

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian tentang pengujian perangkat lunak pada aspek *usability* telah banyak dilakukan sebelumnya. *Usability* dapat didefinisikan sebagai tingkat di mana sebuah produk dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu oleh efektif, efisien, dan memperoleh kepuasan dalam konteks penggunaannya (Ghazali, 2007). Penelitian pengujian perangkat lunak dengan standar ISO 9126 untuk mendapatkan penilaian yang terbaik dari kedua aplikasi dan menggunakan metode ISO 9126 yang sebenarnya belum banyak dijadikan bahan penelitian. Penelitian sebelumnya yang akan dijadikan acuan atau landasan dalam pengerjaan penelitian merupakan penelitian dari masing-masing metode yang diujikan untuk menjalankan penelitian.

Dalam makalah Komiyama (Toshihiro, 2008) yaitu mengenai evaluasi usabilitas berdasarkan standar internasional untuk evaluasi kualitas software. Makalah ini berfokus pada kegunaan (*usability*), menjelaskan sudut pandang evaluasi usabilitas dan metrik evaluasi pengujian usabilitas yang didasari dari standar internasional untuk evaluasi kualitas perangkat lunak. Makalah ini menjelaskan evaluasi software menggunakan standar internasional untuk menguji perangkat lunak menggunakan 4 faktor-faktor *usability* dan di setiap faktor tersebut terdapat nama metrik yang diujikan untuk mengetahui nilai pengujian perangkat lunak dari pengguna.

Beberapa penelitian pengujian perangkat lunak pada aspek *usability* yang dijadikan sebagai acuan dalam penelitian ini adalah tentang pengukuran *usability* sistem menggunakan *use questionnaire* pada aplikasi android. Penelitian menggunakan Nielsen model dalam melakukan penelitian. Penelitian ini bertujuan untuk mendapatkan hasil analisis aplikasi android memiliki akseptabilitas berdasarkan kriteria *usability*. Berdasarkan hasil pengujian, didapatkan rata-rata nilai di atas nilai 3 dari skala 5 yang mana aplikasi android memiliki aspek *usability* dan sangat banyak digunakan oleh semua kalangan dikarenakan sangat mudah di pelajari (Rahadi, 2014).

Penelitian selanjutnya tentang pengukuran tingkat ketergunaan (*usability*) sistem informasi keuangan dengan studi kasus *Duta Wacana Internal Transaction* (Duwit). Penelitian yang dilakukan menggunakan Nielsen model untuk menyelesaikannya. Bertujuan untuk mendapatkan seberapa besar tingkatan pemahaman dan kesulitan *user* dalam menggunakan

sistem transaksi melalui antarmuka aplikasi yang sudah dibuat. Pengujian dilakukan dengan menghitung nilai setiap pertanyaan dari faktor-faktor Nielsen model. Berdasarkan pengujiannya dihasilkan tingkat ketergantungan (*usability*) dikatakan baik dengan rata-rata sekitar 72% (Hadiwidjojo & Ernawati, 2016).

Penelitian berikutnya tentang analisis kualitas dan pengembangan sistem informasi akademik berbasis web menggunakan standar ISO 9126. Penelitian ini menguji seluruh aspek dalam pengujian dengan standar ISO 9126, salah satunya adalah aspek *usability*. Dilakukannya pengujian pada *usability* ini bertujuan untuk mendapatkan kelayakan perangkat lunak dari sisi pengguna. Pengujian dilakukan dengan kuesioner menggunakan perhitungan *Alpha Cronbanch* berdasarkan pengujian yang sudah dilakukan diperoleh nilai 74,33% (Novita S, 2016). Perbandingan penelitian *usability* sebelumnya dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan penelitian

Nama Penulis dan Tahun	Standar Pengujian	Aspek Pengujian	Pengujian
(Toshihiro, 2008)	ISO 9126	<i>Usability</i>	Evaluasi usabilitas dengan ISO 9126
(Rahadi, 2014)	Nielsen model	<i>Usability</i>	Pengujian yang dilakukan menggunakan kuesioner dengan faktor penilaian Nielsen model.
(Hadiwidjojo & Ernawati, 2016)	Nielsen Model	<i>Usability</i>	Menghitung dari faktor penilaian Nielsen model
(Novita S, 2016)	ISO 9126	<i>Usability</i>	Perhitungan pengujian <i>usability</i> dengan menggunakan <i>Alpha Cronbanch</i>

Penelitian sebelumnya yang dilakukan dari 4 jurnal yang digunakan sebagai acuan, pengujian usabilitas tidak terpaut dengan dokumen standar ISO 9126. Penelitian tersebut hanya mengambil faktor-faktor dari pengujian usabilitas, tanpa menguji dengan nama metrik dari faktor-faktor yang terdapat dari standar ISO 9126. Sehingga penelitian menggunakan kuesioner dengan pernyataan-pernyataan secara umum dari setiap faktor pengujian usabilitas.

Penelitian yang akan dilakukan mempunyai perbedaan yaitu pengujian dengan menggunakan nama metrik pada setiap faktor usabilitas pada dokumen ISO 9126, sehingga pengujian perlu untuk dijalankan. Pada penelitian ini, pengujian akan dilakukan dengan 2 aplikasi *e-commerce* yang akan digunakan langsung oleh pengguna sebelum menilai aplikasi tersebut.

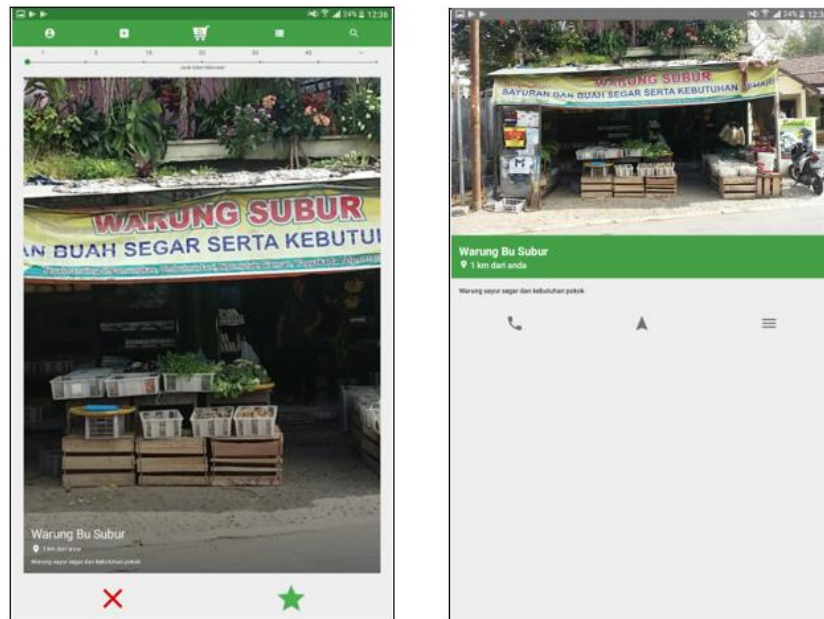
2.2 Aplikasi yang Diuji

Penelitian ini menggunakan aplikasi *e-commerce* untuk pengujian *usability*. *E-commerce* adalah kegiatan-kegiatan bisnis yang menyangkut konsumen (*consumers*), manufaktur (*manufactures*), *service provider*, dan pedagang perantara (*intermediaries*), dengan menggunakan jaringan-jaringan komputer (*computer network*) yaitu Internet (Berkatullah & Prasetyo, 2005). Aplikasi *e-commerce* digunakan sebagai pengujian, dikarenakan tampilan yang disajikan pada aplikasi berpengaruh langsung dengan pengguna aplikasi tersebut. Pengaruh penggunaan aplikasi *e-commerce* terutama pada bagian antarmuka yang menjadikan aplikasi tersebut mudah digunakan oleh pengguna aplikasi. Aplikasi *e-commerce* juga merupakan aplikasi yang banyak digunakan pada saat ini.

E-commerce dengan kata lain merupakan industri teknologi informasi yang sangat pesat, karena dapat dijangkau oleh siapapun pada saat ini. Pada penelitian ini diperlukan 2 aplikasi *e-commerce* untuk melakukan pengujian, adapun aplikasi *e-commerce* yang diujikan pada penelitian ini adalah aplikasi *People-Nearby Application* (PNA) dan OLX.

2.2.1 Aplikasi People-Nearby Application (PNA)

Aplikasi *People-Nearby Application* (PNA) ini merupakan aplikasi yang dikembangkan oleh mahasiswa Teknik informatika yang dikerjakan untuk tugas akhir (Ardiansyah, 2017). Aplikasi ini ditujukan untuk jual beli barang dengan memanfaatkan orang-orang di sekitar pengguna sebagai subyek dan obyek yang ada di dalam aplikasi. Pengguna di sekitar berperan sebagai penyedia layanan kepada sesama pengguna. Aplikasi PNA menggunakan *Location Based Services* (LBS) sebagai acuan untuk pembuatan aplikasi, LBS yaitu layanan yang berbasis lokasi (Schiller, 2004). Aplikasi PNA memudahkan pengguna untuk berbelanja barang di sekitar tempat aplikasi dinyalakan atau digunakan. Berikut merupakan tampilan awal Aplikasi PNA setelah dibuka, gambar dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tampilan Aplikasi PNA

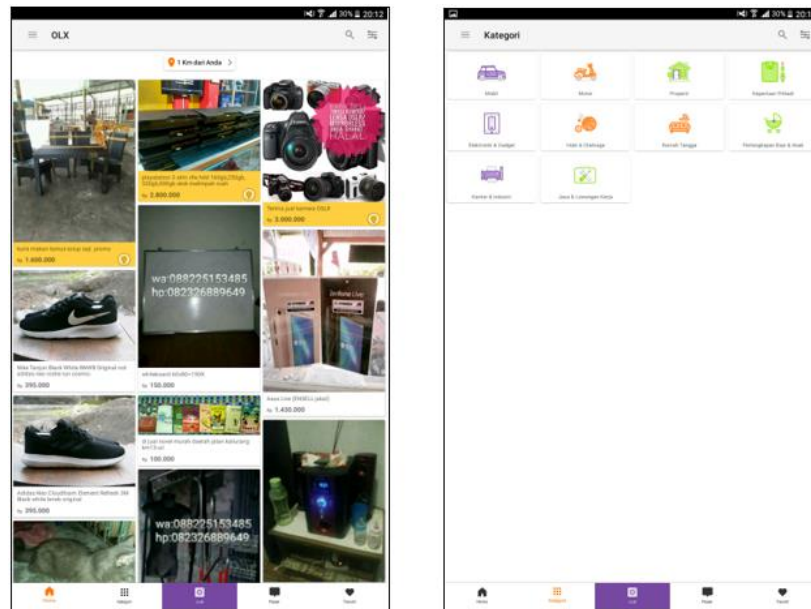
Aplikasi PNA melakukan transaksi dengan cara mencari toko terdekat dengan aplikasi kemudian aplikasi akan menampilkan toko apa saja yang ada di sekitar aplikasi. Kemudian aplikasi memberikan pilihan jika toko tidak diinginkan maka geser gambar toko ke sebelah kiri dan ketika pengguna mendapatkan toko yang tepat, geser foto toko ke sebelah kanan. Pengguna juga dapat menggunakan menu *search* untuk langsung menemukan toko yang diinginkan.

Aplikasi PNA masih memiliki kekurangan yang perlu ditambahkan agar dapat digunakan secara maksimal. Beberapa saran pengembangan aplikasi PNA yang diperlukan adalah terdapat tutorial untuk pengguna yang sudah tidak lagi muda, *flexibilitas* penambahan foto barang, mengganti server dengan server yang lebih kompeten, dan menambah keamanan sistem (Ardiansyah, 2017). Sehingga dari saran tersebut diperlukan pengembangan untuk aplikasi, agar aplikasi berjalan dengan baik.

2.2.2 Aplikasi OLX

OLX adalah perusahaan *e-commerce* yang diluncurkan pada tahun 2014, sebelumnya OLX bernama tokobagus.com yang diluncurkan pada tahun 2005. OLX menyediakan media yang mudah bagi para penjual untuk memasang iklan agar dapat dengan cepat menjual barangnya, sekaligus menjadi media bagi pembeli untuk mencari beragam produk barang bekas dan barang baru untuk kebutuhannya (OLX, 2011). Tujuan dibuatnya aplikasi ini adalah agar mempermudah seluruh pembeli yang ada dan penjual dapat saling bertemu di dalam satu ruang karena jarak yang terpisah dan juga sifat dari internet yang tidak mengenal ruang dan waktu.

Dengan berkembangnya zaman teknologi, OLX memunculkan aplikasi OLX berbasis android untuk semakin memudahkan pengguna aplikasi OLX. OLX menggunakan konsep *hyperlocal* yang mana barang-barang terdekat yang dijual atau ditampilkan ada di sekitar pengguna. Adapun tampilan dari aplikasi OLX terdapat pada Gambar 2.2.



Gambar 2.2 Tampilan Aplikasi OLX

2.3 Pengujian Perangkat Lunak

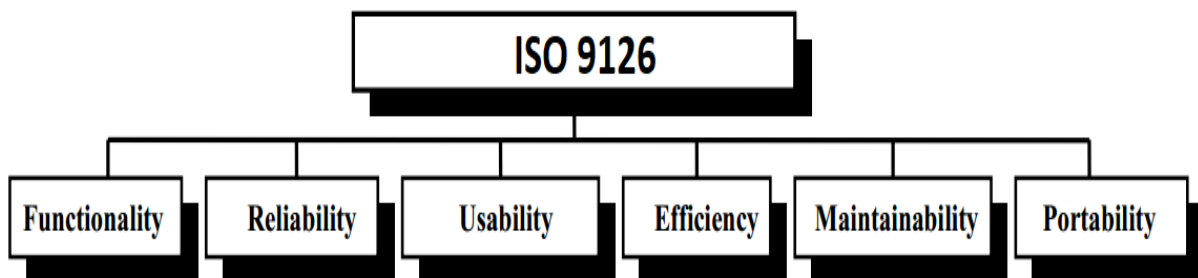
Pengujian dilakukan untuk mengetahui apakah perangkat lunak yang sedang diuji sudah sesuai atau belum dan juga jaminan dari kualitas perangkat lunak tersebut. Pengujian pada salah satu aspek pengujian ISO 9126 yaitu aspek *usability* pada perangkat lunak menjadi penting dikarenakan melibatkan adanya kesalahan manusia, ketidak mampuan manusia melakukan penggunaan fungsi, dan berkomunikasi dengan perangkat lunak tidak berjalan dengan baik. Pengujian perangkat lunak adalah untuk menjalankan sebuah program atau sistem untuk mencari kesalahan (Janner, 2010). Pengujian perangkat lunak menurut Grover (Grover, 2016) menyebutkan ada tingkat pengujian yang berbeda dalam pengujian perangkat lunak. pengujian perangkat lunak yang terpenting adalah pengujian fungsionalitas dan non fungsionalitas.

Pengujian fungsionalitas merupakan perangkat lunak atau sistem dapat menjalankan fungsi dan menampilkannya sedangkan non fungsionalitas adalah hubungan dengan sifat sistem seperti *security*, *privacy*, *usability*, *reliability*, *availability*, dan *performance*. Pengujian

perangkat lunak yang dilakukan pada penelitian ini menggunakan 2 metode yaitu Nielsen model dan ISO 9126.

2.3.1 Standar ISO 9126

ISO 9126 adalah standar internasional untuk melakukan evaluasi perangkat lunak. Standar ini dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. Al-qutaish (Al-qutaish, 2010) menyebutkan bahwa standar ISO 9126 memiliki 6 aspek pengujian, aspek pengujian seperti pada Gambar 2.3.

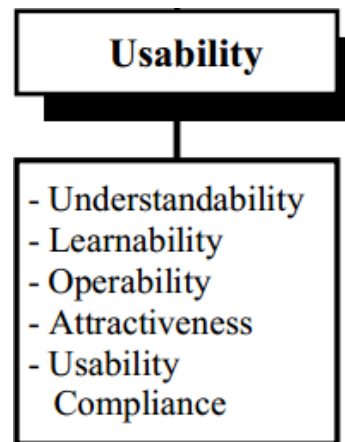


Gambar 2.3 Aspek pengujian pada standar ISO 9126

ISO 9126 memiliki 6 aspek untuk pengujian perangkat lunak agar mendapatkan evaluasi penuh. *Functionality* yaitu kemampuan perangkat lunak untuk menyediakan fungsi yang memenuhi syarat kebutuhan dalam kondisi tertentu, *reliability* berkaitan dengan kemungkinan suatu perangkat lunak dapat melakukan fungsi-fungsi tertentu sesuai dengan tingkat ketelitian yang diinginkan, *usability* digunakan untuk menilai seberapa mudah tampilan antar muka suatu produk untuk digunakan dan *usability* mengacu pada metode untuk meningkatkan kemudahan penggunaan selama proses desain, *efficiency* kemampuan perangkat lunak untuk menampilkan performa relatif terhadap penggunaan sumberdaya, *maintainability* sebagai usaha yang diperlukan untuk mencari dan membetulkan kesalahan pada sebuah program, sedangkan *portability* sebagai kemudahan sebuah perangkat lunak dapat dipindahkan dari suatu lingkungan ke lingkungan lain dengan mengacu pada indikator.

Aspek *usability* berfokus pada bagaimana tampilan dari perangkat lunak mudah dipelajari dan digunakan serta mempermudah dalam menyelesaikan pekerjaan. Aspek *usability* memiliki 4 faktor untuk pengujian yaitu *understandability* yaitu bagaimana suatu perangkat lunak mampu dipahami dengan mudah, *learnability* bagaimana perangkat lunak mudah untuk dipelajari, *operability* bagaimana perangkat lunak mudah dioperasikan atau digunakan dengan mudah, *attractiveness* yaitu bagaimana perangkat lunak itu menarik pengguna, dan *usability*

compliance yaitu untuk mengetahui kepatuhan kegunaan. Faktor-faktor pengujian *usability* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Faktor pengujian *usability* pada ISO 9126

Pada pengujian usabilitas dengan standar ISO 9126 pada setiap faktor memiliki nama metrik yang diujikan. Di setiap nama metrik yang diujikan terdapat rumus yang berbeda-beda untuk pengolahan data dari pengujian yang dilakukan. Berikut merupakan rumus-rumus yang ada pada setiap nama metrik untuk menghitung hasil pengujian aspek usabilitas.

a. Nama Metrik Pengujian Faktor Understandability

Pengujian faktor *understandability* memiliki 7 nama metrik yang diujikan. Berikut adalah nama metrik dan rumus untuk menghitung pengujiannya:

1. *Completeness of description* menghitung berapa fungsi yang dipahami oleh pengguna. Pengujian yang akan dilakukan seperti memberikan tugas kepada pengguna sesuai dengan fungsi yang ada kemudian fungsi tersebut diuji oleh pengguna dan dilihat hasil pengujiannya. Nilai pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$\text{CoD} = \frac{\text{jumlah fungsi yang dipahami}}{\text{total jumlah fungsi}} \quad (2.1)$$

2. *Demonstration accessibility* menghitung jumlah fungsi yang berhasil dijalankan setelah melihat tutorial. Pengujian yang akan dilakukan pengguna melihat tutorial pada aplikasi, serta menjalankan fungsi yang ada pada tutorial yang di sediakan. Nilai

pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$DA = \frac{\text{jumlah fungsi yang dipahami}}{\text{jumlah fungsi yang ada}} \quad (2.2)$$

3. *Demonstration accessibility in use* menghitung berapa kali pengguna membutuhkan tutorial selama menjalankan aplikasi. Pengujian yang akan dijalankan dengan menanyakan fungsi pada aplikasi perlu dilihat kembali tutorialnya. Nilai pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$DAU = \frac{\text{jumlah pengguna berhasil kembali melihat tutorial}}{\text{jumlah kasus pengguna kembali melihat tutorial}} \quad (2.3)$$

4. *Demonstration effectiveness* menguji seberapa paham pengguna dengan adanya tutorial. Pengujian yang akan dijalankan yaitu dengan pertanyaan sesuai dengan penjelasan nama metrik. Nilai pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$DE = \frac{\text{jumlah fungsi yang dioperasikan dengan sukses}}{\text{jumlah tutorial yang dipahami}} \quad (2.4)$$

5. *Evident functions* menghitung jumlah fungsi yang dipahami oleh pengguna. Pengujian yang dilakukan dengan memberikan pertanyaan sesuai penjelasan nama metrik. Nilai pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$EF = \frac{\text{jumlah fungsi yang dikenali pengguna}}{\text{jumlah fungsi yang ada}} \quad (2.5)$$

6. *Function understand-ability* menghitung jumlah fungsi antarmuka yang mudah dipahami oleh pengguna. Pengujian yang akan dilakukan dengan memberikan pertanyaan sesuai fungsi antar muka pada aplikasi. Nilai pengujian yang didapat lebih

dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$FU = \frac{\text{jumlah fungsi antarmuka yang benar}}{\text{total jumlah fungsi antarmuka yang tersedia}} \quad (2.6)$$

7. *Understandable input and output* menghitung jumlah item *input* dan *output* data yang dipahami oleh pengguna. Pengujian yang dilakukan dengan cara memberikan pertanyaan sesuai dengan item input atau output yang ada pada aplikasi. Nilai pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$U - I/O = \frac{\text{jumlah input dan output data yang dipahami}}{\text{total jumlah input dan output yang tersedia}} \quad (2.7)$$

b. Nama Metrik Pengujian Faktor Learnability

Pengujian faktor *learnability* memiliki 5 nama metrik yang diujikan. Berikut adalah nama metrik dan rumus untuk menghitung pengujiannya:

1. *Ease of function learning* menghitung berapa lama pengguna untuk belajar menggunakan fungsi pada aplikasi. Cara yang digunakan untuk menguji metrik ini yaitu dengan menghitung durasi pengguna mempelajari fungsi pada aplikasi. Jika waktu yang didapat sedikit maka hasil lebih baik.
2. *Ease of learning to perform a task in use* menghitung berapa lama pengguna menggunakan aplikasi dengan lancar. Cara yang digunakan untuk menguji metrik ini yaitu dengan menghitung durasi pengguna mempelajari fungsi pada aplikasi. Jika waktu yang didapat sedikit maka hasil lebih baik.
3. *Effectiveness of the user documentation and/or help system* menghitung berapa tugas yang diselesaikan oleh pengguna setelah melihat tutorial. Pengujian yang akan dilakukan berupa memberikan tugas berupa pertanyaan sesuai penjelasan nama metrik. Nilai pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$EHS = \frac{\text{jumlah tugas yang dipahami}}{\text{jumlah tugas yang ada}} \quad (2.8)$$

4. *Help accessibility* menghitung jumlah bantuan online yang pengguna lakukan. Pengujian yang dilakukan adalah memberikan tugas kepada pengguna berupa pertanyaan sesuai penjelasan nama metrik. Nilai pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$HA = \frac{\text{jumlah fungsi yang memberikan bantuan}}{\text{jumlah fungsi yang ada}} \quad (2.9)$$

5. *Help frequency* menghitung penggunaan tutorial kembali oleh pengguna pada aplikasi untuk menyelesaikan tugas-tugas selama menjalankan pengujian faktor *learnability*. Nilai pengujian akan lebih baik jika nilai lebih sedikit dari 0.

c. Nama Metrik Pengujian faktor *Operability*

Pengujian faktor *operability* memiliki 12 nama metrik yang diujikan. Berikut adalah nama metrik dan rumus untuk menghitung pengujiannya:

1. *Operational consistency in use* memberikan pertanyaan tentang konsistensi dari user interface pada aplikasi. Nilai pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$OCU = \frac{\text{jumlah fungsi yang tidak sesuai dengan pengguna}}{\text{jumlah fungsi}} \quad (2.10)$$

2. *Error correction* menghitung waktu penyelesaian kesalahan jika pengguna melakukannya. Pengujian yang dilakukan yaitu memberikan waktu kepada pengguna untuk penyelesaian kesalahan. Hasil akan lebih baik jika waktu yang di dapat semakin pendek. Rumus untuk menghitungnya adalah sebagai berikut:

$$EC = \text{waktu menyelesaikan kesalahan} - \text{waktu memulai kesalahan} \quad (2.11)$$

3. *Error correction in use* yaitu pengguna dengan mudah memulihkan masukkan yang dilakukan. pengujian yang dilakukan dengan menghitung berapa kali pengguna melakukan pemulihan sehingga berhasil dan membandingkan dengan jumlah fungsi masukkan yang ada. nilai hasil pengujian lebih dekat dengan 1 maka hasil lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$EDU = \frac{\text{jumlah fungsi yang berhasil diperbaiki}}{\text{jumlah fungsi}} \quad (2.12)$$

4. *Default value availability in use* ialah pengguna dengan mudah memilih nilai parameter untuk operasi yang mudah. Pengujian yang dilakukan dengan memberikan pertanyaan mengenai parameter pada fungsi yang ada dan menghitung berapa kali pengguna mencoba. Nilai hasil pengujian yang didapat lebih dekat dengan 1, maka hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$DV = 1 - \frac{\text{jumlah pengguna gagal memilih nilai parameter}}{\text{jumlah upaya pemilihan parameter}} \quad (2.13)$$

5. *Message understand-ability in use* merupakan pengamatan terhadap waktu pengguna dalam memahami pesan dari aplikasi yang menyebabkan pengguna terhambat dalam penggunaan aplikasi. Pengujian yang dilakukan dengan mengukur waktu penggunaan aplikasi. Pengujian akan lebih baik jika hasilnya lebih kecil dari 0. Rumus untuk menghitungnya adalah sebagai berikut:

$$MU = \frac{\text{jumlah pengguna berhenti karena pesan tidak dipahami}}{\text{waktu operasi pengguna}} \quad (2.14)$$

6. *Self-explanatory error messages* yaitu kondisi kesalahan yang dapat pengguna usulkan untuk tindakan pemulihan yang benar. Pengujian yang dilakukan dengan memberikan pertanyaan dari fungsi yang ada kemudian diujikan. Pengujian akan lebih baik jika hasil lebih dekat dengan 1. Rumus untuk menghitungnya adalah sebagai berikut:

$$SEM = \frac{\text{jumlah fungsi yang diusulkan pengguna}}{\text{total jumlah fungsi yang diuji}} \quad (2.15)$$

7. *Operational error recoverability in use* yaitu pengguna dapat dengan mudah memulihkan keadaan yang buruk dari aplikasi. Pengujian yang dilakukan dengan memberikan tugas dan mengamati pengguna selama melakukan tugas yang diberikan. Pengujian akan lebih baik jika hasil lebih dekat dengan 1. Rumus untuk menghitungnya adalah sebagai berikut:

$$\text{OER} = 1 - \frac{\text{jumlah situasi yang tidak berhasil pulih oleh pengguna}}{\text{jumlah kesalahan pengguna}} \quad (2.16)$$

8. *Time between human error operations in use* yaitu dapatkah pengguna mengoperasikan aplikasi cukup lama tanpa ada kesalahan. Pengujian yang dilakukan dengan mengamati waktu penggunaan aplikasi dan melihat berapa waktu yang didapatkan ketika mengalami kesalahan saat menggunakan aplikasi yang diujikan. pada nama metrik ini memiliki nilai Batasan pengujian yaitu ketika hasil pengujian lebih baik jika hasilnya lebih besar dari 0. Rumus untuk menghitungnya adalah sebagai berikut:

$$\text{TBH} = \frac{\text{waktu pengujian}}{\text{total kesalahan kesalahan selama operasi}} \quad (2.17)$$

9. *Undoability (User error correction)* menghitung berapa pengguna melakukan kesalahan input pada fungsi tertentu dan berhasil mengoreksi kesalahan. Hasil untuk pengujian ini akan lebih baik jika hasil yang didapat dekat dengan nilai 1. Rumus untuk menghitungnya adalah sebagai berikut:

$$U = \frac{\text{jumlah kesalahan input berhasil dikoreksi}}{\text{total upaya memperbaiki kesalahan input}} \quad (2.18)$$

10. *Customisability* menghitung berapa fungsi yang nyaman digunakan oleh pengguna. Pengujian yang dilakukan dengan memberikan tugas menjalankan fungsi dan menilai seberapa nyaman pengguna untuk menggunakan fungsi yang dijalankan. Hasil untuk pengujian ini akan lebih baik jika hasil yang didapat dekat dengan nilai 1. Rumus untuk menghitungnya adalah sebagai berikut:

$$C = \frac{\text{jumlah fungsi yang nyaman digunakan}}{\text{total jumlah fungsi disediakan}} \quad (2.19)$$

11. *Operation procedure reduction* menghitung prosedur saat menjalankan fungsi tertentu. Pengujian yang dilakukan adalah memberikan tugas kepada pengguna seperti menjalankan fungsi tertentu dengan prosedur yang memudahkan pengguna atau sesuai kenyamanan pengguna. Hasil untuk pengujian ini akan lebih baik jika hasil yang didapat dekat dengan nilai 1. prosedur Rumus untuk menghitungnya adalah sebagai berikut:

$$OPR = 1 - \frac{\text{jumlah prosedur operasi yang baru}}{\text{total jumlah prosedur asli}} \quad (2.20)$$

12. *Physical accessibility* menghitung berapa fungsi yang berhasil digunakan oleh pengguna dengan kekurangan fisik. Rumus untuk menghitungnya adalah sebagai berikut:

$$PA = \frac{\text{jumlah fungsi yang dipahami}}{\text{total jumlah fungsi}} \quad (2.21)$$

d. Nama Metrik Pengujian Faktor Attractiveness

Pengujian faktor *attractiveness* memiliki 2 nama metrik yang diujikan. Berikut adalah nama metrik dan rumus untuk menghitung pengujiannya:

1. *Interface appearance customisability* menghitung kepuasan pengguna dari antarmuka yang ada pada aplikasi. Pengujian yang dilakukan yaitu memberikan pernyataan mengenai antarmuka aplikasi yang ada dan melihat kepuasan pengguna terhadap antarmuka aplikasi. Nilai pengujian yang didapat lebih dekat dengan 1, hasil pengujian lebih baik. Rumus untuk menghitungnya adalah sebagai berikut:

$$IAC = \frac{\text{jumlah antarmuka yang di sukai oleh pengguna}}{\text{total jumlah antarmuka yang tersedia}} \quad (2.22)$$

2. *Attractive interaction* yaitu bagaimana antar muka pada aplikasi menarik pengguna. Pada metrik *attractive interaction* penguji diminta mengisi kuesioner dengan 5 skala. Jawaban yang diberikan berdasarkan apa yang dirasakan selama menggunakan

aplikasi untuk mengukur kesukaan pengguna kepada aplikasi yang diuji. Berikut merupakan keterangan dan bobot dari setiap skala penilaiannya dapat dilihat pada Tabel 2.2.

Tabel 2.2 Keterangan dan bobot penilaian kuesioner

Nilai	Keterangan	Bobot
STS	Sangat Tidak Setuju	1
TS	Tidak Setuju	2
N	Netral	3
S	Setuju	4
SS	Sangat Setuju	5

Pengujian metrik dengan menggunakan skala likert memerlukan beberapa tahapan untuk mendapatkan hasil akhir agar mengetahui seberapa baik tanggapan pengguna terhadap kedua aplikasi yang diujikan. Penilaian pada nama metrik ini dilakukan pada setiap responden. Pada satu responden akan langsung dihitung untuk mengetahui berapa nilai kepuasan terhadap aplikasi. Untuk mendapatkan nilai kepuasan dengan bentuk persentase diperlukan perhitungan sebagai berikut:

$$= \text{hasil penilaian} \times \text{rentang skala} \times 100\% \quad (2.23)$$

Nilai hasil penilaian yang dapat diperoleh dari setiap bobot dari skala penilaian pada masing-masing pernyataan. Perhitungann untuk memperoleh hasil penilaian adalah sebagai berikut:

$$= \frac{\sum \text{jumlah bobot setiap pernyataan}}{\text{total pernyataan}} \quad (2.24)$$

Setelah mendapatkan nilai kepuasan pengujian dari responden, terdapat kategori penilaian untuk mngetahui hasil nilai yang didapatkan termasuk kedalam kategori berdasarkan rentang nilai pengujian. Kategori hasil penilaian kuesioner dapat dilihat Tabel 2.3.

Tabel 2.3 Kategori penilaian kuesioner

Skor Pengujian	Kategori Penilaian
0% - 19,99%	Sangat Kurang
20% - 39,99%	Kurang
40% - 59,99%	Cukup
60% - 79,99%	Baik
80% - 100%	Sangat Baik

e. Pengujian faktor *Usability Compliance*

Pengujian faktor *usability compliance* bertujuan untuk mengetahui tingkat kepatuhan aplikasi memenuhi praturan, standar, dan konvensi yang ada. pengujian dilakukan dengan menghitung jumlah metrik yang belum terpenuhi dibandingkan dengan jumlah kasus yang diharapkan dapat terpenuhi. Rumus untuk menghitungnya adalah sebagai berikut:

$$UC = 1 - \frac{\text{jumlah metrik yang belum diimplementasikan}}{\text{jumlah metrik yang ada}} \quad (2.24)$$

2.4 Kuesioner

Penelitian ini menggunakan kuesioner untuk melakukan pengujian. Kuesioner dianggap sebagai media penilaian paling tepat untuk menjalankan pengujian *usability* untuk perangkat lunak dengan metode ISO 9126. Kuesioner merupakan daftar pertanyaan tertulis yang diberikan kepada subjek yang diteliti untuk mengumpulkan informasi yang dibutuhkan penulis (Kusumah & Dedi, 2011). Menggunakan kuesioner bertujuan untuk mendapatkan penilaian saat proses pengujian. Berikut merupakan contoh dari kuesioner pengujian yang diperlukan, yang terdapat pada Tabel 2.4 dan Tabel 2.5.

Tabel 2.4 Contoh Kuesioner

No	Pernyataan	Pengujian		Ket
		Ya	Tidak	
Nama metrik : Deskripsi				
1				
2				
3				

Tabel 2.5 Contoh Kuesioner dengan skala Likert

No	Pernyataan	Penilaian				
		STS	TS	N	S	SS
1						
2						
3						

Kuesioner pada umumnya bermodel tabel. Kuesioner yang terdiri dari baris dan kolom, pada kolom pertama berisi pernyataan yang sesuai dengan kebutuhan penilaian, kemudian kolom selanjutnya berisi tentang skala nilai untuk mengetahui nilai dari setiap pernyataan yang disajikan. Penelitian ini membutuhkan dua tipe skala untuk pengujian aspek usability. Kuesioner pertama dibentuk dalam skala dua poin dengan jawaban ya dan tidak, yang dianggap cocok untuk menghitung rumus di setiap nama metrik. Kuesioner kedua dibentuk dalam skala lima poin dengan model skala *Likert* untuk memilih jawaban sebagai pengukuran tingkat persetujuan pengguna terhadap pernyataan.

Pengujian usability ini memerlukan observasi oleh penulis untuk memenuhi beberapa nama metrik yang berkaitan dengan waktu selama penggunaan aplikasi. Sehingga penelitian ini terdapat dua pengujian, yaitu pengujian ditempat yang mana kuesioner akan langsung diisi oleh penguji sesuai apa yang dialami selama penggunaan aplikasi dan observasi yang dilakukan penulis dengan mengamati waktu selama pengujian dijalankan.