

# PENGUKURAN KUALITAS PERANGKAT LUNAK PADA SIMRS KHANZA MENGGUNAKAN FUNCTION POINT DAN SYSTEM USABILITY SCALE

Akbar Satrio Nugroho<sup>1,\*</sup>, Fajri Rakhmat Umbara<sup>2</sup>, Rezki Yuniarti<sup>3</sup>

<sup>1</sup>Fakultas Sains dan Informatika, Informatika, Universitas Jenderal Achmad Yani, Cimahi, Indonesia  
Email: <sup>1,\*</sup>Akbarsat4@gmail.com

**Abstrak**— Pesaatnya pertumbuhan perangkat lunak pada masa ini membuat setiap sektor memiliki kebutuhannya masing-masing dalam menggunakan suatu produk perangkat lunak, tidak terkecuali di sektor layanan kesehatan yang menjadi salah satu sektor paling penting dimasa pandemi COVID-19 sebelumnya. Setelah melewati masa pandemi COVID-19 masyarakat menjadi sadar akan pentingnya sektor E-health dalam membantu akses layanan kesehatan. Sehingga hal ini perlu diukur terkait kesiapan untuk situasi mendatang dimana perangkat lunak SIMRS Khanza menjadi salah satu objek pengukuran pada penelitian kali ini. Proses pengukuran ini dapat menggunakan metode Function Point yang bertujuan untuk mengukur tingkat kompleksitas disisi fungsionalitas dan penggunaan metode System Usability Scale dalam mengukur kegunaan menurut pengguna disisi Usabilitas. Dengan adanya dukungan data hasil pengukuran ini maka, penentuan kualitas dalam suatu produk perangkat lunak dapat ditentukan dengan mudah. Dari permasalahan tersebut menemukan bahwa dalam penelitian ini dalam menghitung nilai Function Point perangkat lunak SIMRS Khanza memperoleh nilai sebesar 6011,28 dengan nilai Crude Function Point (CFP) sebesar 4968 dan Relative Complexity Adjustment Factor (RCAF) sebesar 1.21. Pada pengukuran skala nilai System Usability Scale mendapatkan nilai Skor sebesar 62,14 dimana sistem dapat dikategorikan yang berada pada tingkat Grade Good dari jumlah responden sebanyak 55 orang yang telah dilakukan uji reliabilitas dengan skor Cronbach Alpha sebesar 0,634. Berdasarkan dari hasil tersebut maka, dapat disimpulkan bahwa kualitas perangkat lunak SIMRS Khanza dari sisi fungsionalitas memiliki tingkat kompleksitas yang sangat tinggi namun dari hasil evaluasi pengukuran metode System Usability Scale dimana penggunaannya masih bisa menggunakannya secara efektif dan masih dapat merasa puas dalam menggunakannya.

**Kata Kunci:** Perangkat lunak, kualitas, Function Point, System Usability Scale, SIMRS, Usability.

**Abstract**— The rapid growth of software at this time makes each sector have its own needs in using a software product, including in the health service sector which is one of the most important sectors during the previous COVID-19 pandemic. After going through the COVID-19 pandemic, people became aware of the importance of the E-health sector in helping access to health services. So this needs to be measured regarding readiness for future situations where the Khanza SIMRS software is one of the objects of measurement in this study. This measurement process can use the Function Point method which aims to measure the level of complexity on the functionality side and the use of the System Usability Scale method in measuring usability according to users on the Usability side. With the support of this measurement data, the determination of quality in a software product can be easily determined. From this problem, it was found that in this study in calculating the Function Point value, the Khanza SIMRS software obtained a value of 6011.28 with a Crude Function Point (CFP) value of 4968 and a Relative Complexity Adjustment Factor (RCAF) of 1.21. In the measurement of the System Usability Scale value scale, the score was 62.14, where the system can be categorized as being at the Grade Good level from the number of respondents as many as 55 people who have been tested for reliability with a Cronbach Alpha score of 0.634. Based on these results, it can be concluded that the quality of the Khanza SIMRS software in terms of functionality has a very high level of complexity, but from the results of the System Usability Scale method measurement evaluation, users can still use it effectively and still feel satisfied in using it.

**Keywords:** Software, Quality, Function Point, System Usability Scale, SIMRS, Usability.

## 1. PENDAHULUAN

Kualitas perangkat lunak adalah suatu hal yang mendasar bagi keberhasilan suatu produk perangkat lunak. Tetapi kualitas itu sendiri yang sulit untuk didefinisikan secara jelas dan sulit untuk dipahami [1]. Produk yang dihasilkan dapat memunculkan berbagai macam pandangan mengenai masalah kualitas. Pada dasarnya, kualitas dapat digambarkan secara variatif oleh setiap individu.

Produk perangkat lunak layanan kesehatan (E-health) mengalami peningkatan selama pandemi COVID-19 berlangsung [2]. Pada saat kondisi pandemi yang sedang berlangsung, aksi yang disarankan adalah meminimalisir interaksi fisik antara pasien dan tenaga medis untuk menghambat penyebaran virus. Penggunaan layanan kesehatan E-health dapat menjadi solusi untuk menjadi alternatif yang optimal [3]. Dari banyaknya produk perangkat lunak yang dikembangkan selama pademi COVID-19 seringkali memiliki kualitas kode yang rendah (low code quality) [4]. Untuk itu perlu diketahui kualitas yang diberikan selama pengembangan pada masa itu. Selain itu, hasil dari pengembangan perangkat lunak ini secara garis besar memiliki bug yang dibagi kedalam beberapa kategori mulai dari algoritma, performa hingga keamanan perangkat lunak [5].

Definisi kualitas yang diperkenalkan oleh International Standards Organization (ISO) mendefinisikan kualitas sebagai “Totalitas fitur dan karakteristik dari suatu produk yang mampu memenuhi kebutuhan tertentu”

[1]. Urgensi dalam pengembangan perangkat lunak pada saat itu menyebabkan keraguan terhadap kualitas produk yang dihasilkan, mengingat kecepatan proses pengembangan yang diperlukan [2]. Keraguan yang dialami pengguna karena didorong oleh salah satu faktor seperti keharusan dalam menggunakan perangkat lunak dengan situasi yang baru. Akibatnya, setelah para pengguna merasa dipaksa oleh keadaan tersebut maka secara umum para pengguna sadar akan pentingnya meninjau ulang layanan kesehatan dan memprioritaskan layanan kesehatan di masa post-pandemic [6].

Pandemi COVID-19 telah mempercepat adopsi layanan E-health di sektor kesehatan [7]. Banyak rumah sakit di Indonesia kini telah menerapkan layanan E-health sebagai bagian penting yang tidak dapat diabaikan karena memiliki peluang yang sangat luar biasa [8]. Hal ini juga didukung oleh data yang diterbitkan oleh Kementerian Kesehatan (Kemenkes) RI pada tahun 2022 telah mencapai 88% rumah sakit di Indonesia yang telah menerapkan Sistem Informasi Manajemen Rumah Sakit (SIMRS), namun angka tersebut faktanya belum semua diimplementasikan secara optimal. Kebutuhan mendesak selama pandemi telah menunjukkan manfaat nyata dari sistem E-health, namun juga menyoroti tantangan dalam memastikan kualitas perangkat lunak yang cepat dikembangkan. Layanan kesehatan E-health ini meliputi berbagai aspek pelayanan administrasi seperti registrasi pasien, rekam medis, rawat jalan, rawat inap. Kemudian dalam aspek pelayanan obat, hasil test laboratorium dan fitur pembayaran seperti billing pasien. Selain itu, banyaknya model software quality yang telah diajukan oleh peneliti sebelumnya dapat menjadi dasar keberagaman dalam mengukur perangkat lunak E-health.

Di dalam software quality terdapat beberapa quality attribute yang perlu diukur dan diperiksa kualitasnya, salah satu pengukuran yang akan dilakukan pada penelitian kali ini adalah quality attribute Functionality dan Usability. Quality attribute ini disektor layanan kesehatan (E-health) menjadi salah satu faktor utama dalam suatu kualitas perangkat lunak dimana functionality dengan usability memiliki tingkat kepentingan yang tidak jauh berbeda [9]. Para pengguna lebih cenderung menggunakan suatu perangkat lunak yang berfungsi secara fungsional dan dapat digunakan secara sederhana dan mudah untuk digunakan. Semakin banyak fungsi yang ditawarkan kepada pengguna, maka pengguna perlu semakin terampil dalam memahami struktur sistem yang kompleks. Oleh karena itu, functionality yang berlebihan dapat mengurangi usability perangkat lunak. Penting untuk menyeimbangkan fungsionalitas dan usability dalam suatu perangkat lunak [10]. Dengan demikian kedua quality attribute ini perlu dan sangat penting diketahui tentang penerimaannya dari pengguna terhadap perangkat lunak itu sendiri.

Proses pengukuran suatu perangkat lunak dapat dilakukan dengan berbagai macam metode dan standar kualitas yang ada. Salah satunya metode yang akan digunakan pada penelitian ini adalah metode Function Point Analysis (FPA) yang diusulkan oleh Allan Albrecht di tahun 1979 dan System Usability Scale (SUS) yang dikembangkan oleh John Brooke pada tahun 1986. Didalam metode FPA yang diusulkan tersebut terdapat beberapa aspek yang digunakan sebagai parameter untuk mengukur perangkat lunak yaitu mulai dari aspek input, output dan file terkait dengan 14 faktor kompleksitas yang telah ditentukan [11][12]. Penggunaan metode FPA ini bertujuan untuk mencapai suatu ukuran kuantitatif dari sudut pandang pengguna (user) dan dari segi fungsionalitasnya.

Penelitian serupa yang telah dilakukan untuk mengukur suatu perangkat lunak menggunakan metode Function point telah banyak dilakukan, salah satunya adalah penelitian yang dilakukan oleh Khairani [13]. Pada penelitian yang dilakukannya Khairani melakukan pengukuran Function Point pada aplikasi Sistem Informasi Penerimaan Siswa Baru, dimana pada hasil pengukurannya aplikasi tersebut berhasil diukur dengan memperoleh nilai sebesar 108.12. Selain disektor pendidikan, pengukuran perangkat lunak telah dilakukan pada sektor pariwisata yang dilakukan oleh Parlita dkk [14]. Dari penelitian tersebut dilakukan pengukuran yang dilakukan pada aplikasi pemesanan tiket bus yang memperoleh nilai Function Point sebesar 148.50. Selain dari kedua penelitian tersebut dalam penelitian yang dilakukan oleh Lima Júnior, Muniz Farias dan Dias Belchior [15], mereka menambahkan satu tahapan yang dilakukan dalam menentukan ketidakpastian dalam metode function point dengan menggunakan bantuan Fuzzy Model sehingga dalam penelitiannya metode function point dikembangkan menjadi FFPA (Fuzzy Function Point Analysis).

Skala pengukuran evaluasi untuk mengetahui kegunaan (usability) dapat dilakukan dengan menggunakan metode System Usability Scale (SUS) yang bertujuan untuk menilai kegunaan produk dan mengumpulkan anggapan yang berbeda dari para penggunanya mengenai kegunaan perangkat lunak tersebut. Metode System Usability Scale ini telah banyak digunakan dan tidak memerlukan biaya yang tinggi [16]. Kemudahan lainnya juga dirasakan karena metode evaluasi ini telah diadaptasi dan tersedia dalam versi Bahasa Indonesia [17].

Terdapat standar internasional yang membahas tentang proses pengukuran kualitas perangkat lunak dengan menggunakan kedua metode tersebut, yaitu ISO 20926 dan ISO-9241. Standar tersebut membahas secara teknis bagaimana proses pengukuran perangkat lunak dijalankan. ISO-20926 secara langsung mengadopsi dokumen asli dari organisasi International Function Point User Group (IFPUG) Sehingga, standar tersebut dapat dijadikan sebuah acuan dalam mengukur suatu perangkat lunak. Dengan dasar keadaan tersebut maka, penelitian ini bertujuan untuk mengukur dan mengetahui kualitas perangkat lunak dalam konteks perangkat lunak di sektor pelayanan kesehatan (E-health). Hasil pengukuran tersebut akan dijadikan dasar untuk mengetahui nilai kualitas pada perangkat lunak dan kualitas layanan kesehatan elektronik disisi pelayanan dan informasi kesehatan.

## 2. METODE PENELITIAN

### 2.1 Function Point Analysis (FPA)

Function Point Analysis (FPA) adalah suatu metode yang berguna untuk mengukur suatu sistem atau aplikasi dari segi fungsionalitas dan kompleksitasnya. Dalam melakukan pengukuran perangkat lunak juga dapat memperkirakan upaya estimasi usaha (Software Effort Estimation) dan estimasi biaya (Software Cost Estimation) yang diperlukan untuk pengembangan perangkat lunak.

Selain untuk menghitung ukuran suatu aplikasi, FPA juga dapat digunakan sebagai pengembangan proyek dan meningkatkan proyek. Penggunaan function point analysis ini memiliki tiga tahapan prosedur yang harus dijalankan yaitu menentukan jenis perhitungan, mengidentifikasi ruang lingkup atau batasan aplikasi yang akan diukur dan melakukan proses perhitungan [7]. Terdapat tiga jenis perhitungan yang dapat dilakukan dalam mengukur suatu aplikasi menggunakan FPA. Tahapan pertama adalah menghitung Crude function Point (CFP), menghitung Relative Complexity Adjustment Factor (RCAF) dan yang terakhir dilakukan adalah menghitung nilai FP [7]. Pada penelitian yang akan dilakukan ini memiliki tahapan perhitungan yang secara umum masih sama dengan pengukuran pada metode function point biasa. Pada tahapan pertama yang dilakukan adalah menentukan jumlah komponen berdasarkan parameter utama untuk perhitungan Crude Function Points (CFP) dimana terdapat lima parameter utama yang dijelaskan pada Tabel 1.

**Tabel 1.** Parameter Utama Untuk Menentukan Bobot CFP

Parameter Utama	Deskripsi Parameter	Bobot
External Input (EI)	Proses dasar yang mengolah data dari luar sistem yang berasal dari pengguna atau sistem eksternal lainnya.	3 – 4 – 6
External Output (EO)	Sebuah proses yang mengirimkan data atau informasi yang diproses terlebih dahulu melalui process logic atau setidaknya melewati satu proses perhitungan matematik.	4 – 5 – 7
External Inquiry (EQ)	Sebuah proses yang mengirimkan data atau informasi untuk menyajikan informasi kepada pengguna (hanya menampilkan data).	3 – 4 – 6
Internal Logical File (ILF)	Peran ILF adalah untuk menyimpan data yang dikelola melalui setidaknya satu proses dasar dari aplikasi.	7 – 10 – 15
External Interface File (EIF)	Kumpulan data atau informasi yang dipelihara oleh aplikasi lain. Peran EIF adalah untuk menyimpan data yang dirujuk melalui aplikasi lain.	5 – 7 – 10

Untuk menentukan bobot dari setiap parameter utama yang ada pada Tabel 1 diatas dapat menggunakan tabel kompleksitas yang berdasarkan pada jumlah Record Element Type (RET) dan jumlah Data Element Type (DET). Dibawah ini tabel klasifikasi tingkat kompleksitas berdasarkan jumlah RET dan DET dapat dilihat pada Tabel 2 dan Tabel 3 sebagai berikut.

**Tabel 2.** Klasifikasi Tingkat Kompleksitas Parameter EI, EO dan EQ

		DETs		
		1-4	5-15	15+
RETS	1	Low	low	Average
	2-3	Low	Average	High
	3+	Average	High	High

**Tabel 3.** Klasifikasi Tingkat Kompleksitas Parameter ILF dan EIF

		DETs		
		1-19	20-50	51+
RETS	1	Low	low	Average
	2-5	Low	Average	High
	6+	Average	High	High

Selanjutnya, untuk menghitung CFP ini dapat dikalikan dengan faktor kali pada setiap parameter utama dengan bobot yang telah ditentukan pada Tabel 1. Pada Tabel 4 dibawah ini berfungsi sebagai proses yang digunakan untuk menghitung nilai CFP setelah mengetahui jumlah komponen utama yang ada didalam aplikasi.

**Tabel 4.** Matrik Untuk Menghitung CFP

Parameter Utama	Tingkat Kerumitan									Total
	Mudah			Sedang			Sulit			
	Count	Bobot	Point	Count	Bobot	Point	Count	Bobot	Point	
EI		3			4			6		
EO		4			5			7		
EQ		3			4			6		
ILF		7			10			15		
EIF		5			7			10		
<b>Total</b>										

Dimana :

1. Count = Jumlah dari komponen parameter utama yang ada pada aplikasi.
2. Bobot = Jumlah bobot berdasarkan dari tabel parameter utama berdasarkan bobot.
3. Point = Nilai count dikalikan dengan Bobot.
4. Total = Hasil dari penjumlahan setiap point.

Pada tahap kedua selanjutnya menghitung nilai RCAF dimana tahapan ini berguna untuk dapat menghitung tingkat kompleksitas berdasarkan dari 14 aspek yang dijelaskan dalam Tabel 5 dibawah ini dengan skala nilai 0-5.

**Tabel 5.** Total Degree Of Influence (TDI)

No.	Subject	Grade
1	Data Communication	
2	Distributed Data Processing	
3	Performance	
4	Heavily Used Configuration	
5	Transaction Rate	
6	Online Data Entry	
7	End-User Efficiency	
8	Online Update	
9	Complex Processing	
10	Reusability	
11	Installation Ease	
12	Operational Ease	
13	Multiple Sites	
14	Facilitate Change	

Dimana skala yang digunakan dalam menentukan ke-14 aspek diatas menggunakan skala 0-5 yang dijelaskan sebagai berikut :

0. Tidak berpengaruh
1. Insidental
2. Sedang
3. Rata-rata
4. Signifikan
5. Penting

Proses dari RCAF dapat dihitung menggunakan Persamaan (1) sebagai berikut :

$$RCAF = 0.65 + 0.01 \times \Sigma Fi \quad (1)$$

Dimana :

1. RCAF = Relative Complexity Adjustment Factor.
2. 0.65 + 0.01 = Konstanta Bawaan.
3.  $\Sigma Fi$  = Total nilai dari TDI.

Pada tahap terakhir yaitu dengan menghitung nilai Function Point dengan menggunakan Persamaan (2) sebagai berikut :

$$FP = CFP \times RCAF \quad (2)$$

Dimana :

1. FP = Function Point.
2. CFP = Crude Function Point.
3. RCAF = Relative Complexity Adjustment Factor.

Untuk mengetahui derajat kompleksitas berdasarkan dari hasil perhitungan sebelumnya, terdapat tabel yang dirumuskan oleh organisasi IFPUG untuk mengklasifikasikan tingkat dari hasil perhitungan function point yang dapat dilihat pada Tabel 6 dibawah ini.

**Tabel 6.** Total Degree Of Influence (TDI)

Function Point Result	Project Size	Complexity Level
1 - 50	Very Small	Low Complexity
51 - 100	Small	Average Complexity
101 - 300	Medium	High Complexity
300 - 700	Large	Very High Complexity
701+	Very Large	Extremely High Complexity

Dari klasifikasi yang ditunjukkan pada Tabel 6 diatas dimana ukuran dan skala proyek perangkat lunak berdasarkan jumlah function points yang dihitung dapat membantu dalam mengetahui ukuran proyek dan tingkat kompleksitas untuk perangkat lunak yang diukur.

## 2.2 System Usability Scale (SUS)

Pengukuran yang dilakukan menggunakan metode System Usability Scale (SUS) dapat melakukan pengukuran secara “quick and dirty” [16]. Penelitian sebelumnya yang telah dilakukan oleh Calisir, Bayraktaroglu dan Gumussoy [9] dalam penelitian yang telah dilakukan mereka melakukan perbandingan antara pentingnya kualitas atribut fungsionalitas dan usabilitas pada website E-health. Dalam penelitiannya tersebut menghasilkan temuan bahwa tingkat fungsionalitas lebih penting daripada usabilitas. Hal ini terjadi karena pengguna lebih berfokus kepada akses dan fungsi yang lengkap, namun keseimbangan antara kedua atribut kualitas tersebut sama

pentingnya untuk suatu perangkat lunak. Berdasarkan dari penelitian tersebut proses evaluasi pada kualitas atribut usability juga perlu dilakukan pada SIMRS Khanza. Penggunaan System Usability Scale digunakan karena hasil perhitungan yang diperoleh dapat mudah dipahami karena menggunakan skala 0 hingga 100 dimana nilai yang semakin besar menandakan semakin baik kualitas usability yang dimiliki [18]. Hal ini menjadikan metode SUS ini paling sering digunakan [19]. Umumnya dalam melakukan pengumpulan data kuisisioner dengan metode SUS ini menggunakan skala Likert [20]. Pada Tabel 7 dibawah ini menjelaskan setiap skor dari jawaban yang diberikan.

**Tabel 7.** Skor Jawaban Kuisisioner

Jawaban	Skor
Sangat tidak setuju	1
Tidak setuju	2
Netral	3
Setuju	4
Sangat setuju	5

Setelah mengetahui metode pengumpulan data yang akan dilakukan, responden perlu menjawab total 10 pernyataan yang telah disediakan dengan menjawab sesuai dengan penilaian masing-masing responden. Berikut ini adalah komponen pernyataan yang dapat dilihat pada Tabel 8 dibawah ini.

**Tabel 8.** Skor Jawaban Kuisisioner

No.	Komponen Pernyataan
1	Saya berpikir akan menggunakan sistem ini lagi.
2	Saya merasa sistem ini rumit untuk digunakan.
3	Saya merasa sistem ini mudah digunakan.
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8	Saya merasa sistem ini membingungkan.
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

Dari 10 pernyataan yang digunakan untuk melakukan perhitungan skor SUS ini, bentuk pernyataan yang digunakan memang telah ditentukan seperti yang tertera pada Tabel 8 diatas. Selain memiliki bentuk pernyataan yang telah ditentukan, bahasa yang digunakan juga telah diadaptasi dengan baik kedalam bahasa Indonesia yang dilakukan dalam penelitian Sharfina dan Santoso [14]. Dari Penelitian yang dilakukan tersebut menghasilkan bahwa versi yang telah diadaptasi kedalam bahasa Indonesia dapat diandalkan dengan skor Cronbach's alpha sebesar 0,841.

Proses perhitungan pada metode SUS ini memiliki aturannya sendiri yang mana proses perhitungan dilakukan dengan setiap poin item pernyataan ganjil (1,3,5,7,9) dikurangi 1, kemudian setiap point item pernyataan genap (2,4,6,8,10) dikurangi oleh 5 [20][21]. Untuk proses perhitungan skor SUS yang lebih jelasnya dapat dilihat pada Persamaan 3 dibawah ini.

$$SUS = (\sum(\text{Pernyataan Ganjil} - 1) + \sum(5 - \text{Pernyataan Genap})) \times 2.5 \quad (3)$$

Dimana :

1. Pernyataan Ganjil - 1 = Nilai yang diberikan oleh responden untuk pernyataan ganjil dalam kuesioner.
2. 5 - Pernyataan Genap = Nilai yang diberikan oleh responden untuk pernyataan genap dalam kuesioner.
3. 2.5 = Mengkonversi total skor penyesuaian dari skala 0 hingga 40 menjadi skala 0 hingga 100.

Hasil perhitungan yang telah dilakukan dapat diketahui tentang kisaran peringkat hasil penilaian pada sistem. Setelah mengetahui setiap skor dari pernyataan yang diberikan responden, selanjutnya adalah menghitung rata-

rata total skor SUS dengan menjumlahkan keseluruhan skor dan dibagi dengan jumlah responden. Untuk mengetahui peringkat skala nilai berdasarkan hasil perhitungan rata-rata sebelumnya dapat dilihat pada Tabel 2.9 dibawah ini.

**Tabel 9.** Skor SUS

Skor	Peringkat	Grade
Skor $\geq$ 86	Best Imaginable	A
Skor $\geq$ 72 dan $<$ 86	Excellent	B
Skor $\geq$ 52 dan $<$ 72	Good	C
Skor $\geq$ 38 dan $<$ 52	Ok/Fair	D
Skor $\geq$ 25 dan $<$ 38	Poor	E
$<$ 25	Worst Imaginable	F

Berdasarkan pada Tabel 9 diatas yang menggunakan skala Grade, hasil perhitungan yang telah dilakukan dapat dikategorikan kedalam 6 kategori, yaitu mulai dari peringkat paling bawah terdapat kategori yang bernilai  $<$ 25 dengan grade F, kemudian diikuti oleh grade E dengan nilai skor  $\geq$ 25 dan  $<$ 38, diikuti dengan grade D dengan skor  $\geq$ 38 dan  $<$ 52, diikuti dengan Grade C dengan skor  $\geq$ 52 dan  $<$ 72, diikuti dengan grade B dengan skor  $\geq$ 72 dan  $<$ 86 dan yang terakhir nilai paling tinggi dengan grade A dengan skor  $\geq$ 86 hingga ke angka 100.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Menghitung Function Point

Berdasarkan hasil dari analisis terhadap jumlah komponen perangkat lunak berdasarkan parameter utama pengukuran dari modul dan fungsi yang ada pada perangkat lunak SIMRS Khanza ini, untuk parameter EI berjumlah 106, EO berjumlah 15, EQ berjumlah 273, ILF berjumlah 253 dan EIF berjumlah 22. Sebaran jumlah data komponen dari setiap modul dapat berbeda, hal ini disebabkan karena terdapat beberapa modul yang hanya menampilkan data tanpa memelihara data. Tingkat kompleksitas dari setiap modul juga dapat dibedakan berdasarkan dari struktur data pada jumlah tabel dan kolom yang digunakan. Untuk menentukan kompleksitas parameter utama ini diambil berdasarkan jumlah RET dan DET yang telah dijelaskan sebelumnya pada Tabel 2 dan Tabel 3.

**Tabel 10.** Hasil Analisis Jumlah Parameter Utama Berdasarkan Tingkat Kompleksitasnya

	Low	Average	High	Total
<b>EI</b>	54	39	13	106
<b>EO</b>	2	0	13	15
<b>EQ</b>	120	46	107	273
<b>ILF</b>	51	42	160	253
<b>EIF</b>	22	0	0	22

Pada tahapan ini akan dilakukan proses perhitungan menggunakan metode Function Point. Dibawah ini akan dijelaskan proses perhitungan terhadap data yang ada pada Tabel 10 dengan melakukan beberapa tahap perhitungan seperti menghitung nilai CFP, menghitung nilai RCAF dan menghitung nilai akhir Function Point.

#### 3.1.1 Menghitung CFP

Pengukuran perangkat lunak menggunakan metode function point diawali dengan melakukan analisis dari jumlah komponen fungsional dari perangkat lunak itu sendiri. Analisis fungsionalitas sendiri dibuat berdasarkan fungsionalitas perangkat lunak yang ada. Tahapan ini adalah untuk menghitung jumlah dari setiap kategori fungsi, dimana terdiri tiga kategori yang digunakan yaitu rendah (*low*), rata-rata (*Average*) dan tingkat kompleksitas yang tinggi (*High*). Berikut merupakan hasil perhitungan CFP yang dapat dilihat pada Tabel 11.

Tabel 11. Hasil Perhitungan CFP

Parameter Utama	Tingkat Kerumitan									Total
	Low			Average			High			
	Count	Bobot	Point	Count	Bobot	Point	Count	Bobot	Point	
EI	54	3	162	39	4	156	13	6	78	396
EO	2	4	8	0	5	0	13	7	91	99
EQ	120	3	360	46	4	184	107	6	642	1186
ILF	51	7	357	42	10	420	160	15	2400	3177
EIF	22	5	110	0	7	0	0	10	0	110
<b>Total</b>										4968

Berdasarkan Tabel 11 diatas, hasil perhitungan CFP diatas, kompleksitas dari setiap parameter utama telah diklasifikasikan sesuai dengan tingkat kompleksitasnya. Hasil analisis dari tabel diatas menunjukkan hasil dari perhitungan semua komponen yang ada menghasilkan total sebesar 4968.

### 3.1.2 Menentukan Nilai RCAF

Untuk membantu menilai kompleksitas suatu fungsionalitas aplikasi yang akan dihitung, setiap karakteristik dari perangkat lunak ditentukan menggunakan Degree Of Influence. Hasil dari tingkat kompleksitas disajikan pada data Tabel 11 sebagai berikut:

Tabel 12. Hasil Pembobotan Nilai Degree Of Influence (DOI)

No	Karakteristik	Grade
1	Data Communication	4
2	Distributed Data Processing	3
3	Performance	5
4	Heavily Used Configuration	5
5	Transaction Rate	4
6	Online Data Entry	5
7	End-User Efficiency	5
8	Online Update	4
9	Complex Processing	4
10	Reusability	3
11	Installation Ease	3
12	Operational Ease	4
13	Multiple Sites	4
14	Facilitate Change	3
<b>Total</b>		56

Dari hasil analisis yang telah dilakukan maka dari Tabel 11 diatas menghasilkan total nilai RCAF sebesar 56 dari 14 karakteristik yang ditentukan. Jika RCAF tidak dihitung maka, nilai RCAF dapat secara langsung ditentukan sebesar 1.00. Hal ini tercantum dalam dokumen ISO 20926 dan dokumen IFPUG yang membahas terkait Value Adjustment Factor. Berdasarkan dari Tabel 11 sebelumnya maka, proses selanjutnya adalah menghitung nilai RCAF dengan menggunakan Persamaan 1 :

$$\begin{aligned}
 RCAF &= 0.65 + 0.01 \times \Sigma Fi \\
 &= 0.65 + (0.01 \times 56) \\
 &= 0.65 + 0.56 \\
 &= 0.65 + 0.56 \\
 &= 1.21
 \end{aligned}
 \tag{3}$$

Setelah proses menghitung nilai RCAF yang mendapatkan hasil sebesar 1.21, selanjutnya dapat dilakukan perhitungan untuk menentukan nilai akhir Function Point.

### 3.1.3 Menghitung Function Point

Tahapan terakhir untuk mendapatkan nilai suatu Function Point adalah dengan menghitung menggunakan Persamaan 2. Setelah mendapatkan nilai perhitungan CFP dan RCAF maka, kedua hasil perhitungan tersebut dapat dilakukan perhitungan sebagai berikut :

$$\begin{aligned}
 FP &= CFP \times RCAF \\
 &= 4968 \times 1.21 \\
 &= 6011,28
 \end{aligned}
 \tag{4}$$

Berdasarkan hasil perhitungan yang telah dilakukan, perolehan nilai function point untuk perangkat lunak SIMRS Khanza ini sebesar 6011,28. Berdasarkan dari Tabel 6 yang telah dijelaskan sebelumnya, bahwa perangkat lunak SIMRS Khanza ini memiliki tingkat kompleksitas yang sangat tinggi dan memiliki ukuran perangkat lunak yang sangat besar. Hasil perhitungan tersebut secara tidak langsung menunjukkan bahwa SIMRS Khanza mencakup berbagai modul dan fitur yang luas, mencakup berbagai aspek manajemen rumah sakit, mulai dari pendaftaran pasien, rekam medis elektronik, hingga manajemen farmasi dan keuangan.

Tingkat kompleksitas memiliki keterlibatan yang cukup erat kaitannya dengan kualitas perangkat lunak. Dari tingkat kompleksitas yang tinggi tidak berarti menandakan bahwa perangkat lunak ini memiliki kualitas yang buruk, namun sistem yang kompleks perlu dirancang dengan antarmuka pengguna yang intuitif untuk memastikan bahwa pengguna dapat mengoperasikannya dengan mudah dan efektif sehingga dapat mempengaruhi pengalaman pengguna.

### 3.2 Menghitung System Usability Scale

Pada tahapan dalam mengukur tingkat usability akan dilakukan proses perhitungan menggunakan metode System Usability Scale. Dibawah ini akan dijelaskan proses perhitungan terhadap data hasil kuisisioner yang telah berhasil dibagikan kepada 55 responden.

#### 3.2.1 Proses Perhitungan SUS Score

Setelah mengetahui hasil dari pengukuran kompleksitas yang telah dilakukan menggunakan function point sebelumnya maka, langkah selanjutnya adalah melakukan pengukuran yang dilakukan untuk mengetahui penerimaannya dari sisi pengguna. Dengan menggabungkan metode function point dan system usability scale, perangkat lunak dapat dievaluasi dengan baik mulai dari aspek fungsional maupun non-fungsional. Hal ini membantu dalam memastikan bahwa perangkat lunak tidak hanya berfungsi sesuai spesifikasi tetapi juga mudah digunakan dan diterima oleh pengguna. Berdasarkan hasil kuisisioner yang telah berhasil dibagikan dan diisi oleh sebanyak 55 responden, dimana kriteria dari responden yang berpartisipasi dalam mengisi kuisisioner ini secara garis besar memiliki profesi sebagai tenaga medis (dokter, perawat dan apoteker) dan staff administrasi. Data yang telah dihitung menggunakan rumus SUS ini dapat dilihat pada Tabel 13 dibawah ini.

**Tabel 13.** Hasil Perhitungan Skor SUS

Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Skor
R1	4	2	4	4	4	3	3	2	4	5	57,5
R2	5	1	5	1	5	1	3	2	4	3	85
R3	3	4	2	3	2	3	2	4	3	4	35
R4	5	5	4	5	4	4	4	4	4	4	47,5
R5	3	4	3	1	3	2	3	2	4	3	60
R6	5	5	5	1	5	2	5	1	5	3	82,5
R7	4	4	3	2	5	2	3	2	4	5	60

R8	4	1	4	2	2	1	2	2	5	3	70
R9	5	1	4	5	5	4	2	3	4	5	55
R10	2	3	3	2	4	4	2	2	4	3	52,5
R11	4	2	4	2	5	4	4	3	5	5	65
R12	4	5	3	2	4	3	3	3	5	5	52,5
R13	4	4	5	2	5	3	4	5	5	5	60
R14	4	4	3	5	5	4	2	3	3	5	40
R15	5	4	5	2	4	3	5	2	5	5	70
R16	5	2	4	5	4	3	4	3	4	5	57,5
R17	5	2	4	2	4	3	4	3	4	5	65
R18	4	1	2	1	5	3	2	1	2	5	60
R19	4	3	3	4	5	3	3	3	4	5	52,5
R20	5	5	4	2	5	4	4	3	4	3	62,5
R21	4	5	5	2	4	4	4	3	4	5	55
R22	4	2	4	2	5	3	4	3	4	5	65
R23	5	3	3	4	5	3	3	4	3	5	50
R24	5	5	4	2	5	3	4	3	5	5	62,5
R25	2	3	3	4	4	3	2	4	2	5	35
R26	4	5	3	4	4	3	2	3	4	5	42,5
R27	3	3	1	2	4	3	2	1	3	5	47,5
R28	5	1	1	2	5	1	1	1	2	2	67,5
R29	4	5	5	3	5	2	4	2	5	5	65
R30	5	2	4	3	5	2	4	2	5	5	72,5
R31	4	1	3	2	4	1	2	3	3	5	60
R32	1	3	3	2	4	1	2	1	4	5	55
R33	4	5	3	3	5	3	2	3	5	5	50
R34	5	5	4	2	5	2	4	5	5	5	60
R35	4	2	3	2	5	3	4	3	5	5	65
R36	4	5	4	5	5	4	3	3	4	5	45
R37	3	2	4	2	2	3	2	3	4	3	55
R38	2	1	4	2	4	2	2	3	4	5	57,5
R39	4	2	4	4	5	4	5	4	5	5	60
R40	5	2	5	2	5	2	4	2	5	5	77,5
R41	5	2	5	3	5	2	3	3	5	5	70
R42	4	3	4	2	4	3	3	2	5	5	62,5
R43	4	2	4	2	4	3	3	2	4	4	65
R44	3	2	4	1	4	3	2	1	4	3	67,5
R45	4	2	4	2	4	2	3	2	4	5	65
R46	3	2	4	2	4	2	4	2	4	4	67,5
R47	5	1	5	5	5	1	5	1	4	5	77,5
R48	1	5	5	1	5	1	5	1	4	3	72,5
R49	5	2	4	3	3	3	2	2	3	5	55
R50	5	1	5	1	5	1	5	1	5	5	90
R51	5	1	5	1	5	1	5	5	5	3	85
R52	5	2	4	4	5	1	3	1	5	4	75
R53	4	1	4	2	5	2	4	2	4	3	77,5
R54	5	1	5	3	5	1	5	1	5	5	85
R55	4	2	4	4	4	2	4	2	4	4	65
<b>Rata-rata Skor SUS</b>											<b>62,14</b>

Pada data Tabel 13 diatas, proses perhitungan dilakukan menggunakan metode SUS yang menghasilkan hasil skor SUS sebesar 62,14. Maka hasil dari skor tersebut berdasarkan Tabel 9 sebelumnya dapat dikategorikan termasuk

skala Grade C yang menunjukkan bahwa perangkat lunak SIMRS Khanza memiliki tingkat Usability dalam kategori Good. Dengan demikian, pengguna masih dapat dengan mudah menggunakan perangkat lunak ini. Dari sebaran data diatas bahwa pengalaman pengguna pada perangkat lunak SIMRS Khanza ini memiliki pengalaman yang sangat variatif. Terlepas dari hasil keseluruhan yang diperoleh, analisa yang dilakukan pada skor beberapa individu memberikan skor yang tinggi dimana beberapa responden memiliki pengalaman yang sangat positif ketika menggunakan perangkat lunak ini.

**3.2.2 Uji Validitas**

Uji Validitas pada SUS adalah suatu proses yang bertujuan untuk mengevaluasi dari setiap instrumen pernyataan yang digunakan dalam menentukan sejauh mana skor setiap item secara keseluruhan. Pengujian ini dilakukan menggunakan bantuan perangkat lunak SPSS dengan jumlah responden 55 orang sehingga nilai r-tabel dapat ditentukan sebesar 0,2656.

**Tabel 14.** Hasil Nilai Uji Validitas

Item Pernyataan	r-Tabel	<i>r-Pearson</i>	N	N Sig.
Q1	0,265	0,493	55	0,001
Q2	0,265	0,474	55	0,001
Q3	0,265	0,487	55	0,001
Q4	0,265	0,449	55	0,001
Q5	0,265	0,442	55	0,001
Q6	0,265	0,432	55	0,001
Q7	0,265	0,587	55	0,001
Q8	0,265	0,524	55	0,001
Q9	0,265	0,522	55	0,001
Q10	0,265	0,494	55	0,001

Untuk dapat dikatakan valid dalam pengujian validitas ini *r-Pearson* perlu memiliki nilai lebih besar dari r-Tabel pada tingkat signifikansi yang pada pengujian ini ditentukan menggunakan tingkat signifikansi sebesar 0,05 maka, item tersebut dapat dianggap valid. Hasil dari uji validitas dapat dilihat pada Tabel 3.6 diatas yang menunjukkan bahwa semua pernyataan kuisioner SUS ini memiliki nilai lebih besar dari nilai r-Tabel (0,265) yang berarti indikator penelitian terhadap variabel usability dinyatakan valid karena nilai *r-Pearson* lebih besar daripada nilai r-Tabel (*r-Pearson* > r-Tabel).

**3.2.3 Uji Reliabilitas**

Uji reliabilitas dilakukan bertujuan untuk dapat mengetahui konsistensi dari setiap item pernyataan yang digunakan agar mengetahui jika hasil yang diberikan konsisten dari waktu ke waktu. Proses uji reliabilitas ini dilakukan menggunakan bantuan perangkat lunak SPSS dengan responden sebanyak 55 orang. Uji reliabilitas dapat dikatakan valid apabila nilai Cronbach’s Alpha bernilai lebih dari 0,6.

**Reliability Statistics**

Cronbach's Alpha	N of Items
.634	10

**Gambar 1.** Hasil Nilai Uji Reliabilitas

Hasil yang ditunjukkan pada uji reliabilitas akan semakin baik apabila semakin mendekati nilai angka 1. Berdasarkan dari hasil uji reliabilitas yang ditunjukkan pada Gambar 1 diatas menunjukkan bahwa hasil dari uji reliabilitas menghasilkan jumlah Cronbach's Alpha sebesar 0,634, sehingga dapat disimpulkan bahwa data tersebut reliabel.

#### 4. KESIMPULAN

Dari hasil penelitian ini, dapat disimpulkan bahwa metode FPA dan SUS cukup efektif digunakan untuk melakukan pengukuran perangkat lunak di sektor layanan kesehatan. Dari hasil pengukuran pada metode FPA yang telah dilakukan, maka diperoleh nilai CFP sebesar 4968, nilai RCAF sebesar 1.21 dan nilai akhir Function Point sebesar 6011,28. Hasil yang diperoleh dalam proses perhitungan ini menunjukkan bahwa SIMRS Khanza termasuk kedalam kategori perangkat lunak yang memiliki tingkat kompleksitas yang sangat tinggi dan ukuran yang sangat besar. Pada hasil pengujian usability menggunakan metode System Usability Scale dengan data responden sebanyak 55 responden menunjukkan bahwa tingkat usability perangkat lunak SIMRS Khanza ini berada di tingkat Good dengan nilai skor yang diperoleh sebesar 62,14. Berdasarkan dari nilai function point yang diperoleh secara tidak langsung mengindikasikan bahwa metode FPA sangat efektif dalam mengidentifikasi, mengukur tingkat kompleksitas dan mengetahui ukuran proyek. Selain itu dari hasil uji usability menunjukkan bahwa perspektif pengguna tentang perangkat lunak SIMRS Khanza masih dapat memberikan pengalaman pengguna yang memadai. Berdasarkan hasil dari pengukuran yang telah dilakukan ini masih banyak sekali quality attribute yang dapat diukur guna memberikan pengetahuan yang menyeluruh terkait kualitas dan quality attribute yang masih dapat dikembangkan lebih jauh. Dengan demikian penulis memberikan saran untuk dapat dilakukan penelitian di masa yang akan datang, diantaranya adalah melakukan perbaikan disisi internal (Source Code) dengan melakukan refactoring pada perangkat lunak SIMRS Khanza ini. Proses refactoring ini dapat dilakukan dengan mengidentifikasi dan memperbaiki kode yang dapat mengganggu fungsionalitas, keterbacaan, dan skalabilitas sistem. Salah satu pendekatan yang dapat digunakan adalah dengan menerapkan prinsip-prinsip clean code.

#### UCAPAN TERIMAKASIH

Dengan penuh rasa syukur kepada Tuhan Yang Maha Esa, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan bimbingan selama proses penyusunan tugas akhir ini. Ucapan terima kasih khusus ditujukan kepada orang tua atas dukungan dan doanya, serta kepada para dosen pembimbing dan penguji yang telah memberikan arahan, saran, dan motivasi yang sangat berarti. Penulis juga berterima kasih kepada seluruh staf pengajar dan karyawan di program studi, serta kepada teman-teman dan pasangan yang telah memberikan semangat. Semoga segala kebaikan yang telah diberikan mendapatkan balasan yang berlimpah dari Tuhan Yang Maha Esa.

#### REFERENCES

- [1] H. Al-Kilidar, K. Cox, and B. Kitchenham, "The use and usefulness of the ISO/IEC 9126 quality standard," *2005 Int. Symp. Empir. Softw. Eng. ISESE 2005*, pp. 126–132, 2005, doi: 10.1109/ISESE.2005.1541821.
- [2] W. Alhakami, A. Binmahfoudh, A. Baz, H. Alhakami, M. T. J. Ansari, and R. A. Khan, "Atrocious Impinging of COVID-19 Pandemic on Software Development Industries," *Comput. Syst. Sci. Eng.*, vol. 36, no. 2, pp. 323–338, 2021, doi: 10.32604/CSSE.2021.014929.
- [3] T. H. Tebeje and J. Klein, "Applications of e-Health to Support Person-Centered Health Care at the Time of COVID-19 Pandemic," *Telemed. e-Health*, vol. 27, no. 2, pp. 150–158, 2021, doi: 10.1089/tmj.2020.0201.
- [4] P. A. Da Mota Silveira Neto *et al.*, "A Deep Dive into the Impact of COVID-19 on Software Development," *IEEE Trans. Softw. Eng.*, vol. 48, no. 9, pp. 3342–3360, 2022, doi: 10.1109/TSE.2021.3088759.
- [5] A. Rahman and E. Farhana, "An Exploratory Characterization of Bugs in COVID-19 Software Projects," 2020, [Online]. Available: <http://arxiv.org/abs/2006.00586>
- [6] Z. Gamsızkan, A. Kaya, and M. A. Sungur, "The increasing importance of the e-health system after the covid-19 outbreak with new healthcare expectations," *Eurasian J. Fam. Med.*, vol. 10, no. 2, pp. 84–91, 2021, doi: 10.33880/ejfm.2021100206.
- [7] N. Sutarsa, P. A. S. Astuti, M. Choy, and M. Moore, "COVID-19 Pandemic: Opportunity to Accelerate e-Health in Indonesia," *Public Heal. Prev. Med. Arch.*, vol. 8, no. 1, pp. 1–3, 2020, doi: 10.15562/phma.v8i1.259.

- [8] F. N. Fuadina, N. W. K. Projo, and S. Mariyah, "What We Know from Telemedicine Data in Indonesia? Study case using Alodokter, Dokter.id, and Honestdocs," *Proc. Int. Conf. Data Sci. Off. Stat.*, vol. 2021, no. 1, pp. 131–144, 2022, doi: 10.34123/icdsos.v2021i1.53.
- [9] F. Calisir, A. E. Bayraktaroglu, C. A. Gumussoy, and H. Topaloglu, "The Relative Importance of Usability and Functionality Factors for E-Health Web Sites," *Adv. Hum. Factors Ergon. Healthc.*, vol. 00, no. 1998, pp. 714–723, 2010, doi: 10.1201/EBK1439834978-81.
- [10] A. E. Bayraktaroglu, F. Calisir, and C. A. Gumussoy, "Usability and functionality: A comparison of project managers' and potential users' evaluations," *IEEM 2009 - IEEE Int. Conf. Ind. Eng. Eng. Manag.*, pp. 2019–2023, 2009, doi: 10.1109/IEEM.2009.5373206.
- [11] R. S. Dewi, Y. S. Dharmawan, and S. N. Aisah, "Measuring Software Size and Effort Estimation on Islamic Banking Application," *2020 3rd Int. Conf. Inf. Commun. Technol. ICOIACT 2020*, pp. 229–233, 2020, doi: 10.1109/ICOIACT50329.2020.9332073.
- [12] A. Abran and P. N. Robillard, "Function points: A study of their measurement processes and scale transformations," *J. Syst. Softw.*, vol. 25, no. 2, pp. 171–184, 1994, doi: 10.1016/0164-1212(94)90004-3.
- [13] D. Khairani, "Studi Kasus Pengukuran Sistem Informasi Menggunakan Function Point (Fp)," *J. Tek. Inform.*, vol. 8, no. 2, pp. 1–7, 2015, doi: 10.15408/jti.v8i2.2442.
- [14] R. Parlika, A. M. Ali, A. D. Maharani, B. A. Julastri, and S. M. Pruiata, "Panduan Pengukuran Perangkat Lunak Metode Function Point Serta Implementasinya pada Website Pemesanan Tiket Bus," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 5, no. 1, p. 1, 2023, doi: 10.36499/jinrpl.v5i1.6633.
- [15] O. de S. Lima Júnior, P. P. Muniz Farias, and A. Dias Belchior, "A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments," *CLEI Electron. J.*, vol. 5, no. 2, pp. 1–14, 2018, doi: 10.19153/cleiej.5.2.4.
- [16] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Int. J. Hum. Comput. Interact.*, vol. 24, no. 6, pp. 574–594, 2008, doi: 10.1080/10447310802205776.
- [17] Z. Sharfina and H. B. Santoso, "An Indonesian adaptation of the System Usability Scale (SUS)," *2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACIS 2016*, pp. 145–148, 2017, doi: 10.1109/ICACIS.2016.7872776.
- [18] F. G. Sembodo, G. F. Fitriana, and N. A. Prasetyo, "Evaluasi Usability Website Shopee Menggunakan System Usability Scale (SUS)," *J. Appl. Informatics Comput.*, vol. 5, no. 2, pp. 146–150, 2021, doi: 10.30871/jaic.v5i2.3293.
- [19] J. R. Lewis, "The System Usability Scale: Past, Present, and Future," *Int. J. Hum. Comput. Interact.*, vol. 34, no. 7, pp. 577–590, 2018, doi: 10.1080/10447318.2018.1455307.
- [20] I. Rachmawati and R. Setyadi, "Evaluasi Usability Pada Sistem Website Absensi Menggunakan Metode SUS," *J. Inf. Syst. Res.*, vol. 4, no. 2, pp. 551–561, 2023, doi: 10.47065/josh.v4i2.2868.
- [21] J. W. Susekti, G. F. Soebagio, and S. R. Wicaksono, "Analisis Pengukuran dan Kualitas Software Menggunakan Function Point Analysis (Studi Kasus Software Harmony)," *J I M P - J. Inform. Merdeka Pasuruan*, vol. 7, no. 3, p. 129, 2023, doi: 10.51213/jimp.v7i3.781.