

Analisis Uji Kualitas Aplikasi Arjuna Laundry Menggunakan Model SQO – OSS

Rasyidah Aisy Ariyanto
Politeknik Negeri Jember
Jember, Indonesia
E41220015@student.polije.ac.id

Rani Purbaningtyas
Politeknik Negeri Jember
Jember, Indonesia
rpurbaningtyas@polije.ac.id

Mochamad Lury Choirul Rizki
Politeknik Negeri Jember
Jember, Indonesia
E41221566@student.polije.ac.id

Mochammad Rifky Ulil Albaab
Politeknik Negeri Jember
Jember, Indonesia
mochrifki@polije.ac.id

Abstract— Kemajuan teknologi mempengaruhi sektor jasa laundry, yang kini membutuhkan solusi efisien. Aplikasi Arjuna Laundry hadir sebagai solusi inovatif guna mempermudah konsumen dalam memesan layanan laundry secara online dan dapat memantau status cucian secara *real-time*. Untuk memastikan kualitas perangkat lunak, aplikasi ini menjalani pengujian kualitas perangkat lunak menggunakan model SQO-OSS dengan metode *profile matching* sebagai acuan perhitungan secara konkrit. Aplikasi Arjuna Laundry menunjukkan hasil sangat baik dengan rata – rata 76% dengan kriteria “Sangat Baik”. Dengan melakukan uji kualitas menggunakan model ini, Aplikasi Arjuna Laundry dapat memastikan bahwa penggunaannya tidak hanya nyaman, tetapi juga aman dan dapat diandalkan. Dengan demikian, penerapan model SQO-OSS dalam pengujian kualitas menjadi langkah krusial dalam meningkatkan performa serta kepercayaan pengguna terhadap Aplikasi Arjuna Laundry.

Kata kunci —Aplikasi Arjuna Laundry, Kualitas, Perangkat Lunak, Profile Matching, SQO – OSS, Teknologi

I. PENDAHULUAN

Kemajuan teknologi dan informasi yang pesat saat ini telah memudahkan dalam mengakses informasi secara akurat, cepat dan terpercaya tanpa terikat jarak dan waktu [1]. Sistem informasi dapat diartikan sebagai bagian dari sistem organisasi yang mewakili kombinasi pengguna dan sumber daya yang tersedia seperti teknologi dan media pengendalian informasi. Hal ini dimaksudkan untuk menjaga saluran komunikasi, memproses jenis transaksi dan mengirimkan sinyal tingkat manajemen sebagai dasar informasi. Dalam membuat keputusan dan menerima informasi sangat memudahkan pengguna [2]. Dalam lingkungan bisnis yang berorientasi pada pelanggan, sistem informasi dapat meningkatkan layanan timbal balik dengan menyediakan informasi yang akurat dan cepat kepada pelanggan serta memungkinkan komunikasi yang efektif. Departemen dapat diintegrasikan dan disinkronkan melalui implementasi fungsi dalam suatu organisasi sehingga memungkinkan koordinasi yang lebih baik antar unit kerja [3].

Usaha kecil dan menengah (UMKM) yang dikelola dengan memasukkan teknologi informasi dan komunikasi ke dalam proses bisnisnya akan memudahkan para konsumen dalam menunjang kegiatan usahanya [4]. Pengenalan teknologi informasi dan komunikasi dalam hal menjadikan bisnis lebih mudah, cepat dan dapat diandalkan, meminimalisir kesalahan manusia. Perusahaan laundry adalah suatu perusahaan yang bergerak di bidang jasa laundry [5]. Perusahaan ini termasuk dalam kategori perusahaan turnover tinggi. Selain itu, laundry juga merupakan bisnis yang berkelanjutan atau selalu dibutuhkan oleh banyak orang. Manusia akan selalu mencuci pakaian selama pakaian merupakan kebutuhan utama manusia [6]. Hal ini, menjadikan peluang bisnis di bidang laundry sangat menjanjikan. Dengan adanya aplikasi Arjuna Laundry, segala sesuatu yang dapat diatur secara sistematis menjadi lebih maksimal dan system manual lambat laun digantikan oleh teknologi yang semakin canggih [7].

Arjuna Laundry adalah usaha yang dijalankan oleh perorangan sesuai dengan tujuan dalam penyediaan jasa yang berinteraksi secara langsung. Untuk jenis laundry memiliki servis cuci basah, cuci kering lipat, cuci setrika, setrika, cuci express. Jasa laundry ini bukan hanya sekedar tempat mencuci melainkan tempat perawatan pakaian agar lebih bersih dan awet, faktor serba instant serta praktis yang menjadi trend di masyarakat saat ini [8]. Rentang waktu antara permintaan layanan pertama dan permintaan berikutnya relative singkat. Jika pakaian daur ulang menjadi kotor dan perlu dicuci, pelanggan akan segera menggunakan layanan laundry ini kembali. Hal ini menjadikan jasa laundry menjadi solusi yang sangat tepat dalam menghadapi kesibukan dan tuntutan kehidupan modern [9].

Namun, tantangan utama Arjuna Laundry meliputi pengumpulan data secara manual dan efisiensi operasional yang terbatas. Proses manual sering kali mengakibatkan kesalahan entri data, penundaan jadwal dan rentang waktu yang tidak akurat. Untuk mengatasi tantangan tersebut, Arjuna Laundry memutuskan untuk menerapkan solusi digitalisasi teknologi. Dengan adanya aplikasi Arjuna Laundry sangat membantu para customer dalam meningkatkan efisiensi waktu dan tenaga serta memfasilitasi pembayar digital. Oleh karena itu, aplikasi Arjuna Laundry tidak hanya memberikan kenyamanan tetapi juga pengalaman yang lebih modern dan efisiensi kepada penggunaannya.

Akan tetapi, sebelum memutuskan untuk melaksanakan pembaharuan menggunakan aplikasi Arjuna Laundry, langkah yang perlu diambil adalah melakukan uji coba terlebih dahulu. Uji coba ini bertujuan untuk memastikan bahwa penggunaan aplikasi tersebut sesuai dengan kebutuhan dan dapat memberikan manfaat yang diharapkan. Dengan melakukan uji coba, dapat dievaluasi kemampuan aplikasi dalam meningkatkan efisiensi operasional, kualitas layanan serta kemudahan pengguna bagi pelanggan dan karyawan. Dalam

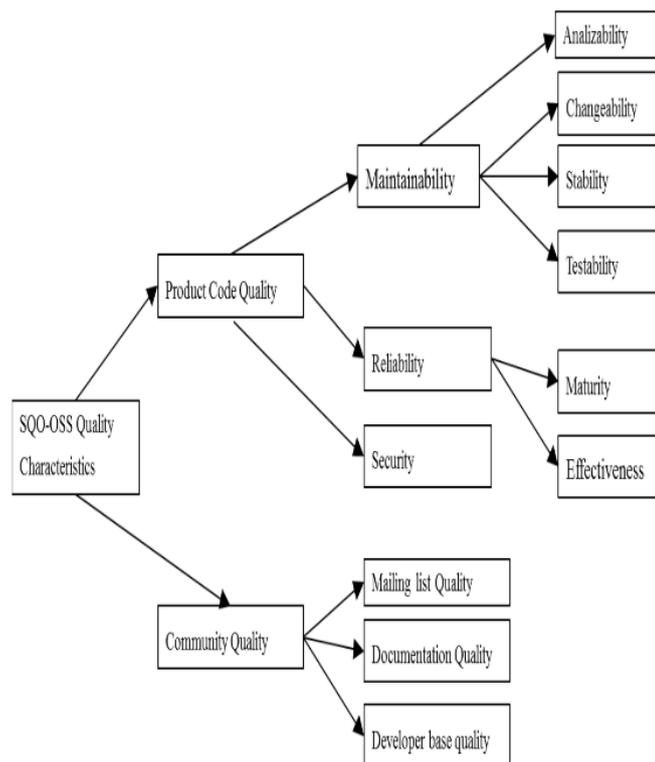
tahap uji coba kami menggunakan model SQO – OSS untuk memastikan bahwa aplikasi ini beroperasi secara optimal dan sesuai dengan standar yang ditetapkan [10]. Ada lima factor utama dalam pengukuran ini, yaitu Maintainability, Reliability, Security, Mailing list Quality, Documentation Quality, dan Developer Base Quality [10]. Dengan demikian, penerapan model SQO-OSS dalam pengujian kualitas menjadi langkah krusial dalam meningkatkan performa serta kepercayaan pengguna terhadap Aplikasi Arjuna Laundry.

II. METODOLOGI PENELITIAN

A. Model SQO – OSS (Heading 2)

Model SQO – OSS adalah model hierarki yang memungkinkan otomatisasi proses komunitas untuk mengevaluasi kode sumber dan menghitung metrik [11]. Modelnya ditunjukkan pada gambar dan menurut, model ini berbeda dengan model lainnya dalam hal berikut :

- Berfokus pada otomatisasi, tidak seperti model lain yang memerlukan banyak intervensi pengguna.
- Inti dari sistem pemantauan kualitas berkelanjutan dan memungkinkan pengumpulan metrik otomatis.
- Fitur tidak dievaluasi.
- Fokusnya adalah pada kode sumber. Kode sumber adalah bagian terpenting dari proyek perangkat lunak.
- Pertimbangkan hanya faktor masyarakat yang dapat di ukur secara otomatis.



Gambar 1. Model SQO - OSS

Indicator model SQO – OSS memiliki enam karakteristik yakni [11] [12] [13] :

- a) *Functionality* : Kemampuan perangkat lunak dalam tahap perbaikan dan adaptasi terhadap spesifikasi fungsional
- b) *Reliability* : Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja dalam kondisi tertentu
- c) *Security* : Kemampuan perangkat lunak yang berfungsi untuk memodifikasi data otorisasi
- d) *Mailing List Quality* : Kemampuan perangkat lunak dalam komunikasi secara keseluruhan untuk meningkatkan efisiensi yang berkaitan dengan proyek
- e) *Documentation Quality* : Dokumentasi mencakup panduan dengan informasi tentang proyek perangkat lunak yang dapat digunakan dan dipahami
- f) *Developer Base Quality* : Fasilitator komunikasi antar kelompok proyek mengenai pemeliharaan proyek

Setiap atribut kualitas perangkat lunak akan dibagi menjadi beberapa sub indikator seperti pada tabel dibawah ini [14] [15]. berdasarkan tabel dibawah ini, termasuk bagian indikator functionality.

Tabel 1. Sub Indikator Functionality

Sub - Karakteristik	Deskripsi
Analizability	Kemampuan perangkat lunak yang menganalisis penyebab eror / bug.
Changeability	Kemampuan perangkat lunak dalam proses perubahan suatu project.
Stability	Kemampuan perangkat lunak dalam meminimalisir perubahan yang non revisi
Testability	Kemampuan perangkat lunak yang di uji guna memvalidasi perangkat lunak.

Berdasarkan tabel 2 dibawah ini, termasuk bagian indikator reliability.

Tabel 2. Sub Indikator Reliability

Sub – Karakteristik	Deskripsi
Maturity	Kemampuan perangkat lunak untuk menghindari kesalahan eror / bug.
Effectiveness	Kemampuan perangkat lunak dalam kinerja project yang relative mudah digunakan dan dipahami.

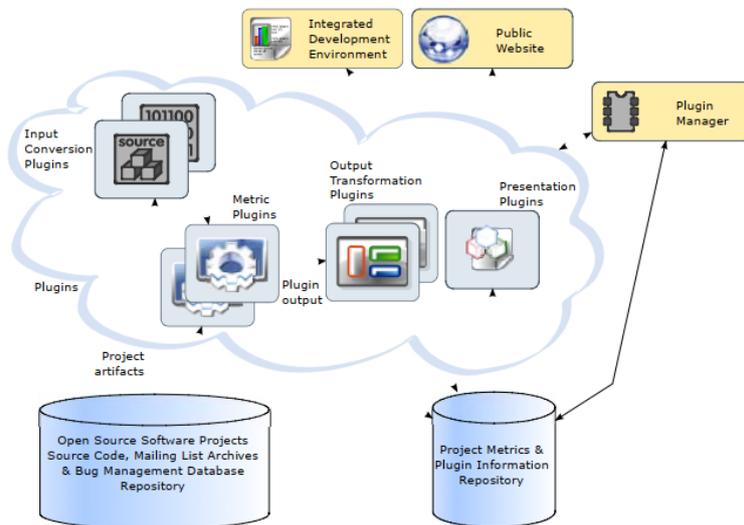
B. Desain SQO – OSS

Desain SQO – OSS didasarkan pada tiga property non – fungsional yaitu [16] :

- *Simplicity*
- *Extensibility*
- *Interopability*

Implementasi SQO-OSS akan didasarkan pada *plugin*, yang terdiri dari komponen-komponen melalui penyimpanan data umum. Segala sesuatu bias menjadi *plugin*, artinya bisa kita lihat pada gambar di bawah ini [17] :

- *Plugin* untuk perhitungan metrik, pemrosasan data dan pemrosesan hasil
- Manajer *plugin* untuk penyisipan, penghapusan dan pembaharuan *plugin*
- Basis data untuk melacak *plugin*, pengukuran dan data mentah



Gambar 2. Our Metric SQO - OSS

C. Profile Matching

Profile Matching merupakan proses yang sangat penting dalam pengelolaan sumber daya manusia yang terlebih dahulu menentukan kompetensi (*skill*) yang dibutuhkan suatu jabatan kompetensi tersebut yang wajib dipenuhi oleh calon pemegang jabatan [18]. *Profile Matching* merupakan proses yang sangat penting dalam pengelolaan sumber daya manusia yang terlebih dahulu menentukan daya saing (*skill*) yang dibutuhkan suatu posisi [19]. Tingkat penilaian setiap *Core Factor* dan *Secondary Factor* ditentukan pada skala 1 sampai 4. Berikut langkah – langkah dan rumusan perhitungan menggunakan metode *Profile Matching* menurut [20] .

a) GAP

Pada tahap ini ditentukan bobot nilai setiap aspek berdasarkan bobot GAP yang telah ditentukan.

b) Pengelolaan Core Factor dan Secondary Factor

Pengelompokkan dari *Core Factor* dan *Secondary Factor* telah ditentukan nilai bobot GAP pada masing – masing sub indikator. Oleh karena itu, setiap sub indicator dikelompokkan menjadi dua bagian yaitu *Core Factor* dan *Secondary Factor*.

- *Core Factor*
Core Factor merupakan elemen utama yang dibutuhkan. Rumus perhitungan *Core Factor* [21] :

$$NCF = \frac{\sum NC}{\sum IC} \quad (1)$$

Keterangan :

NCF : Nilai rata – rata *Core Factor*
 NC : Jumlah nilai *Core Factor*
 IC : Jumlah item *Core Factor*

- *Secondary Factor*
Secondary Factor merupakan aspek penunjang dari *Core Factor*. Rumus perhitungan *Secondary Factor* [21] :

$$NSF = \frac{\sum NS}{\sum IS} \quad (2)$$

NSF : Nilai rata – rata *Secondary Factor*
 NS : Jumlah nilai *Secondary Factor*
 IS : Jumlah item *Secondary Factor*

c) *Perhitungan Nilai Total*

Dari hasil perhitungan *Core Factor* dan *Secondary Factor* sangat berpengaruh dengan uji kualitas aplikasi Arjuna Laundry. Rumus perhitungan nilai total [21]:

$$\text{Nilai Total (NT)} = 60\% \text{ NCF} + 40\% \text{ NSF} \quad (3)$$

Keterangan :

NT : Nilai total
 NCF : Nilai rata – rata *core factor*
 NSF : Nilai rata – rata *secondary factor*
 (x%) : Nilai persen yang di inputkan

d) *Perhitungan Nilai Responden*

Hasil akhir perangkungan nilai responden.. Rumus perhitungan nilai total responden [21] :

$$\text{Nilai Total Responden} = \sum_{i=1}^{i=6} \text{ bobot } i * NT_i \quad (4)$$

Keterangan :

NT : Hasil Akhir
 I : Nilai kapasitas
 Bobot : Skala likert

III. HASIL DAN PEMBAHASAN

Penelitian ini dimulai dengan tahap pertama, membuat pertanyaan yang sesuai dengan kebutuhan Aplikasi Arjuna Laundry dengan menggunakan model SQO – OSS. Kuisisioner tersebut berisi pertanyaan yang mewakili sub indicator SQO – OSS. Kedua, penyebaran kuisisioner terhadap 6 responden. Ketiga, setiap indicator yang dinilai oleh responden akan diklasifikasikan dalam skala likert yang memiliki 4 nilai luaran. Keempat, hasil data kuisisioner akan di uji coba menggunakan metode *Profile Matching*. Kelima, mengklasifikasikan nilai responden sesuai dengan variable yang sama berdasarkan GAP. Keenam, menganalisa hasil data tersebut untuk mengetahui tingkat kepuasan pengguna dalam pemakaian Aplikasi Arjuna Laundry.

Tahap pertama, membuat pertanyaan yang sesuai dengan kebutuhan Aplikasi Arjuna Laundry menggunakan model SQO – OSS. Kuisisioner tersebut berisi pertanyaan yang mewakili sub indicator SQO – OSS. Berikut daftar pertanyaan pada table di bawah ini :

Tabel 3. List Pertanyaan untuk Kandidat

No	Pertanyaan
1.	Bagaimana tingkat kemudahan menganalisis struktur dan logika kode sumber proyek
2.	Apakah proyek menyediakan dokumentasi yang cukup untuk membantu analisis kebutuhan dan desain
3.	Apakah terdapat kebijakan atau pedoman perubahan yang dapat membantu pengembang dalam mengelola perubahan
4.	Bagaimana proyek menjaga stabilitasnya selama periode waktu tertentu?
5.	Apakah terdapat kebijakan atau Pratik untuk mengurangi resiko ketidakstabilan dalam perubahan atau pembaharuan?
6.	Seberapa mudah menguji fungsionalitas proyek?
7.	Apakah proyek menyediakan kerangka pengujian atau skrip otomatis untuk memfasilitasi pengujian

8.	Bagaimana tingkat kematangan proyek dalam siklus pengembangan perangkat lunak?
9.	Apakah ada bukti adopsi praktik pengembangan yang matang dan teruji
10.	Apakah ada upaya khusus untuk mengukur dan meningkatkan efektivitas pengembangan

Tahap kedua, teknik pengumpulan data dilakukan dengan melakukan penyebaran kuisioner terhadap 6 responden. Setiap indikator yang dinilai oleh responden akan diklasifikasikan dalam 4 nilai luaran menggunakan skala likert. Setiap jawaban dihubungkan dalam bentuk pertanyaan yang di jabarkan pada table dibawah ini :

Tabel 4. Nilai Luaran

Luaran	Nilai
Sangat Setuju	4
Setuju	3
Kurang Setuju	2
Tidak Setuju	1

Untuk skala likert dapat dilihat pada table dibawah ini :

Tabel 5. Skala Likert

Skala Likert	Nilai
Sangat Baik	76% - 100%
Baik	51% - 75%
Kurang Baik	26% - 50%
Tidak Baik	0 - 25%

Menguraikan atribut tersebut pada model kualitas kami menjadi kriteria yang lebih terperinci dan kemudian turun ke metrik berarti profil tersebut sesuai dengan setiap kriteria yang didekomposisikan. Oleh karena itu, untuk mengkarakterisasi kualitas produk sebagai “Sangat Baik”, pemeliharaan, keandalan dan keamanan juga harus ditandai sebagai “Sangat Baik” [22]. Tabel 4 menunjukkan bahwa komponen perangkat lunak yang mendapat skor “Sangat Bagus” dalam pemeliharaan harus mendapat skor “Sangat Bagus” di sub atribut analyzability juga. hal tersebut, menerapkan pada komponen yang sudah dievaluasi harus menghasilkan nilai yang setara atau lebih tinggi. Tahap ketiga, hasil pengisian kuisioner akan dihitung menggunakan profile matching :

a) GAP

Table 6. GAP

No	Sub Indikator	Prioritas	Nilai Luaran	Indikator	Presentase
1.	<i>Analizability</i>	CF	3	<i>Maintainability</i>	40%
2.	<i>Changeability</i>	SF	3		
3.	<i>Stability</i>	CF	3		
4.	<i>Testability</i>	SF	4	<i>Reliability</i>	20%
5.	<i>Maturity</i>	SF	3		
6.	<i>Effectiveness</i>	CF	4	<i>Security</i>	10%
7.		CF	4		
8.		CF	4	<i>Mailing List Quality</i>	10%
9.		CF	4	<i>Documentation Quality</i>	10%
10.		CF	3	<i>Developer Base Quality</i>	10%

Setelah didapatkan tiap GAP dari masing masing sub indicator akan diberi bobot nilai dengan skala likert pada table dibawah ini :

Tabel 7. Nilai Selisih GAP

Selisih	Bobot	Keterangan
0	4	Nilai responden sesuai dengan kriteria
-1	3	Nilai responden kurang 1 poin
1	3.5	Nilai responden lebih 1 poin
-2	2	Nilai responden kurang 2 poin
2	2.5	Nilai responden lebih 2 poin
-3	1	Nilai responden kurang 4 poin

b) Pengelompokkan Core Factory dan Secondary Factory

Pada pengelompokkan Core Factory dan Secondary Factory memiliki nilai masing dari dua variable tersebut. Untuk nilai dari Core Factory dan Secondary Factory.

Tabel 8. Nilai FRUPS Core Factory

No	Kandidat	Indikator Faktor Nilai FURPS					
		Maintanability	Reliability	Security	Mailing List Quality	Documentation Quality	Developer Base Quality
1	Kandidat 1	3	3	3	3	4	3.5
2	Kandidat 2	3.25	3	4	3.5	3.5	3
3	Kandidat 3	3	3	3	4	3	3
4	Kandidat 4	3	3	3.5	3	3	3
5	Kandidat 5	3	3	3.5	3	3	3
6	Kandidat 6	3	3	3	4	3	3

Tabel 9. Nilai FRUPS Secondary Factory

No	Kandidat	Indikator Faktor Nilai FURPS					
		Maintanability	Reliability	Security	Mailing List Quality	Documentation Quality	Developer Base Quality
1	Kandidat 1	3.25	3	3	3	4	3.5
2	Kandidat 2	3	3	4	3.5	3.5	3
3	Kandidat 3	3.5	3	3	4	3	3
4	Kandidat 4	3	3	3.5	3	3	3
5	Kandidat 5	3	3	3.5	3	3	3
6	Kandidat 6	3	3	3	4	3	3

c) Perhitungan Nilai Total

Perhitungan nilai total ini di uji menggunakan rumus nilai total dengan presentase 60% dan 40% dari masing – masing kelompok. Berikut nilai total pada table di bawah ini :

Table 10. Nilai Total

No	Kandidat	Indikator nilai FRUPS					
		Maintanability	Reliability	Security	Mailing List Quality	Documentation Quality	Developer Base Quality
1	Kandidat 1	3.1	3	3	3	4	3.5
2	Kandidat 2	3.15	3	4	3.5	3.5	3
3	Kandidat 3	3.2	3	3	4	3	3
4	Kandidat 4	3	3	3.5	3	3	3
5	Kandidat 5	3	3	3.5	3	3	3
6	Kandidat 6	3	3	3	4	3	3

d) Perhitungan Nilai Total Responden

Adapun hasil perankingan responden yang menjadi tingkatan akhir dalam pengujian menggunakan metode Profile Matching. Berikut ini hasil perankingan responden pada table dibawah ini :

Tabel 11. Nilai Total Responden

No	Kandidat	Total Semua Nilai FRUPS
1	Kandidat 1	78%
2	Kandidat 2	81%
3	Kandidat 3	77%
4	Kandidat 4	74%
5	Kandidat 5	74%
6	Kandidat 6	76%

IV. KESIMPULAN

Berdasarkan pengujian tersebut dapat disimpulkan bahwa kualitas perangkat lunak aplikasi Arjuna Laundry memenuhi kriteria model SQO – OSS dengan “Sangat Baik”. Presentase keseluruhannya rata – rata 76% yang menempatkan pada kriteria “Sangat Baik”. Informasi presentase kandidat juga menunjukkan hasil yang memuaskan. Kandidat 1,2,3 dan 6 berada di atas rata – rata 76% dengan kriteria “Sangat Baik”, sedangkan kandidat 4 dan 5 berada di bawah rata – rata 76% dengan kriteria “Baik”. Dari hasil rata – rata memberikan dampak positif mengenai aplikasi Arjuna Laundry karena telah memenuhi atau melampaui ekspektasi. Oleh karena itu, aplikasi Arjuna Laundry dinilai efisiensi operasional dan cocok digunakan di lingkungan Jasa Laundry.

Pengembangan lebih lanjut kualitas aplikasi Arjuna Laundry berdasarkan model SQO – OSS, perlu adanya pembaruan dari segi keamanan data dan informasi pengguna untuk mengatasi potensi ancaman keamanan. Selanjutnya untuk memperkuat keakuratan aplikasi perlu adanya pengujian secara menyeluruh dan implementasi mekanisme pemulihan yang efektif. Dengan menerapkan saran ini, diharapkan aplikasi Arjuna Laundry dapat terus meningkatkan kualitas dalam model SQO – OSS, sehingga memberikan manfaat maksimal bagi pengguna dan pemilik bisnis.

DAFTAR PUSTAKA

- [1] N. W. Rahmat Hidayat, "Sistem Informasi Pelayanan Jasa Laundry Berbasis Aplikasi Android," *Journal Of Industrial Engineering Innovation* , 2023.
- [2] E. P. Wisuwati N., "Analisis Penerapan Manajemen Resiko Berdasarkan ISO 31000," *Junral Ilmiah Teknik Industri* , no. 5, 2009.
- [3] S. N. W. Yohanes De Deo Abhinaya Sigit Kumara, "Perancangan Sistem Informasi Berbasis Web D'Lofa Laundry Menggunakan Metode Waterfall," *IJSCR The Indonesian Journal Of Computer Science Research*, vol. 3 No. 1, pp. 10 - 17, 2024.
- [4] J. A. B. Mulyadi, "Aplikasi Sistem Pemesanan Jasa Laundry (E-Laundry) Berbasis Android," *ZO J.Sistem Informasi* , Vols. Vol. 1, No. 1, pp. 48-57, 2019.
- [5] N. Y. S. M. E. A. Elang Setia Ryananda, "Rancang Bangun Sistem Informasi E-Laundry Dengan Implementasi Berbasis Web (Programming)," *INNOVATIVE : Journal Of Social Science Research* , Vols. Vol. 2, No. 1, pp. 533 - 540, 2022.
- [6] G. S. B. D. S. W. R. V. R. W. S.M. Mulyatun, "Penerapan Search Engine Optimization Pada Website Kursus Online Anak Koding.ID," *Indonesian Journal Of Bussiness Intelligence (IJUBI)*, vol. Vol. 4 No. 2, 2021.
- [7] Y. A. d. T. L. M.A.Lubis, "Start Up Jasa Jemput Antar Laundry Berbasis Web," *Jurnal Teknologi Informasi* , Vols. Vol. 6, No. 1, pp. 117 - 129, 2019.
- [8] Y. T. L. S. R. B. S. T. F. Thanri, "Implementasi Aplikasi Knowledge Management Sistem Untuk Kinerja Washer Pada Wazy Laundry Medan," *Jurnal Teknologi Informasi* , vol. Vol. 2, pp. 228 - 240, 2023.
- [9] R. A. A. Wijayanti, "Perhitungan Estimasi Waktu Pada Produksi Barang Dengan Menerapkan Algoritma Naive Bayes Klasifikasi (Studi Kasus : PT. Hasil Raya Industries)," *JIKA* , vol. Vol. 5, pp. 109 - 118 , 2021.
- [10] W. A. R. H. S. Muhammad Andryan Wahyu Saputra, "Survei Teknik - Teknik Pengukuran Kualitas Perangkat Lunak," *ILKOMNIKA : Journal Of Computer Science And Applied Informatics* , Vols. Vol 3, nO. 1, pp. 11 - 29, 2021.
- [11] D. M. G. R. Jose P. Miguel, "A Review Of Software Quality Models For The Evaluation Of Software Products," *International Journal Of Software Engineering & Applications (IJSEA)*, vol. Vol. 5 No. 6, pp. 31 -52 , 2014.
- [12] C. M. & S. W. & G. E., "In Search For A Widely Applicable And Accepted Software Quality Model For Software Quality Engineering Paper," *Software Quality Engineering* , vol. Vol. 15, pp. 401 - 416, 2007.
- [13] A. -. B. A. Bassam, "Software Quality Evaluation: User's View," *2011*, vol. 5, no. 3, pp. 200 - 207, International Journal Of Applied Mathematics Informatics .
- [14] G. J. & P. E.M, "Software Quality Models : A Comparative Study," *International Journal Of Advanced Research In Computer Science And Electronics Engineering (IJARCSEE)*, vol. 2, no. 1, pp. 42 -51, 2013.
- [15] S. M. N. O. Adewumi Adewole, "A Review Of Models For Evaluating Quality In Open Source Software," *International Conference On Electronic Engineering and Computer Science* , no. IERI Procedia 4 , pp. 88 -92, 2013.
- [16] V. K. K. S. Gergious Gousios, "Software Quality Assesment Of Open Source Software 1.2," *ResearchGate (Panhellenic Conference In Informatics)*, vol. 11, pp. 304 - 315, 2016.
- [17] P. W. S. D. A. Z. Thomas Zimmermann, "Mining Version Histories To Guide Software Changes," *IEEE Transaction On Software Engineering*, vol. Vol. 31 No. 6, pp. 429 - 445, 2005.
- [18] A. J. A. Latif L, Buku Ajar Sistem Pendukung Keputusan Teori dan Implementasi, Yogyakarta: CV . Budi Utama, 2018.
- [19] R. D. S. Wijaya, "Sistem Pendukung Keputusan Pemilihan Siswa Berprestasi Pada Sekolah Menengah Pertama Dengan Metode AHP," *PROSISKO*, vol. 27, 2015.
- [20] I. D. Saputra, "Penerapan Metode Profile Matching Dalam Penentuan Jenis Tanaman," *Jurnal Sistem Dan Informatika* , pp. 46 -51, 2019.
- [21] D. M. A. S. Kusnadi, "Penerapan Metode Pofile Matching Untuk Penilaian Kenaikan Jabatan Karyawan (Studi Kasus : PT. Ilham Bangun Mandiri)," *Jurnal DIGIT*, vol. Vol. 5 No. 2, pp. 146 - 158, 2015.
- [22] G. G. V. K. P. L. Diomidis Spinellis, "Evaluating The Quality Of Open Source Software," *Electronics Notes In Theoritital Computer Sciences* , vol. 233, pp. 5 - 28, 2009.