

Analisis Pembiayaan Sistem *Digital Signage* dengan Metode *Function Point Analysis* pada Industri F&B

Cost Analysis of Digital Signage System Using the Function Point Analysis Method in the F&B Industry

DEDE RUKMANA^{1*}, HUSTINAWATI¹

Abstrak

Digital Signage merupakan teknologi strategis dalam industri makanan dan minuman (F&B) yang memungkinkan penyampaian konten dinamis, seperti menu, promosi, dan iklan secara *real-time*, sehingga meningkatkan pengalaman pelanggan dan efisiensi operasional. Sistem *Digital Signage* pada penelitian ini dirancang menggunakan arsitektur *microservices* berbasis *Java Spring Boot* dan *React JS*, serta terintegrasi dengan platform *MagicInfo* untuk mendukung manajemen konten secara efisien. Fitur *Sync Play* memungkinkan penayangan konten secara sinkron di berbagai layar. Penelitian ini bertujuan untuk mengukur fungsionalitas dan kompleksitas sistem menggunakan metode *Function Point Analysis* (FPA), guna menghasilkan estimasi yang akurat terkait upaya, biaya, dan waktu pengembangan. Hasil perhitungan menunjukkan selisih antara estimasi dan ukuran aktual sebesar 1,68%, yang mencerminkan tingkat akurasi tinggi dari metode ini. Penelitian ini diharapkan dapat mendukung transformasi digital di industri F&B melalui implementasi solusi *Digital Signage* yang inovatif dan efektif.

Kata Kunci: Analisis *function point*, *digital signage*, estimasi sistem, industri F&B, *MagicInfo*

Abstract

Digital signage has emerged as a strategic technology in the food and beverage (F&B) industry, delivering dynamic content—such as menus, promotions, and advertisements—in real-time to enhance customer experience and operational efficiency. This study presents a digital signage system designed using a *microservices* architecture based on *Java Spring Boot* and *React JS*, integrated with the *MagicInfo* platform to facilitate efficient content management. The *Sync Play* feature enables synchronized content display across multiple screens. The functionality and complexity of the system are evaluated using the *Function Point Analysis* (FPA) method to generate accurate estimates of development effort, cost, and time. The results indicate a deviation of only 1.68% between the estimated and actual system size, demonstrating the high accuracy of the method. These findings are expected to support digital transformation in the F&B industry through the deployment of innovative and effective *Digital Signage* solutions.

Keywords: *digital signage*, food and beverage industry, *function point analysis*, *MagicInfo*, system estimation

PENDAHULUAN

Dalam era digital yang terus berkembang, teknologi *digital signage* semakin banyak digunakan di berbagai sektor bisnis. Khususnya dalam industri makanan dan minuman (F&B), penerapan *digital signage* berperan penting dalam meningkatkan pengalaman pelanggan serta efisiensi operasional. Teknologi ini secara luas menggantikan media konvensional dengan memanfaatkan beragam aplikasi dan inovasi teknologi. Selain itu, teknologi seperti RFID memungkinkan penggunaan sensor untuk membaca data secara otomatis, yang dapat mendukung berbagai kebutuhan operasional (Fachrurrazi dan Hizli 2021). Dalam penelitian sejenis mengenai

¹Departemen Teknologi dan Rekayasa, Fakultas Magister Manajemen Sistem Informasi, Universitas Gunadarma

*Penulis Korespondensi: Surel: insan.pembelajar8911@gmail.com.

digital signage yang dilakukan oleh Dentsu Agency, ditemukan bahwa keberhasilan sebuah iklan dalam menarik perhatian dan membangun kesadaran audiens dipengaruhi oleh kemampuan konten iklan dalam menonjolkan elemen audio dan visual dari media yang digunakan (Ningrum, 2022).

Pada penelitian sebelumnya yang dilakukan oleh Hapsari dan Husen (2015), hasil pengujian menggunakan metrik berbasis fungsi (*function-oriented metrics*) untuk estimasi kualitas perangkat lunak dengan metode *function point* menunjukkan tingkat akurasi sebesar 75% yang dinilai cukup baik. Dalam pengukuran perangkat lunak, metrik kuantitatif terbagi menjadi dua, yaitu *size-oriented* dan *function-oriented*. Metrik *size-oriented*, seperti *Line of Code (LOC)*, memiliki kelemahan karena sangat bergantung pada bahasa pemrograman dan gaya pengkodean, sehingga tidak menjamin akurasi dalam menggambarkan kompleksitas atau upaya pengembangan. Sebaliknya, *function-oriented metrics*, seperti *Function Point Analysis (FPA)*, menawarkan pendekatan yang lebih objektif dan independen dari teknologi implementasi karena mengukur berdasarkan fungsi yang disediakan kepada pengguna. *Function Point Analysis* adalah teknik terstruktur untuk mengklasifikasikan komponen-komponen dari sebuah sistem (Longstreet, 2005). Oleh karena itu, *function point* dianggap lebih unggul dalam memperkirakan ukuran dan usaha pengembangan perangkat lunak secara konsisten dan akurat.

Dalam konteks kampus, *digital signage* belum banyak dimanfaatkan secara optimal, meskipun telah ada upaya untuk menggunakan media seperti *website* yang menampilkan informasi berbentuk teks berjalan. Namun, kombinasi media yang lebih kaya seperti video, foto, pengumuman, dan teks berjalan dalam satu layar *digital signage* terbukti lebih menarik dan interaktif dibandingkan metode konvensional, serta dapat menggantikan penggunaan papan pengumuman fisik (Muhajirin dan Daryono 2018).

Penelitian terkait *digital signage* juga dilakukan oleh Roodasri dan Pourmohammadi (2021) untuk implementasi di industri *Food and Beverage (F&B)*. Pada penelitian ini, penerapan digital signage dilakukan untuk menampilkan menu dan informasi pada kafe Robert's Coffee Redi pada smart TV. Sistem ini dapat diatur secara jarak jauh dari antarmuka *website*.

Pada penelitian lainnya yang dilakukan oleh Taufiqurrahman (2022), analisis *function point* diterapkan dalam pengukuran perangkat lunak pada Aplikasi Akademik Siswa Berbasis *Website (AASBW)*. Sebagai upaya meminimalkan unsur subjektivitas dalam analisis studi kasus ini, penulis mengadopsi metode perhitungan *function point* yang mengacu pada pedoman resmi dari *International Function Point Users Group (IFPUG, 2024)*. Berdasarkan hasil perhitungan, diperoleh nilai *function point* sebesar 152,90. Nilai ini kemudian dikonversikan untuk menghasilkan estimasi dalam beberapa parameter pengukuran, seperti estimasi waktu pengembangan, kebutuhan biaya, serta biaya manajerial yang diperlukan dalam pengembangan dan pengelolaan perangkat lunak tersebut.

Berdasarkan penelitian sebelumnya, *digital signage* telah digunakan secara luas di berbagai sektor untuk menyampaikan informasi secara efektif dan efisien, termasuk di industri F&B dan lingkungan kampus. Namun, penelitian yang ada lebih berfokus pada implementasi teknologinya tanpa adanya pengukuran mendalam terkait ukuran fungsional sistem tersebut. Hingga saat ini, belum ada penelitian yang secara spesifik mengukur sistem *digital signage* menggunakan pendekatan kuantitatif seperti *function point analysis*. Oleh karena itu, penelitian ini dilakukan untuk mengisi kesenjangan tersebut dengan menggunakan metode *function point* guna menghitung ukuran fungsional sistem. Pendekatan ini memungkinkan dalam memperkirakan biaya, upaya, dan waktu pengembangan secara lebih akurat, serta memberikan dasar yang lebih terukur untuk pengembangan dan implementasi sistem *digital signage* di masa mendatang.

METODE PENELITIAN

Function point analysis adalah teknik terstruktur untuk mengklasifikasikan komponen sistem dengan memecahnya menjadi bagian-bagian kecil agar lebih mudah dipahami dan dianalisis. Tujuan dari pengukuran menggunakan *function point* adalah untuk mendapatkan ukuran biaya, durasi, dan jumlah sumber daya yang diperlukan oleh sebuah proyek perangkat lunak dan dapat dilakukan pada setiap tahap pengembangan perangkat lunak (Rambu *et al.* 2019).

Menurut Hapsari dan Husen (2015), parameter yang digunakan dalam pengukuran *function point* meliputi:

- External Input* (EI) merupakan masukan eksternal yang berasal dari pengguna atau aplikasi lain. EI digunakan untuk menyediakan data spesifik aplikasi atau mengontrol informasi dan untuk memperbarui *Internal Logical File* (ILF).
- External Output* (EO) merupakan keluaran eksternal yang menyajikan informasi kepada pengguna berdasarkan data aplikasi, seperti laporan, tampilan layar, atau pesan kesalahan.
- External Inquiry* (EQ) merupakan kombinasi masukan dan keluaran yang menghasilkan *output* sederhana sebagai respon atas permintaan data, misalnya hasil pencarian data.
- Internal Logical File* (ILF) merupakan data atau informasi kontrol yang sepenuhnya dikelola oleh aplikasi. File ini dapat berbentuk file sederhana atau tabel tunggal dalam *database* relasional.
- External Interface Files* (EIF) merupakan sekelompok data atau informasi kontrol yang berasal dari aplikasi lain tetapi diperlukan untuk operasi aplikasi tersebut.

a. Menentukan *Function Point Complexity*

Setiap *user function* dalam perangkat lunak perlu dihitung tingkat kompleksitasnya. Menurut Hapsari dan Husen (2015) dan Rachmat and Saparudin (2017), pengelompokan *user function* dilakukan berdasarkan tingkat kompleksitas yang terbagi menjadi tiga kategori: *high*, *medium*, dan *low*.

Tabel 1 Kompleksitas *function point*

Tipe <i>user function</i>	<i>complexity weight</i>		
	<i>Low</i>	<i>Medium</i>	<i>High</i>
<i>External Input</i> (EI)	3	4	6
<i>External Output</i> (EO)	4	5	7
<i>Internal Logical File</i> (ILF)	7	10	15
<i>External Interface File</i> (EIF)	5	7	10
<i>External Inquiry</i> (EQ)	3	4	6

Dilihat dari Tabel 1, nilai dari setiap *user function* pada tiap kategori kompleksitas akan dikalikan dengan *complexity weight* sebagaimana tertera pada tabel tersebut.

b. Menghitung Total Nilai *Unadjusted Function Point* (UFP)

Setelah mengetahui tingkat kompleksitas *user function*, maka selanjutnya kita bisa menghitung nilai total dari *Unadjusted Function Point* (UFP) dengan formula sebagai terdapat pada Persamaan 1.

$$UFP = \sum_{i=1}^5 \sum_{j=1}^3 x_{ij} xw_{ij} \quad (1)$$

Unadjusted Function Point (UFP) dihitung dengan menjumlahkan bobot dari seluruh *Unadjusted Function* (UF). Perhitungan ini mencakup jumlah UF untuk setiap jenis *i* dengan tingkat kompleksitas *j* serta bobot yang sesuai. Dengan kata lain, UFP diperoleh dengan mengalikan bobot dengan jumlah fitur pada setiap UF yang memiliki karakteristik berbeda. Hasil

dari perhitungan lima fungsi tersebut kemudian dijumlahkan untuk memperoleh nilai Total Unadjusted *Function Point* (TUF_P) (Rachmat dan Saparuddin 2017).

c. Menghitung *Technical Complexity Adjustment* (TCA)

Ada 14 faktor yang mempengaruhi tingkat kompleksitas dalam implementasi sistem operasional. Setiap faktor dinilai dengan skala 0 hingga 5, di mana nilai 0 menunjukkan bahwa faktor tersebut tidak memiliki dampak, sedangkan nilai 5 menunjukkan bahwa faktor tersebut sangat mempengaruhi perangkat lunak yang dievaluasi. Dalam *Function Point Analysis* (FPA), faktor-faktor ini dikenal sebagai *value adjustment factors* yang dideskripsikan pada Tabel 2.

Tabel 2 *Value adjustment factors*

ID	Factor	IDE	Factor
C1	<i>Data communications</i>	C8	<i>On-line update</i>
C2	<i>Distributed function</i>	C9	<i>Complex processing</i>
C3	<i>Performance objectives</i>	C10	<i>Reusability</i>
C4	<i>Heavily used configuration</i>	C11	<i>Installation ease</i>
C5	<i>Transaction rate</i>	C12	<i>Operational ease</i>
C6	<i>On-line data entry</i>	C13	<i>Multiple sites</i>
C7	<i>End-user efficiency</i>	C14	<i>Facilitate change</i>

Jika seluruh faktor dan skor sudah ditentukan, maka selanjutnya adalah menghitung nilai Total *Degree of Influence* (TDI) dengan menjumlahkan ke 14 faktor tersebut. Jika sudah dapat nilai TDI, maka selanjutnya kita bisa menghitung nilai TCA dengan formula pada Persamaan 2.

$$TCA = 0.65 + 0.01 \times TDI \quad (2)$$

d. Menghitung *Function Point*

Cara menghitung nilai *function point* dari sebuah perangkat lunak adalah dengan menggunakan formula pada Persamaan 3.

$$FP = UFP \times TCA \quad (3)$$

Function Point (FP) dapat digunakan untuk memperkirakan jumlah *Lines of Code* (LOC) perangkat lunak, biaya pengembangan, upaya yang dibutuhkan, alokasi sumber daya, serta durasi pengerjaan (Rachmat dan Saparuddin 2017). Estimasi ukuran perangkat lunak dalam satuan LOC diperoleh dengan mengalikan nilai FP dengan *productivity factor* sesuai bahasa pemrograman yang digunakan. *Productivity factor* menunjukkan jumlah baris kode logis per *function point*, yang nilainya berbeda-beda tergantung pada bahasa pemrograman, sebagaimana dapat dilihat pada Tabel 3.

Tabel 3 *Productivity factor version 5.0*

Programming Language	Productivity Factor			
	Avg	Low	Median	High
Jawa	53	53	14	134
JavaScript	47	53	31	63
PL/SQL	37	35	13	60
SQL	21	21	13	37

Sumber: <https://www.qsm.com/resources/function-point-languages-table> (diakses 18 November 2024)

e. Menentukan Resource yang Terkait dengan Pengembangan Perangkat Lunak

Dalam pengembangan sebuah perangkat lunak tentu akan melibatkan beberapa tim inti termasuk dalam pengembangn aplikasi *Digital Signage* ini, yang dapat dilihat pada Tabel 4.

Tabel 4 Estimasi tim pengembang dan tarif

No	Team Project	Tarif / FP
1	<i>Project Manager</i>	150,000,-
2	<i>System Analyst</i>	125,000,-
3	<i>Programmer Back End</i>	112,500,-
4	<i>Programmer Front End</i>	75,000,-

Dari Tabel 4, dapat dilihat estimasi biaya yang harus dikeluarkan untuk menyelesaikan sebuah perangkat lunak. Nilai tersebut bergantung pada kebijakan tim terkait dengan proses pengembangan perangkat lunak, dan sifatnya tentu saja dapat berbeda-beda.

HASIL DAN PEMBAHASAN

Pada penelitian ini akan menghitung *function point* dari aplikasi *digital signage* yang merupakan aplikasi yang digunakan untuk pengiriman konten dari kantor pusat ke *outlet* yang tersebar di seluruh Indonesia. Pada proses awal melibatkan identifikasi dan penghitungan jumlah fungsi dan kompleksitas setiap *User Function (UF)* berdasarkan kategori kompleksitas tinggi, menengah, dan rendah.

Tipe Komponen	Deskripsi	Kompleksitas	Values	Tipe Komponen	Deskripsi	Kompleksitas	Values
External Input (EI)	Form Login	Low	3	External Inquiry (EQ)	Check Availability	High	6
	Form Master Agency	Medium	4		Check Order Status	Medium	4
	Form Master City	Low	3		Check Delivery Content	Medium	4
	Form Master Client	Low	3		Log Report	High	6
	Form Product	Medium	4		Download Materi form Server	Medium	4
	Form Product Category	Low	3		Convert to MP4	High	6
	Form Production House	Low	3		Combine Conten	High	6
	Form Rate	Medium	4		Calculation Amount Rate	High	6
	Form Slots	Low	3		Check Sequence Content	High	6
	Form Screen Formation	Low	3		Check F&B Content when Generate	High	6
	Form Theater Company	Low	3		Check Promo Content when Generate	High	6
	Form Theater & Screen	Low	3		Check Iklan Content when Generate	High	6
	Form Campaign	Low	3		Check Corcom Content when Generate	High	6
	Form Order List	High	6		TOTAL		72
	Form Details Order	High	6	Internal Logical File (ILF)	Generate Playlist Synplay	High	15
	Form Placement	High	6		Generate Playlist Regular	High	15
	Form Assign Creative	High	6		Convert to MP4	High	15
	Form Save Placement	Low	3		Service Login	Medium	10
	Form Generate PlayList SynPlay	High	6		Service Agency	Medium	10
	Form Generate Playlist Regular	High	6		Service City	Medium	10
	Form Management User	Medium	4		Service Client	Medium	10
	Form Search	Medium	4		Service Product	Medium	10
	Form Sub Agency	Low	3		Service Product Category	Medium	10
	Form Add PIC	Low	3		Service Theater & Screen	Medium	10
	Form Add Product Version	Low	3		Service Company	Medium	10
	Upload Product Version	Medium	4		Service Rate	Medium	10
	TOTAL		102		Service Order	High	15
External Output (EO)	Document Penawaran	Medium	5		Service Campaign	Medium	10
	Invoice	Medium	5		Service Order Details	High	15
	Formulir Control Pemasangan Iklan	Medium	5		Service Save Placement	High	15
	Arsip Keuangan	Medium	5		Service Log	High	15
	Document Log	High	7		Auth Service	High	15
	Raw Log	High	7		Gateway Service	High	15
	Feed Orders	High	7		TOTAL		235
	Feed Schedule	High	7	External Interface File (EIF)	Integrasi Oracle Service	High	10
	Feed Creative	High	7		Integrasi Outlet Service	High	10
	Feed outlet	High	7		Outlet Service	High	10
	Feed City	High	7		Integrasi M-tix Data	High	10
	Feed Download Creative File	High	7		TOTAL		40
	TOTAL		76		GRAND TOTAL		525

Gambar 1 Komponen-komponen function point

Pada Gambar 1, penghitungan *Unadjusted Function Point (UFP)* dilakukan dengan melihat setiap UF yang terklasifikasi sesuai tingkat kompleksitasnya diberi bobot yang telah ditentukan, lalu dihitung nilai Total *Unadjusted Function Point (TUFP)* dengan menjumlahkan hasil perkalian

bobot dan jumlah fitur di setiap UF. Adapun rumus untuk menghitung UFP dapat dilihat pada Persamaan 1.

Penyesuaian kompleksitas teknis yang menggunakan 14 faktor *Technical Complexity Adjustment* (TCA), seperti keamanan dan kemampuan pemulihan sistem, yang masing-masing diberi nilai dari 0 hingga 5 untuk menentukan dampaknya pada keseluruhan sistem.

Tabel 5 Faktor Technical Complexity Adjustment (TCA)

No	Karakteristik	Bobot
1	<i>Data communications</i> : seberapa banyak fungsi aplikasi yang memerlukan komunikasi data.	4
2	<i>Distributed data processing</i> : tingkat distribusi pemrosesan data dalam aplikasi	5
3	<i>Performance</i> : kebutuhan kinerja yang diharapkan dari aplikasi.	5
4	<i>Heavily used configuration</i> : frekuensi penggunaan perangkat keras tertentu.	4
5	<i>Transaction rate</i> : tingkat atau frekuensi transaksi yang tinggi.	5
6	<i>On-line data entry</i> : seberapa banyak fungsi yang memerlukan input data langsung (online).	5
7	<i>End-user efficiency</i> : desain aplikasi untuk meningkatkan efisiensi pengguna.	4
8	<i>On-line update</i> : fungsi yang memperbarui data secara langsung.	5
9	<i>Complex processing</i> : tingkat kompleksitas pemrosesan dalam aplikasi.	5
10	<i>Reusability</i> : kebutuhan akan fitur atau fungsi yang dapat digunakan kembali.	4
11	<i>Installation ease</i> : kemudahan instalasi dan implementasi sistem.	4
12	<i>Operational ease</i> : kemudahan operasi sehari-hari dari sistem.	4
13	<i>Multiple sites</i> : kebutuhan untuk mendukung penggunaan di beberapa lokasi.	5
14	<i>Facilitate change</i> : kemampuan sistem untuk beradaptasi dengan perubahan.	4
Total		63

Tabel 5 menunjukkan nilai dari masing-masing Faktor *Technical Adjustment* (TCA) yang mana poin tersebut diberikan sesuai dengan tingkat pengaruh terhadap aplikasi. Nilai *Function Point* (FP) diperoleh dengan mengalikan *Unadjusted Function Point* (UFP) dengan nilai TCA, yang kemudian dapat digunakan untuk memperkirakan ukuran perangkat lunak dalam satuan *Lines of Code* (LOC), biaya, sumber daya, serta waktu yang diperlukan untuk pengerjaan. Rumus untuk menghitung FP disajikan pada Tabel 6.

Tabel 6 Perhitungan Function Point

Tipe Komponen	Level Kompleksitas									Total CFP
	Sederhana			Menengah			Kompleks			
	Jml	Bobot	Point	Jml	Bobot	Point	Jml	Bobot	Point	
External Input (EI)	14	3	42	6	4	24	6	6	36	102
External Output (EO)	0	4	0	4	5	20	8	7	56	76
External Inquiry (EQ)	0	4	0	3	4	12	10	6	60	72
Internal Logical File (ILF)	0	7	0	10	10	100	9	15	135	235
External Interface File (EIF)	0	5	0	0	7	0	4	10	40	40
TOTAL										525

Pada Tabel 6 di atas terdapat perhitungan antara masing-masing komponen dengan faktornya, sehingga terdapat nilai sebesar 525. Setelah mengetahui variabel dalam hitungan *Function Point*, maka kita masukkan ke formula pada Persamaan 2 dan Persamaan 3, sehingga menghasilkan nilai FP sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times TDI) \\
 &= 525 \times (0.65 + 0.01 \times 63) \\
 &= 525 \times (1,28) \\
 &= \mathbf{672}
 \end{aligned}$$

Konversikan *function point* kedalam estimasi biaya pengembangan aplikasi *digital signage* sebagai terdapat pada Tabel 7.

Tabel 7 Biaya tim pengembangan *digital signage*

No	Tim	Tarif/FP	Jam/FP	Alokasi SDM
1	<i>Project manager</i>	150,000,-	1	1
2	<i>System analyst</i>	125,000,-	1	1
3	<i>Programmer Back-End</i>	112,500,-	1	2
4	<i>Programmer Front-End</i>	75,000,-	1	1

Berdasarkan Tabel 7, dapat dihitung estimasi biaya tim pengembangan perangkat lunak sebagai berikut:

Project Manager	: 150,000 x 672 = Rp 100,800,000,-
System Analyst	: 125,000 x 672 = Rp 84,000,000,-
Programmer Back End	: 112,500 x 672 = Rp 151,200,000 ,-
Programmer Front End	: 75,000 x 672 = Rp 50,400,000,-

Jadi Estimasi Nilai Pengembangan project sebesar Rp 386,400,000,-.

Estimasi Development Aplikasi *Digital Signage* = 1 x 425.5 = 672 Jam.

Estimasi jam kerja = 8 Jam

Estimasi Hari Kerja = 20 Hari

Maka 672 Jam / 8 Jam = 84 Hari

Maka 84 Hari / 20 Hari = 4 Bulan 4 Hari Kerja

Adapun untuk menghitung LOC dihitung dengan Persamaan 4.

$$LOC = FP \times Productivity Factor \quad (4)$$

$$LOC = 672 \times 53 = 35616$$

Ukuran proyek telah diestimasi sebesar 35,616 baris kode menggunakan metode *Function Point Analysis (FPA)*. Setelah proyek selesai, ukuran aktual dari proyek tercatat sebesar 36,213 baris kode. Jika dibandingkan, perbedaan antara nilai estimasi dan ukuran aktual adalah sekitar 1.68%. Hasil estimasi ini mendekati hasil penelitian sebelumnya yang menunjukkan perbedaan sebesar 1.5%. Meskipun demikian, mengingat proyek yang dihitung merupakan aplikasi berskala besar dan berbeda, perbedaan antara estimasi dan ukuran aktual tetap mungkin terjadi.

SIMPULAN

Penelitian ini menunjukkan bahwa metode *Function Point Analysis (FPA)* dapat diterapkan secara efektif untuk mengestimasi ukuran perangkat lunak, biaya, waktu, dan sumber daya yang dibutuhkan dalam pengembangan sistem *digital signage* berbasis web. Hasil estimasi menunjukkan tingkat akurasi yang tinggi, dengan selisih hanya 1.68% antara estimasi ukuran proyek (35,616 baris kode) dan ukuran aktual (36,213 baris kode). Dengan estimasi biaya sebesar Rp 386,400,000,- dan durasi pengerjaan selama 4 bulan 4 hari kerja, FPA terbukti sebagai alat perencanaan yang bermanfaat untuk mengurangi risiko kegagalan serta kerugian dalam proyek perangkat lunak berskala besar. Hasil penelitian ini menunjukkan bahwa FPA tidak hanya efektif untuk estimasi teknis dalam proyek perangkat lunak seperti *digital signage*, tetapi juga memiliki implikasi yang

lebih luas sebagai standar manajemen proyek teknologi informasi lintas sektor. Dengan pendekatan kuantitatif dan objektif, FPA mendukung pengambilan keputusan strategis yang lebih akurat dalam perencanaan anggaran, alokasi sumber daya, serta mitigasi risiko, sekaligus memperkuat tata kelola teknologi informasi dan komunikasi antara tim teknis dan manajerial. Potensi ini menjadikan FPA relevan dalam transformasi digital dan evaluasi investasi teknologi pada berbagai skala dan sektor industri.

Penelitian selanjutnya disarankan untuk mengintegrasikan FPA dengan teknologi modern, seperti *machine learning*, guna meningkatkan akurasi estimasi dan mengatasi tantangan dalam klasifikasi fungsi serta tingkat kompleksitas. Selain itu, penelitian dapat diperluas ke proyek dengan skala yang berbeda untuk menguji generalisasi metode ini. Pengembangan alat otomatisasi perhitungan FPA juga direkomendasikan untuk meningkatkan efisiensi dan meminimalkan kesalahan manual, sekaligus mengukur dampak *digital signage* terhadap efisiensi operasional dan kepuasan pengguna.

DAFTAR PUSTAKA

- Fachrurrazi S, Hizli H. 2021. Digital Signage Sebagai Media Layanan Informasi. *Sisfo J. Ilm. Sist. Inf.* 5, 33–43. <https://doi.org/10.29103/sisfo.v5i2.6226>.
- Hapsari RK, Husen MJ. 2015. Estimasi Kualitas Perangkat Lunak Berdasarkan Pengukuran Kompleksitas Menggunakan Metrik Function Oriented. *Semin. Nas. Sains dan Teknol. Terap.* III, 425–434.
- International Function Point Users Group (IFPUG). 2024. Function Point Analysis (FPA) [WWW Document]. IFPUG. URL <https://ifpug.org/ifpug-standards/fpa> (accessed 11.18.24).
- Longstreet D. 2005. *Fundamentals & Introduction of Function Point Analysis*. SoftwareMetrics.
- Muhajirin N, Daryono W. 2018. The use of Digital Signage as Media Campus Information (Penggunaan Digital Signage sebagai Media Informasi Kampus). *J. Pekommas.* 3(1). <https://doi.org/10.30818/jpkm.2018.2030101>.
- Ningrum KP. 2022. Digital Signage : Makna, Strategi Pembuatan Pesan, Dan Implementasi Oleh Dentsu Agency. *Media Bahasa, Sastra, dan Budaya Wahana* 28, 624–632. <https://doi.org/10.33751/wahana.v28i1.5223>.
- Rachmat N, Saparudin. 2017. Estimasi Ukuran Perangkat Lunak Menggunakan Function Point Analysis-Studi Kasus Aplikasi Pengujian dan Pembelajaran Berbasis Web. *Annu. Res. Semin.* 3, 1–5.
- Rambu BH, Mayank A, Soetam R. 2019. Analisis Kualitas Perangkat Lunak Menggunakan Metode Function Point Analysis (Studi Kasus: Transaksi Pembelian di eBay). *Jurnal Informatika dan RPL.* 1(1), 31–36.
- Roodasri BP. 2021. Web-based digital signage design and implementation for food and beverage industry. 1–54.
- Taufiqurrahman RF. 2022. Estimasi Pengukuran Perangkat Lunak Menggunakan Metode Matrik Function Point Analysis – Studi Kasus Aplikasi Akademik Siswa Berbasis Website. *J. Sist. Inf. dan Sains Teknol.* 4, 1–15. <https://doi.org/10.31326/sistek.v4i2.1325>.