboard*

utama memantau jumlah transaksi yang sudah dilakukan oleh *user*, menampilkan data cabang, kuasa, *user*, pendaftaran sebagai *inputan* yang akan masuk ke dalam tampilan akta di menu akta

d. Karakteristik *Controllers*

sekumpulan proses yang terdiri dari proses menampilkan, proses tambah, proses hapus, proses ubah, dan proses pencarian dan *generate* akta.

e. Karakteristik *Aunthentication*

Mengatur hak akses *user* yaitu, owner, admin dan editor. Yang di *setting* setiap hak akses memiliki tugas pokok dan fungsi masing-masing.

f. Karakteristik *Database*

Karakteristik *database* yang digunakan dalam aplikasi fidusia memiliki *table-table* yang saling berelasi, yang terdiri dari struktur tabel yang memiliki *primary key* masing-masing tabel.

2. Analisis *Domain* Perangkat Lunak

Analisis karakteristik berdasarkan *domain* perangkat lunak menjelaskan tentang *fungsi-fungsionalitas* yang terdapat pada aplikasi fidusia. Karakteristik yang dimiliki oleh aplikasi fidusia adalah sebagai berikut[18] :

a. *Informational*

Aplikasi fidusia memiliki tampilan *dashboard* sebagai sarana *informatif* dalam menampilkan jumlah data yang di *inputkan* oleh *user*.

b. *Customizable*

User dapat mengganti tema atau latar belakang halaman aplikasi sesuai dengan tema yang telah disediakan, fitur ini digunakan untuk mengcostume tampilan sesuai keinginan *user*.

c. *Interaction*

Interaksi yang terjalin pada aplikasi fidusia yaitu antara tugas pokok dan fungsi *user* masing-masing saling berhubungan

d. *User Input*

Aplikasi fidusia memiliki fasilitas *user input*, seperti *input* cabang, kuasa, *user* dan pendaftaran.

e. *Help*

Pada aplikasi fidusia disediakan *fungsi-fungsionalitas* sebagai pedoman dalam menggunakan aplikasi, sehingga *user* dapat mengetahui informasi cara penggunaan melalui *fungsi-fungsionalitas* tersebut.

2.2.2 Penentuan Model Faktor Kualitas

Analisis faktor kualitas perangkat lunak menjelaskan tentang analisis dari model kualitas perangkat lunak, metrik keterkaitan karakteristik dengan model kualitas perangkat lunak. Berikut adalah hasil dari pemetaan keterkaitan berdasarkan komponen sistem informasi dan domain perangkat lunak terhadap standar ISO 25010[5], dapat dilihat pada Tabel 1.

Tabel 1. Faktor dan Sub-faktor Yang Digunakan

Faktor	Sub-faktor
<i>Functional Suitability</i>	<i>Functional Completeness</i>
	<i>Functionl Correctness</i>
	<i>Functionl Appropriateness</i>
<i>Performance Efficiency</i>	<i>Time Behaviour</i>
	<i>Resouce Utilization</i>
	<i>Capacity</i>
<i>Usability</i>	<i>Appopriateness Recognizability</i>
	<i>Operability</i>
	<i>Use Error Protection</i>
	<i>Use Interface Aesthetentics</i>
	<i>Accessibility</i>
<i>Reliability</i>	<i>Maturity</i>
	<i>Availability</i>
	<i>Fault Tolerance</i>
<i>Security</i>	<i>Confidentiality</i>
	<i>Integrity</i>
	<i>Authenticity</i>
<i>Maintability</i>	<i>Reusability</i>
	<i>Analysability</i>
	<i>Testability</i>
<i>Portability</i>	<i>Replaceability</i>

2.2.3 Pembentukan Pertanyaan Dengan Metode Goal Question Metrics (GQM)

Berikut ini adalah penjelasan terkait dengan menggunakan metode GQM (*Goal Question Metric*) untuk pembentukan pertanyaan berdasarkan faktor kualitas menggunakan metode skala likert [19] dan skala guttman [20], seperti terlihat pada Tabel 2.

Tabel 2. Pembentukan Pertanyaan Berdasarkan Sub-faktor

<i>Goal</i>	Mencari nilai sub-faktor kualitas
<i>Question</i>	Beberapa pertanyaan berdasarkan sub-faktor kualitas
<i>Metric</i>	M1 = Bobot Skala Likert X Hasil Jawaban Responden $x = \sum \frac{\text{Total}}{n}$
	M2 = Bobot Skala Guttman X Hasil Jawaban Responden $x = \sum \frac{\text{Total}}{n}$

2.2.4 Perhitungan Metrik

Berikut adalah hasil yang didapat dari pertanyaan yang telah dibuat. Berikut adalah contoh perhitungan metrik yang terdapat pada faktor *functionl suitability* dengan sub-faktor meliputi *functionl completeness*, *functional correctness*, dan *functional appropriateness*.

Functionl Complpeteness.

Berikut adalah perhitungan nilai rata-rata dari jawaban 20 responden admin, 5 responden editor dan 1 responden owner terhadap masing-masing pertanyaan yang berkaitan dengan sub-faktor *functional completeness*.

Tabel 3. Tanggapan Responden Admin Terhadap Sub-Faktor *Functional Completeness*

No. Pertanyaan	Jawaban		Jumlah Niali		Total	Rata-rata
	Ya	Tidak	Ya	Tidak		
1	8	12	40	12	52	2,6

Tabel 4. Tanggapan Responden Editor Terhadap Sub-faktor *Functional Completeness*

No. Pertanyaan	Jawaban		Jumlah Niali		Total	Rata-rata
	Ya	Tidak	Ya	Tidak		
1	1	4	5	4	9	1,8

Tabel 5. Tanggapan Responden Owner Terhadap Sub-faktor *Functionl Completeness*

No. Pertanyaan	Jawaban		Jumlah Niali		Total	Rata-rata
	Ya	Tidak	Ya	Tidak		
1	1	0	5	0	5	5

Setelah didapat nilai rata-rata pada seperti yang terlihat pada Tabel , Tabel , dan Tabel maka dapat dihitung nilai dari sub-faktor *functional completeness*. Berikut perhitungan sub-faktor *functional completeness* dapat dilihat pada Tabel 6.

Tabel 6. Perhitungan Sub-faktor *Functionl Completeness*

Sub-Faktor	Perhitungan
<i>Functional Completeness</i>	$x = \sum \frac{\text{Total}}{n}$ <p> Xadmin = 52/20 = 2,6 Xeditor = 9/5 = 1,8 Xowner = 5/1 = 5 XFunctional Completeness = (2,6 + 1,8 + 5)/3 = 3,13 </p>

Jadi, nilai sub-faktor *functional completeness* berdasarkan perhitungan pada Tabel 6 adalah 3,13.

2.2.5 Pembentukan Pembobotan

Pembobotan pada faktor standar kualitas menggunakan metode *Rank Order Centroid (ROC)* diperlukan untuk menghitung faktor kualitas[21]. Berikut adalah rumus untuk persamaan *ROC* atau dapat dijelaskan sebagai berikut:

Jika $Cr1 \geq Cr2 \geq Cr3 \geq \dots \geq Crn$
Maka $W1 \geq W2 \geq W3 \geq \dots \geq Wn$
Selanjutnya, jika k merupakan banyaknya kriteria, maka

$$W1 = \frac{1 + \frac{1}{2} + \frac{1}{3} + \dots + 1/k}{k}$$

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

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

Secara umum pembobotan ROC dapat dirumuskan sebagai berikut :

$$Wk = 1/k \sum_{i=1}^k (1/i) \quad (3)$$

w = Nilai Pembobotan kriteria

k = Jumlah kriteria

i = Niali alternatif

Nilai k yang dimiliki oleh faktor kualitas adalah 7 dapat dilihat dari banyaknya keterkaitannya, maka pembobotan faktor standar kualitas adalah sebagai berikut dapat dilihat pada Tabel 7.

Tabel 7. Perhitungan Pembobotan Kualitas

Faktor	Perhitungan Bobot
<i>Functional Suitability</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0,37$
<i>Maintability</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0,23$
<i>Usability</i>	$\frac{1 + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0,15$
<i>Security</i>	$\frac{1 + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0,11$
<i>Reliability</i>	$\frac{1 + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}}{7} = 0,07$
<i>Performance Efficiency</i>	$\frac{1 + \frac{1}{6} + \frac{1}{7}}{7} = 0,04$
<i>Portability</i>	$\frac{1 + \frac{1}{7}}{7} = 0,02$

Pembobotan sub-faktor kualitas diperlukan untuk menghitung nilai dari sub-faktor kualitas. Pembobotan sub-faktor kualitas diperlukan untuk menghitung nilai dari sub-faktor kualitas. Nilai k yang dimiliki oleh sub-faktor kualitas berbeda-beda sesuai dengan sub-faktor yang dimiliki oleh faktor kualitas [21]. Dapat dilihat jumlah sub-faktor yang dimiliki oleh setiap faktor kualitas pada Tabel 8.

Tabel 8. Jumlah Sub-faktor Untuk Setiap Faktor

Faktor	Jumlah Sub-faktor
<i>Functional Suitability</i>	3
<i>Performance Efficiency</i>	3
<i>Usability</i>	5
<i>Reliability</i>	3
<i>Security</i>	3
<i>Maintability</i>	3
<i>Portability</i>	1

Berdasarkan Tabel maka akan dihitung pembobotan dari sub-faktor kualitas, dari standar faktor kualitas. Perhitungan bobot kualitas faktor berdasarkan persamaan ROC dapat dilihat pada Tabel 9.

Tabel 9. Pembobotan Sub-faktor Kualitas

Faktor	Sub-faktor	Perhitungan Pembobotan
<i>Functional Suitability</i>	<i>Functional Completeness</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,61$
	<i>Functional Correctness</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,28$
	<i>Functional Appropriateness</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,11$
<i>Performance Efficiency</i>	<i>Time Behaviour</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,61$
	<i>Resource Utilization</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,28$
	<i>Capacity</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,11$

Faktor	Sub-faktor	Perhitungan Pembobotan
<i>Usability</i>	<i>Appropriateness</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}}{5} = 0,46$
	<i>Recognizability</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}}{5} = 0,26$
	<i>Accessibility</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}}{5} = 0,16$
	<i>Operability</i>	$\frac{1 + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}}{5} = 0,09$
	<i>User Error Protection</i>	$\frac{1 + \frac{1}{4} + \frac{1}{5}}{5} = 0,04$
	<i>User Interface</i>	$\frac{1 + \frac{1}{5}}{5} = 0,04$
<i>Reliability</i>	<i>Maturity</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,61$
	<i>Availability</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,28$
	<i>Fault Tolerance</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,11$
<i>Security</i>	<i>Confidentiality</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,61$
	<i>Authenticity</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,28$
	<i>Integrity</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,11$
<i>Maintability</i>	<i>Testability</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,61$
	<i>Reusability</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,28$
	<i>Analysability</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 0,11$
<i>Portability</i>	<i>Replaceability</i>	$\frac{1 + \frac{1}{2} + \frac{1}{3}}{3} = 1$

2.2.6 Pengukuran Nilai Perangkat Lunak

Berdasarkan perhitungan metrik kualitas pada Tabel 2 dan pembobotan yang telah dilakukan pada Tabel 7 dan Tabel 9, maka pengukuran nilai perangkat lunak dapat dilakukan menggunakan metode *weighted summation* [22]. Berikut adalah hasil perhitungan setiap faktor kualitas terhadap aplikasi *fidusia* menggunakan persamaan $score(C_i) = \sum_{i=1}^n W_i.X_i(C_i)$ dapat dilihat sebagai berikut.

Functional Suitability

Berikut adalah perhitungan nilai faktor kualitas *functional suitability* menggunakan persamaan $score(C_i) = \sum_{i=1}^n W_i.X_i(C_i)$ (4)

$$\begin{aligned}
 C_1 &= (W_{functional\ completeness} \cdot X_{functional\ completeness}) + (W_{functional\ correctness} \cdot X_{functional\ correctness}) + (W_{functional\ appropriateness} \cdot X_{functional\ appropriateness}) \\
 &= (0,61 \cdot 3,13) + (0,28 \cdot 2,73) + (0,11 \cdot 3,5) \\
 &= 1,91 + 0,76 + 0,38 \\
 &= 3,05
 \end{aligned}$$

Berikut adalah hasil perhitungan dapat dilihat pada Tabel 10.

Tabel 10. Hasil Perhitungan Faktor Kualitas

No.	Faktor	Nilai
1	<i>Functional Suitability</i>	3,05
2	<i>Performance Efficiency</i>	3,34
3	<i>Usability</i>	2,98
4	<i>Reliability</i>	1,61
5	<i>Security</i>	3,36
6	<i>Maintability</i>	3,1
7	<i>Portability</i>	2,77

Berdasarkan nilai faktor yang terdapat pada Tabel. 10, maka nilai-nilai tersebut akan dilihat termasuk ke dalam kategori mana. Penentuan kategori merujuk pada metrik *eksternal* pada kualitas perangkat lunak digunakan tiga kategori kualitas, pembentukan kategori tersebut dapat dilihat pada Tabel 11.

Tabel 11. Kategori Kualitas

Kategori	Interval Nilai
<i>High</i>	3,36 – 5
<i>Medium</i>	1,68 – 3,35
<i>Low</i>	0 – 0,167

Dapat dilihat pada Tabel 12 dari hasil penilaian dan pembentukan kategori kualitas.

Tabel 12. Nilai Faktor Kualitas

Faktor	Nilai	Kategori Kualitas
<i>Functional Suitability</i>	3,05	<i>Medium</i>
<i>Performance Efficiency</i>	3,34	<i>Medium</i>
<i>Usability</i>	2,98	<i>Medium</i>
<i>Reliability</i>	1,61	<i>Low</i>
<i>Security</i>	3,36	<i>High</i>
<i>Maintainability</i>	3,1	<i>Medium</i>
<i>Portability</i>	2,77	<i>Medium</i>

Berikut nilai setiap faktor dapat dilihat pada Tabel 12 dan sub-faktor pada Tabel 13.

Tabel 13. Nilai Dan Faktor Kualitas

Sub-Faktor	Nilai	Kategori Kualitas
<i>Functional Completeness</i>	3,13	<i>Medium</i>
<i>Functionl Correctness</i>	2,73	<i>Medium</i>
<i>Functionl Appropriateness</i>	3,5	<i>High</i>
<i>Time Behaviour</i>	3,15	<i>Medium</i>
<i>Resouce Utilization</i>	3,22	<i>Medium</i>
<i>Capacity</i>	3,93	<i>High</i>
<i>Appopriateness Recognizability</i>	3,6	<i>High</i>
<i>Accessibility</i>	2,38	<i>Medium</i>
<i>Operability</i>	2,93	<i>Medium</i>
<i>Use Error Protection</i>	1,87	<i>Medium</i>
<i>User Interface Aesthetics</i>	1,47	<i>Low</i>
<i>Maturity</i>	1,32	<i>Low</i>
<i>Availability</i>	2,15	<i>Medium</i>
<i>Fault Tolerance</i>	1,88	<i>Medium</i>
<i>Confidentiality</i>	3,8	<i>High</i>

Sub-Faktor	Nilai	Kategori Kualitas
<i>Authencity</i>	3,18	<i>Medium</i>
<i>Integrity</i>	1,33	<i>Low</i>
<i>Testability</i>	3,93	<i>High</i>
<i>Reusability</i>	1,98	<i>Medium</i>
<i>Analysability</i>	1,4	<i>Low</i>
<i>Replaceability</i>	2,77	<i>Medium</i>

2.2.7 Penilaian Akhir Model Kualitas *Product Quality (Eksternal)*

Pembobotan *Product Quality* berdasarkan Tabel 8 dan penilaian faktor dari Tabel 13, maka penilaian akhir *products quality* dapat dilakukan menggunakan metode *weighted summation* [22]. Berikut adalah hasil perhitungan setiap faktor kualitas terhadap aplikasi fidusia menggunakan persamaan dapat dilihat sebagai berikut $score(C_i) = \sum_{i=1}^n W_i \cdot X_i(C_i)$

$$\begin{aligned}
 C_1 &= (W_{functional\ suitability} \cdot X_{functional\ suitability}) + (W_{maintainability} \cdot X_{maintainability}) + (W_{usability} \cdot X_{usability}) + (W_{security} \cdot X_{security}) + (W_{reliability} \cdot X_{reliability}) + (W_{performa\ efficiency} \cdot X_{performa\ efficiency}) + (W_{portability} \cdot X_{portability}) \\
 &= (0,37 \cdot 3,05) + (0,23 \cdot 3,1) + (0,15 \cdot 2,98) + (0,11 \cdot 3,36) + (0,07 \cdot 1,61) + (0,04 \cdot 3,34) + (0,02 \cdot 2,77) \\
 &= 1,13 + 0,71 + 0,45 + 0,37 + 0,11 + 0,13 + 0,05 \\
 &= 2,95
 \end{aligned}$$

Setelah nilai kuesioner dari pengguna aplikasi akta notaris fidusia diolah dari 26 responden yang meliputi 20 responden admin, 5 responden editor, dan 1 responden owner menggunakan fasilitas tersebut, maka didapatkan nilai akhir setelah dibobotkan dan pengukuran adalah 2,95 yang berarti aplikasi akta notaris fidusia dikategorikan medium (cukup baik). Nilai 2.95 itu diambil dari pemetaan berdasarkan Tabel 11.

2.2.8 Penentuan Perhitungan OO Metrik (*Internal*)

Berikut adalah hasil dari analisis internal menggunakan metrik Chidamber dan Khemerer didapatkan hasil berupa tabel analisa internal yang dapat dilihat pada Tabel 14.

Tabel 14. Hasil Perhitungan CK-Metrics

WMC	DIT	NOC	CBO	RFC	LOCM
7,62	0,95	18	1,24	11	0,72

Berdasarkan pengaruh kualitas dan nilai metrik, maka dapat ditentukan nilai *threshold*-nya. Nilai *threshold* yang direkomendasikan pada CK-Metrik ditujukan pada Tabel 15.

Tabel 15. Nilai Threshold Kualitas Untuk CK-Metrik

Metric	Nilai
WMC	<i>low</i> , < 11
DIT	<i>low</i> , < 4
NOC	<i>low</i> , < 8
CBO	<i>low</i> , < 3
RFC	<i>low</i> , < 12
LCOM	<i>low</i> , < 4

Pada penelitian ini, nilai kualitas akan dibagi menjadi *high* dan *low* [14]. Berdasarkan Tabel 14 terhadap Tabel 15, maka dapat disimpulkan hasil nilai yang telah di analisis, dapat dilihat pada Tabel 16.

Tabel 16. Hasil Nilai Akhir

Metrik	Nilai	Kualitas
WMC	7,62	<i>Low</i>
DIT	0,95	<i>Low</i>
NOC	18	<i>High</i>
CBO	1,24	<i>Low</i>
RFC	11	<i>Low</i>
LCOM	0,72	<i>Low</i>

Nilai metrik yang direkomendasikan untuk mendapatkan perangkat lunak yang berkualitas dilihat pada Tabel 17.

Tabel 17. Nilai Kualitas yang Direkomendasikan Untuk CK-Metrik

CK Metric	Recommended
WMC	<i>Low</i>
DIT	<i>Low</i>
NOC	<i>Low</i>
CBO	<i>Low</i>
RFC	<i>Low</i>
LCOM	<i>Low</i>

Berdasarkan Tabel 16 dan 17, maka untuk metrik NOC akan dilakukan perbaikan.

2.3 Perbaikan Sistem

Perbaikan sistem merupakan langkah selanjutnya dari langkah analisis sistem yang di dalamnya terdapat hasil dari penilaian perangkat lunak akta notaris fidusia. Perbaikan dilakukan berdasarkan hasil penilaian yang telah dilakukan. Berikut adalah perbaikan perangkat lunak akta notaris fidusia, berdasarkan hasil pemetaan kategori nilai *low* dapat dilihat pada Tabel 18.

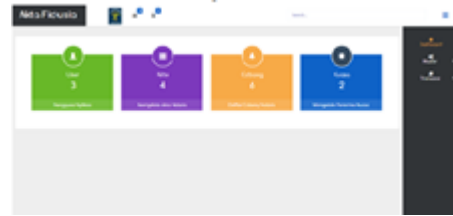
Tabel 18. Hasil Pemetaan Rekomendasi Perbaikan

Sub-Faktor	Nilai	Kategori Kualitas
<i>User Interface aesthetics</i>	1,47	<i>Low</i>
<i>Maturity</i>	1,32	<i>Low</i>
<i>Integrity</i>	1,33	<i>Low</i>
<i>Analysability</i>	1,4	<i>Low</i>

Berdasarkan Tabel 18, maka dapat dijabarkan perbaikan pada aplikasi akta notaris fidusia sebagai berikut :

1. *User Interface Aesthetics*

Pada sisi *responsif* yang di mana untuk sub-faktor *user interface aesthetics* mempunyai nilai yang *low*. Maka dari itu perbaikan akan dilakukan pada sisi *responsive* dengan menerapkan *design responsive* pada *web* aplikasi akta notaris fidusia



Gambar 4. Penerapan *Design Responsive* Pada Kode Program CSS Baru Untuk Ukuran Layar Komputer *Desktop*

2. *Maturity*

Berdasarkan Tabel 17 pada sub-faktor *maturity* mempunyai nilai *low* yang di mana terdapat masalah pada *user experience*[23] juga adalah di mana *web* aplikasi akta notaris fidusia tidak memiliki pesan kesalahan jika admin melakukan kesalahan seperti contoh ketika salah memasukan data yang seharusnya *numerik* atau angka tapi admin memasukan data bertipe karakter atau huruf

*Isi dengan angka

Jenis Kelamin

☐ Pria

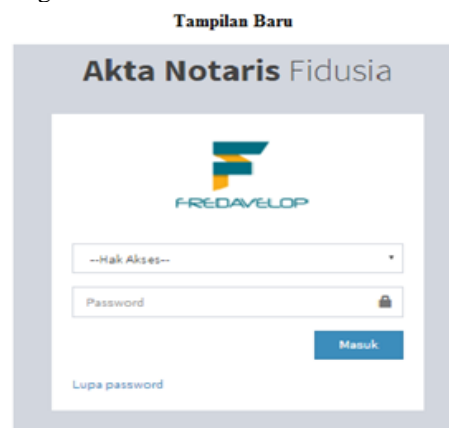
☐ Wanita

*No. Telp isi dengan angka

Gambar 5. Pesan Kesalahan

3. *Integrity*

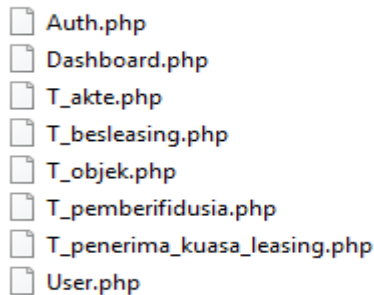
Berdasarkan Tabel 17 pada sub faktor *integrity* yang mempunyai nilai *low* yang juga terdapat masalah pada *user experience*[23] adalah di mana *web* aplikasi akta fidusia tidak memiliki sebuah fungsi pada halaman berupa hak akses, guna mencegah akses tidak sah



Gambar 6. Penerapan Hak Akses

4. *Analysability*

Perbaikan struktur kode program dilakukan berdasarkan hasil penilaian yang telah dilakukan dan berdasarkan hasil penilaian pada Tabel 17 didapat sub faktor *analysability* mempunyai nilai *low*, sehingga akan dilakukan perbaikan struktur kode[24]. Perbaikan standar kode aplikasi akta notaris fidusia menggunakan panduan dari *CodeIgniter*, berikut adalah contoh hasil penerapan standar kode dengan *file naming*



Gambar 7. Penerapan *File Naming*

5. Perbaikan *class* pada metrik NOC

Perbaikan *class* pada metrik NOC dilakukan berdasarkan hasil penilaian yang telah dilakukan dan berdasarkan hasil penilaian pada Tabel 16 didapat NOC memiliki nilai *high*, sedangkan yang direkomendasikan untuk nilai NOC berdasarkan Tabel 17 adalah *low*. Maka, dilakukan perbaikan *class*. Berikut adalah perbaikan NOC pada aplikasi akta notaris fidusia.

Tabel 19. Perbaikan NOC

<i>Class</i>	Jumlah NOC
Auth	0
T_fungsional	0
T_transaksi	0
T_fungsional_model	0
T_transaksi_model	0
My_controller	3
CI_model	2
Total	5

Berdasarkan Tabel 19 didapatkan hasil nilai perbaikan NOC adalah 5 dengan kualitas *low* berdasarkan Tabel 15, sedangkan nilai NOC sebelumnya pada Tabel 14 adalah 18 dengan kualitas *high*, maka dapat dikatakan perbaikan NOC dianggap baik karena sesuai dengan yang direkomendasikan.

2.4 Pengujian Sistem

Pada tahap pengujian ini sistem akan dilakukan pengujian untuk mengetahui kesalahan atau kekurangan pada *web* aplikasi akta notasi fidusia yang telah diperbaiki atau direkomendasikan. Pengujian sistem ini juga bertujuan untuk mengetahui apakah *web* aplikasi akta notaris fidusia yang telah diperbaiki sudah memenuhi tujuan atau belum. Pengujian sistem yang dilakukan adalah

menggunakan pengujian *user acceptance testing* (UAT).

2.4.1 Pengujian *User Acceptance Testing* (UAT)

Pengujian *User Acceptance Testing* Merupakan pengujian yang dilakukan sehubungan dengan kebutuhan pengguna terakhir atau *end user* dengan tujuan untuk menentukan apakah perangkat lunak yang dibangun sudah memenuhi atau tidak kriteria penerimaan dan untuk memungkinkan pengguna, pelanggan atau badan lainnya yang berwenang menentukan apakah perangkat lunak sudah memenuhi syarat diterima atau tidak[25]. Pengujian menggunakan metode kuesioner kepada pengguna aplikasi akta notaris fidusia

2.4.1.1 Kuesioner Pengujian

Adapun pertanyaan dari kuesioner pengujian beta adalah sama dengan yang telah dijelaskan pada Tabel 2 . Berikut adalah jawaban dari setiap pertanyaan kuesioner yang akan dibagikan ke 26 *user*, meliputi 20 admin, 5 editor, dan 1 owner dapat dilihat pada Tabel 20.

Tabel 20. Jawaban Kuesioner

Skala Likert		Skala gutmann	
Jawaban	Keterangan	Jawaban	Keterangan
A	Sangat Setuju	A	Ya
B	Setuju	B	Tidak
C	Biasa Saja		
D	Kurang Setuju		
E	Tidak Setuju		

2.4.1.2 Hasil Kuesioner

Adapun hasil kuesioner pengujian ini berdasarkan Tabel 18 yang di mana hanya menguji hasil kuesioner untuk sub-faktor *user interface aesthetics*, *maturity*, *integrity*, dan *analysability*. Berikut hasil pengujian kuesioner sebagai berikut :

- Hasil pengujian kuesioner soal nomer 10 untuk sub-faktor *user interface aesthetics*.

Tabel 21. Hasil Pengujian Kuesioner Responden Admin Soal no.10

Jawaban	Keterangan	Responden Admin	Jumlah Nilai
A	Ya	11	55
B	Tidak	9	9
Total		20	64
Rata-rata		3,2	

Tabel 22. Hasil Pengujian Kuesioner Responden Editor Soal no.10

Jawaban	Keterangan	Responden Editor	Jumlah Nilai
A	Ya	2	10
B	Tidak	3	3
Total		5	13
Rata-rata		2,6	

Tabel 23. Hasil Pengujian Kuesioner Responden Owner Soal no.10

Jawaban	Keterangan	Responden Owner	Jumlah Nilai
A	Ya	1	5
B	Tidak	0	0
Total		1	5
Rata-rata		5	

Tabel 24. Hasil Pengujian no.10

Responden	Hasil pengujian
Admin	3,2
Editor	2,6
Owner	5
Total	10,8
Rata - rata	3,6

Berdasarkan hasil pengujian pada Tabel 21, Tabel 22, dan Tabel 23 di dapat hasil yang dapat dilihat pada Tabel 24 dengan nilai akhir 3,6 pada aplikasi yang telah di perbaiki. Dapat dilihat perbandingan nilai pada aplikasi yang lama pada Tabel 18 yang mempunyai nilai akhir 1,47 dan nilai aplikasi yang baru pada Tabel 24 dengan nilai akhir 3,6. Untuk lebih jelasnya, hasil perbandingan dapat dilihat pada Tabel 25.

Tabel 25. Perbandingan Hasil Pengujian Soal no.10

Aplikasi Lama		Aplikasi Baru	
Responden	Hasil Pengujian	Responden	Hasil Pengujian
Admin	1,6	Admin	3,2
Editor	1,8	Editor	2,6
Owner	1	Owner	5
Total	4,4	Total	10,8
Rata - rata	1,47	Rata - rata	3,6

Berikut adalah hasil perbandingan dari sub faktor yang lama dengan yang telah diperbaiki dapat dilihat pada Tabel 26 dan Tabel 27.

Tabel 26. Hasil Perbandingan Nilai Sub-Faktor

Aplikasi Lama			Aplikasi Baru		
Sub-faktor	Nilai	Kualitas	Sub-faktor	Nilai	Kualitas
User Interface aesthetics	1,47	Low	User Interface aesthetics	3,6	High
Maturity	1,32	Low	Maturity	3,73	High
Integrity	1,33	Low	Integrity	3,08	Medium
Analysability	1,4	Low	Analysability	3,87	High

Tabel 27. Perbandingan Nilai Perangkat Lunak

Aplikasi Lama			Aplikasi Baru		
Faktor	Nilai	Kualitas	Faktor	Nilai	Kualitas
Usability	2,98	Medium	Usability	3,06	Medium
Reliability	1,61	Low	Reliability	3,08	Medium
Security	3,36	High	Security	3,55	High
Maintability	3,1	Medium	Maintability	3,37	High

Berikut adalah hasil perhitungan setiap faktor kualitas yang telah direkomendasikan terhadap aplikasi fidusia menggunakan persamaan dapat dilihat sebagai berikut

$$score (Ci) = \sum_{i=1}^n Wi . Xi (Ci)$$

Ci=

$$\begin{aligned}
 & (W_{Functional\ Suitability} \cdot X_{Functional\ Suitability}) + \\
 & (W_{Maintability} \cdot X_{Maintability}) + \\
 & (W_{Usability} \cdot X_{Usability}) + (W_{Security} \cdot X_{Security}) + \\
 & (W_{Reliability} \cdot X_{Reliability}) + \\
 & (W_{Performa\ Efficency} \cdot X_{Performa\ Efficency}) + \\
 & (W_{portability} \cdot X_{portability}) \\
 & = (0,37.3,05) + (0,23.3,37) + (0,15.3,06) + \\
 & (0,11.3,55) + (0,07.3,08) + (0,04.3,34) + (0,02.2,77) \\
 & = 1,13 + 0,77 + 0,46 + 0,39 + 0,21 + 0,13 + 0,05 \\
 & = 3,14
 \end{aligned}$$

Setelah nilai kuesioner dari pengguna aplikasi akta notaris fidusia diolah dari 26 responden yang meliputi 20 responden admin, 5 responden editor, dan 1 responden owner menggunakan fasilitas tersebut, maka didapatkan nilai akhir setelah dibobotkan dan pengukuran adalah 3,14 dikategorikan *medium* (cukup baik) yang di mana nilai aplikasi lama adalah 2,95.

3. PENUTUP

Berdasarkan pembahasan analisis yang telah dijelaskan maka dapat ditarik kesimpulan dan saran sebagai berikut :

3.1 Kesimpulan

Berdasarkan hasil yang didapatkan dalam penelitian ini, maka diperoleh kesimpulan bahwa:

1. Dapat disimpulkan bahwa aplikasi akta notaris fidusia yang direkomendasikan lebih menarik dan nyaman ketika diakses. Hal ini dilihat dari perbandingan hasil penilaian terhadap perangkat lunak menggunakan standar ISO-25010.
2. Dapat disimpulkan bahwa penggunaan standar kode program memberikan kemudahan bagi tim pengembang untuk melakukan *maintenance*, dan hal ini membuat pekerjaan *maintenance* lebih cepat.

3.2 Saran

Dan untuk saran ke depannya disampaikan sebagai berikut:

1. Untuk Penilaian kualitas bisa dilakukan dari sisi pengguna bukan hanya Aris sisi perangkat lunaknya.
2. Untuk menguji kualitas aplikasi akta notaris fidusia dapat ditambahkan dengan cara melakukan penilaian terhadap tampilan UI/UX nya.

DAFTAR PUSTAKA

- [1] I. Sommerville, Software Engineering, Software Engineering, 2010, pp. 56-81.
- [2] A. Martinez, "Introduction to Object-oriented Programming," *Java.net*, pp. 1-32, 2014.
- [3] E. Fernandi, F. Alfandri, G. A. P. P. O. dan R. Mathias, "Analisis dan Design Berorientasi Objek (OOAD)," no. 52412513, 2014.
- [4] E. Marcotte, Responsive Web Design, 2010, pp. 1-22.
- [5] J. Tian, "What Is Software Quality?," pp. 15-26, 2005.
- [6] D. Galin, Software quality assurance: from theory to implementation, Software Engineering, IEEE, 2004, p. 616.
- [7] "ISO/IEC 25010," [Online]. Available: <http://iso25000.com>. [Diakses 4 September 2016].
- [8] S. Chidamber dan C. Kemerer, "A Metrics Suite for Object Oriented System," pp. 476-493, 1994.
- [9] J. Simarmata, Rekayasa Web, N. WK, Penyunt., Yogyakarta: Andi, 2010.
- [10] R. Pressman, Rekayasa Perangkat Lunak, Yogyakarta: Andi, 2002.
- [11] S. C. Ag, "iso 25010, ISO/IEC JTC1/SC7 N4098," no. Resolution 937, p. 42, 2008.
- [12] S. Wagner, Software product quality control, 2013, pp. 1-210.
- [13] V. R. Basili, G. Caldiera dan R. H. D. , "The goal question metric approach," *Encyclopedia of Software Engineering*, vol. 2, pp. 528-532, 1994.
- [14] P. Antony, "Predicting Reliability of Software Using Thresholds of CK Metrics.," *International Journal of Advanced Networking*, vol. 1785, no. 6, pp. 1778-1785, 2013.
- [15] I. Permana dan Zarnelly, "Penerepan Herarchical Clustering Pada Pengukuran Kohesi Kelas," vol. 12, no. 1, pp. 136-145, 2014.
- [16] R. Seban, "An overview of object-oriented design and C++," *Proceedings of 1994 IEEE Aerospace Applications Conference Proceedings*, no. March, pp. 65-86, 1994.
- [17] Z. A. Hasibuan, "Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi," no. Universitas Indonesia, pp. 1-192, 2007.
- [18] W. G. S. P. Prawita and L. A. A. R. Putri, "Komponen Penilaian Kualitas Perangkat Lunak berdasarkan Software Quality Models," *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2012 (Semantik 2012)*, no. Semantik, pp. 89-94, 2012.
- [19] W. Budiaji, "LIKERT (The Measurement Scale and The Number of Responses in Likert Scale)," vol. 2, no. 1, pp. 125-131, 2013.
- [20] BPS, "Alat Ukur Penelitian Skala Linkert dan Skala Guttman," *Alat Ukur Penelitian*, pp. 1-3, 2010.
- [21] E. Roszkowska, "Rank ordering criteria weighting methods," *Optimum. Studia Ekonomiczne*, vol. 5, no. 5, pp. 1-20, 2013.
- [22] I. Y. Kim dan O. de Weck, "Adaptive Weighted Sum Method for Multiobjective Optimization," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, no. September, pp. 1-13, 2004.
- [23] S. Farrell and J. Nielsen, "User Experience Careers," p. 194, 2014.
- [24] D. Gunawan, "Analisis dan Implementasi Metode Heuristic Evaluation dan Metode Code Refactoring pada Situs Web Museum Sribaduga Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)," *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, pp. 1-8.
- [25] H. Brian dan V. G. Pauline, "Testing Basic For UAT," dalam *User Acceptance Testing- A step - by - step guide*, BCS Learning & Development Limited, 2013.