

PENERAPAN REQUIREMENT CHANGE MANAGEMENT PROCESS DAN FUNCTION POINT ANALYSIS PADA BACKEND SISTEM MANAJEMEN PROYEK (STUDI KASUS PERUSAHAAN XYZ)

Ahmad Rafaly¹, Wahyu Eko Sulistiono², Resty Annisa³, Gigih Forda Nama³

^{1,2,3} Universitas Lampung; Jl. Prof. Dr. Ir. Sumantri Brojonegoro No.1, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung 35141; Telp. (0721) 701609

Keywords:

Requirement change management;
Function Point Analysis;
Sistem Manajemen Proyek;
Software Cost Estimation;

Correspondent Email:

2215061011@unila.ac.id

Abstrak. Perubahan kebutuhan perangkat lunak yang tidak dikelola dengan baik seringkali menyebabkan keterlambatan dan pembengkakan biaya proyek. Untuk mengatasi masalah tersebut di perusahaan XYZ, penelitian ini mengembangkan *backend* Sistem Manajemen Proyek berbasis Laravel dengan menerapkan metode *Requirement Change Management* (RCM) dan *Function Point Analysis* (FPA). Pendekatan RCM digunakan untuk mendokumentasikan dan mengendalikan usulan perubahan, sementara FPA dimanfaatkan untuk mengukur kompleksitas serta memberikan estimasi waktu dan biaya di awal proyek. Hasil pengujian fungsional menunjukkan bahwa seluruh fungsi *backend* berjalan sempurna tanpa kesalahan. Selain itu, validasi estimasi pada dua proyek riil (Sistem X dan Y) membuktikan bahwa implementasi FPA pada sistem mampu menghasilkan tingkat akurasi estimasi rata-rata sebesar 89% untuk biaya dan 84% untuk waktu.



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. Poorly managed software requirements changes often lead to project delays and cost overruns. To solve these issues at XYZ Company, this study developed a Laravel-based Project Management System backend by implementing the Requirement Change Management (RCM) and Function Point Analysis (FPA) methods. The RCM approach was used to document and control proposed changes, while FPA was utilized to measure complexity and provide time and cost estimates at the beginning of the project. The functional testing results showed that all backend functions ran perfectly without errors. In addition, estimation validation on two real projects (System X and Y) proved that the implementation of FPA in the system was able to produce an average estimation accuracy rate of 89% for cost and 84% for time.

1 PENDAHULUAN

Perubahan kebutuhan adalah hal yang tidak dapat dihindari dalam pengembangan perangkat lunak [1], terutama di perusahaan teknologi seperti perusahaan XYZ. Kebutuhan terhadap suatu *software* sering berubah akibat regulasi baru maupun perkembangan layanan digital. Jika perubahan tidak dikelola dengan baik, perubahan tersebut dapat menyebabkan keterlambatan, pembengkakan biaya, dan hasil yang tidak sesuai harapan [1]. Penelitian ini membangun Sistem Manajemen Proyek berbasis Laravel dengan menerapkan konsep *Requirement Change Management* dan *Function Point Analysis* [2] sesuai dengan standar ISO 21502:2020 [3]. Penelitian hanya berfokus pada pengembangan *backend* menggunakan teknologi Laravel yang mengatur logika dan alur data pada sistem.

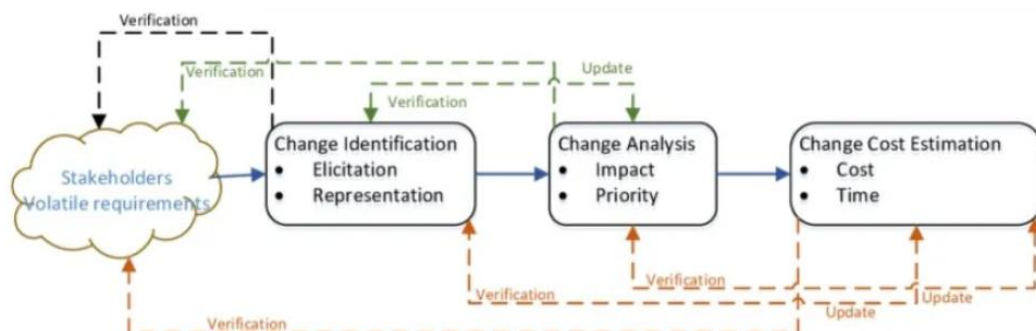
Terdapat beberapa penelitian yang telah membahas konsep *Requirement Change Management* [4][5] dan telah membuat beberapa sistem manajemen proyek yang sudah sangat terkenal seperti Jira [6] dan Trello [7]. namun masih terdapat kesenjangan penelitian berupa belum adanya sistem manajemen proyek yang mengimplementasikan konsep RCM dan dapat membuat estimasi biaya serta waktu yang dibutuhkan dalam suatu proyek menggunakan konsep *Function Point Analysis* (FPA) karena metode FPA terbukti memiliki akurasi yang

cukup baik [8][9] dalam memberikan estimasi awal dari segi waktu dan biaya, jauh sebelum proyek tersebut diimplementasikan. Melalui penelitian ini, diharapkan dapat memberikan kontribusi sebagai model sistem manajemen proyek yang tidak hanya dapat mengelola dan mencatat perubahan kebutuhan namun juga dapat memperkirakan biaya dan waktu yang dibutuhkan dari suatu proyek perangkat lunak berbasis *website* maupun *aplikasi*.

2 TINJAUAN PUSTAKA

2.1 *Requirement Change Management (RCM)*

Requirement Change Management (RCM) merupakan kegiatan pengelolaan perubahan kebutuhan sepanjang siklus pengembangan perangkat lunak [1]. Tujuan RCM adalah memastikan setiap usulan perubahan dianalisis, terdokumentasi, serta terkendali sehingga tidak menyimpang dari sasaran proyek. Berdasarkan Gambar 1, *RCM Process* mencakup tiga tahapan utama, yaitu identifikasi perubahan, analisis dampak, dan estimasi sumber daya [1]. Identifikasi menjadi tahap paling krusial karena menentukan kejelasan masalah dan kebutuhan yang diusulkan. Selanjutnya, analisis dampak dilakukan untuk menilai konsekuensi perubahan terhadap komponen lain dalam sistem. Tahap akhir adalah estimasi biaya dan waktu tambahan yang diperlukan untuk menerapkan suatu perubahan.



Gambar 1 RCM Process [1]

2.2 Function Point Analysis (FPA)

Function Point Analysis merupakan bagian dari metode *Functional Size Measurement* yang pertama kali diperkenalkan oleh Albrecht [10] sebagai metode untuk mengukur tingkat kompleksitas dari fungsionalitas suatu proyek perangkat lunak. Adapun rumus dalam menghitung *Function Point* yang telah diperbarui IFPUG [2] adalah sebagai berikut :

$$FP = UFP \times ((0,01 \times TDI) + 0,65)(1)$$

$FP = \text{Function Point}$

$UFP = \text{Unadjusted Function Points}$

$TDI = \text{Total degree influence}$

Unadjusted Function Point menghitung ukuran sistem perangkat lunak berdasarkan lima komponen utama. Setiap komponen dinilai dengan bobot tertentu sesuai tingkat kompleksitasnya, yang dibagi menjadi tiga kategori: *Simple*, *Average*, dan *Complex* [10]. Lima komponen tersebut meliputi:

- 1) *External Input* (EI) yaitu data masukan yang diterima dari luar sistem.
- 2) *External Output* (EO) yaitu hasil keluaran yang dihasilkan oleh sistem, misalnya notifikasi atau *file*.
- 3) *External Inquiry* (EQ) berupa permintaan data dari pengguna terhadap sistem tanpa melakukan perubahan apapun pada data.
- 4) *Internal Logical File* (ILF) mencakup jumlah berkas atau basis data internal yang digunakan oleh sistem.
- 5) *External Interface File* (EIF) yaitu jumlah berkas atau basis data eksternal yang terhubung dan berinteraksi dengan sistem.

Masing-masing komponen ini dikalikan dengan bobot kompleksitas implementasinya yang dikategorikan sebagai berikut :

Tabel 1 Bobot *Unadjusted Function Point* [2]

Parameter	<i>Simple</i>	<i>Average</i>	<i>Complex</i>
<i>External Input</i>	3	4	6
<i>External Output</i>	4	5	7
<i>Logical Internal files</i>	7	10	15
<i>External Interface files</i>	5	7	10
<i>External Inquiry</i>	3	4	6

General System Characteristics digunakan untuk menghitung kompleksitas umum sistem yang memiliki pengaruh dari skala 0 hingga 5 yang selanjutnya diakumulasi menjadi sebuah nilai bernama *Total Degree Influence*. Adapun kompleksitas GSC dan keterangan skala dijelaskan pada tabel di bawah :

Tabel 2 Karakteristik GSC [2]

No	Karakteristik
1	Tingkat kompleksitas komunikasi data
2	Tingkat kompleksitas pemrosesan terdistribusi
3	Tingkat kompleksitas performa
4	Tingkat kompleksitas konfigurasi

5	Tingkat frekuensi penggunaan perangkat lunak
6	Tingkat frekuensi <i>input</i> data
7	Tingkat kemudahan penggunaan bagi pengguna
8	Tingkat frekuensi perbarui data
9	Tingkat kompleksitas pemrosesan data
10	Tingkat <i>reusable code</i> program
11	Tingkat kemudahan dalam instalasi
12	Tingkat kemudahan operasional perangkat lunak
13	Tingkat perangkat lunak dibuat untuk <i>multi</i> organisasi
14	Tingkat kompleksitas dalam mengadaptasi perubahan

Tabel 3 Keterangan Skala GSC [2]

Nilai Skala	Keterangan
0	Tidak berpengaruh
1	Pengaruh tidak signifikan
2	Pengaruh Moderat
3	Pengaruh rata – rata
4	Berpengaruh Signifikan
5	Berpengaruh Kuat

2.3 Software Cost Estimation

Perhitungan estimasi waktu pada penelitian ini, menggunakan rumus yang diberikan oleh Ali [11] yaitu sebagai berikut :

$$E = \left\{ Nw \times \left(\frac{factor \times FP}{n} \right) \right\} [11]$$

E = Perkiraan waktu pengembangan (jam)

Nw = Waktu pengerjaan dalam sehari

$Factor$ = Baris kode pada perangkat lunak per *Function Point*

FP = *Function Point*

n = Jumlah baris kode per hari

Persamaan untuk menghitung estimasi biaya yaitu sebagai berikut :

$$\Delta C_D = E \times W_p [11] (3)$$

ΔC_D = Estimasi biaya

E = Perkiraan waktu pengembangan (jam)

W_p = Biaya pengembangan per jam

Factor adalah jumlah baris kode pada suatu bahasa pemrograman per *Function Point* [11] Adapun nilai *Factor* untuk setiap bahasa pemrograman, menggunakan acuan yang dibuat oleh Capers Jones pada tabel di bawah:

Tabel 4 *Factor* Bahasa Pemrograman [12]

Bahasa Pemrograman	<i>Factor Productivity</i>
Python	53
HTML	160
PHP	53
Javascript	71

2.4 Sistem Manajemen Proyek

Manajemen Proyek adalah bidang yang melibatkan proses perencanaan, pengorganisasian, pengendalian untuk mencapai tujuan yang telah ditetapkan di dalam suatu proyek [13]. Sementara itu, sistem manajemen proyek adalah sebuah

sistem yang dibuat dengan tujuan merencanakan dan mengendalikan proses dari awal hingga akhir proyek. Ruang lingkup ini mencakup aspek teknis, pendokumentasian, dan diseminasi informasi sesuai kebutuhan organisasi. Implementasi sistem manajemen proyek akan memudahkan perusahaan untuk melakukan supervisi terhadap berbagai aktivitas di lokasi yang berbeda, sekaligus memfasilitasi pelaporan kemajuan proyek serta efisiensi waktu dan biaya [13]. Berdasarkan standar yang telah ditetapkan pada ISO 21502:2020, sistem manajemen proyek harus menggabungkan praktek pengarahannya, perencanaan, pemantauan, pengendalian, dan penutupan proyek [3].

3 METODE PENELITIAN

3.1 Pengembangan Sistem

Penelitian berfokus pada *backend* dimana data disimpan dan dikelola pada *database* PostgreSQL, kemudian diakses oleh model sebagai representasi data dan logika. Model menyediakan data kepada *controller*, yang bertugas memproses logika dan menyiapkan respon untuk permintaan yang masuk. Seluruh permintaan dari luar diarahkan terlebih dahulu ke *Route* yang berfungsi untuk menentukan *controller* mana yang harus menangani permintaan tersebut, *frontend* yang menggunakan ReactJS berperan sebagai antarmuka pengguna yang mengirimkan *request* ke *Route* pada *backend*.

3.2 Pengujian Sistem

Pengujian akurasi estimasi awal biaya dan waktu *function point* dilakukan dengan membandingkan hasil perhitungan dengan 2 data proyek asli berbasis *website* yang sama-sama menggunakan teknologi laravel dengan *developer* yang sama yaitu sistem X dan Y.

Pada tahap pengujian fungsional sistem, dilakukan pengujian *blackbox testing* untuk mengevaluasi

fungsionalitas sistem dengan cara membandingkan kesesuaian antara *input* dan *output* dari sistem. Melalui penerapan pengujian *blackbox* ini, dapat dipastikan bahwa sistem yang dikembangkan berfungsi dengan baik serta sesuai dengan kebutuhan pengguna. Selain pengujian terhadap fungsional sistem, terdapat juga pengujian *Unit Testing* untuk memastikan setiap fungsi pada sistem dapat berjalan sesuai dengan yang diharapkan. Pengujian dilakukan menggunakan *feature* bawaan Laravel yaitu PHPUnit, jika hasil *output* sesuai dengan yang diharapkan, maka fungsi tersebut telah memenuhi kriteria fungsionalitas yang benar.

3.3 Pengukuran Estimasi Biaya dan Waktu menggunakan Function Point

Metode Function Point digunakan untuk mengukur kompleksitas ukuran perangkat lunak [14] yang berfokus pada fungsional sistem [15] baik *website* [16][17][18][19] maupun aplikasi [20][21][22] bergantung pada seberapa baik dokumentasi *software*-nya [23][24]. Terdapat penelitian yang menggabungkan *Function Point* dengan EVM [25] yang dapat digunakan untuk menjaga waktu dan biaya selama proyek berlangsung, namun pengukuran estimasi biaya dan waktu suatu proyek pada penelitian ini didapat dengan melakukan perhitungan menggunakan rumus *Function Point* dan *Software Cost Estimation* [11]. Langkah – langkah yang dilakukan yaitu sebagai berikut :

- 1) Menghitung bobot masing-masing fitur menggunakan *Unadjusted Function Point*

Langkah pertama adalah menghitung ukuran fungsional dari setiap fitur sistem menggunakan *Unadjusted Function Point* yang akan diisi oleh *Project Manager* di perusahaan XYZ, UFP diperoleh dengan menjumlahkan seluruh fitur sistem berdasarkan lima komponen

- 2) Menghitung karakteristik sistem menggunakan TDI

Tahap ini digunakan untuk menilai kompleksitas sistem secara keseluruhan berdasarkan 14 faktor umum sistem, Setiap faktor diberi skor 0–5 tergantung tingkat pengaruhnya terhadap sistem.

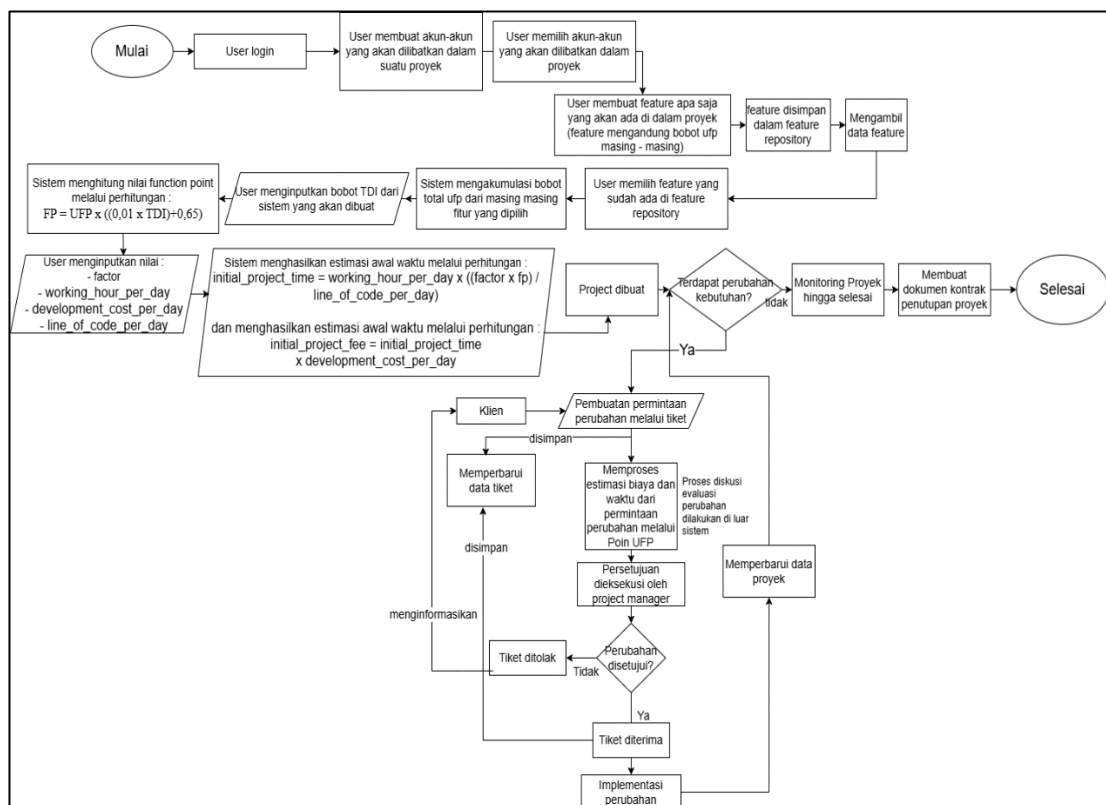
3) Menghitung Nilai *Function Point*
Setelah memperoleh nilai UFP dan TDI, langkah selanjutnya yaitu menghitung nilai *Function Point* dengan rumus (1)

4) Menghitung Estimasi Biaya dan Waktu menggunakan nilai dari *Function Point*

Nilai FP yang telah diperoleh selanjutnya digunakan untuk dimasukkan ke dalam rumus *Software Cost Estimation* [11] dengan rumus (2) dan (3).

4 HASIL DAN PEMBAHASAN

Berdasarkan Gambar 2, alur sistem dimulai dengan pengguna melakukan autentikasi melalui menu *login* untuk kemudian mengelola sumber daya manusia dengan membuat serta memilih akun-akun tim yang akan didelegasikan ke dalam proyek. Setelah tim terbentuk, pengguna menentukan fitur apa saja yang akan dikembangkan dalam proyek tersebut. Penentuan fitur ini dapat dilakukan dengan mengambil data fitur yang sudah tersedia di dalam *Feature Repository*. Sistem secara otomatis mengakumulasi total bobot UFP dari seluruh fitur yang dipilih. Pengguna kemudian mengisi *field* TDI serta *field* lainnya seperti bahasa pemrograman, jam kerja sehari, *dev cost* per jam, dan baris kode sehari.



Gambar 2 Alur tahapan sistem

4.1 Blackbox Testing

Metode pengujian pertama yang digunakan pada penelitian ini adalah *Black Box Testing* agar sistem diuji

berdasarkan *input* yang diberikan, proses yang dilakukan, serta output yang dihasilkan. Berdasarkan hasil pengujian

pada 30 fitur, semua fitur telah sesuai dengan hasil yang diharapkan pengguna.

4.2 Unit Testing

Pengujian sistem juga dilakukan menggunakan *unit testing* untuk menilai apakah sistem sudah memiliki kontrol data yang sudah sesuai dengan kebutuhan perusahaan. Berdasarkan hasil *unit testing* pada sisi *backend* sistem, seluruh fungsi yang diuji menunjukkan status semua *Passed* dengan maksimal waktu tidak lebih dari 2 detik. Pengujian dilakukan pada setiap fungsi *controller* menggunakan PHPUnit untuk memastikan bahwa setiap proses berjalan sesuai dengan kebutuhan fungsional sistem.

4.3 Hasil estimasi awal total waktu dan biaya Function Point terhadap kondisi aktual

Sistem yang dibangun berhasil memberikan estimasi awal total waktu dan biaya dengan mengintegrasikan *Function Point* yang disesuaikan dengan parameter produktivitas internal perusahaan XYZ. Hasil perbandingan antara realisasi dan estimasi baik dari segi biaya dan waktu menggunakan perhitungan berikut :

$$\begin{aligned} & \text{Persentase Akurasi(\%)} \\ &= \left| \left(\frac{|\text{Estimasi} - \text{Realisasi}|}{\text{Estimasi}} \right) \times 100 \right. \\ & \left. - 100 \right| (4) \end{aligned}$$

Keandalan mekanisme ini telah divalidasi melalui pengujian pada Sistem X dan Y, di mana sistem mampu menghasilkan akurasi estimasi total biaya yang diperlukan untuk membangun sistem X sebesar 86% dengan akurasi waktu 90% serta pada sistem Y memiliki akurasi waktu 93% dengan biaya 78% dan jika kedua akurasi sistem dirata-ratakan maka menghasilkan persentase akurasi estimasi total waktu 84% dan biaya 89%. Persentase akurasi ini sangat

dipengaruhi oleh kecakapan pengguna dalam menganalisa jumlah dan tingkat kompleksitas komponen UFP dan TDI di dalam FPA, karena jika terjadi kesalahan dalam menganalisa komponen tersebut, hasil analisa bisa sangat jauh atau bahkan tidak masuk akal dari yang semestinya.

5 KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa kesimpulan yaitu sistem yang dibangun telah mengimplementasikan *Requirement Change Management Process* sehingga dapat digunakan untuk menangani setiap perubahan kebutuhan yang muncul. Serta berdasarkan hasil implementasi *Function Point Analysis* pada sistem, estimasi biaya dan waktu yang dibutuhkan untuk membangun sistem X dan Y mampu melakukan perhitungan dengan akurasi rata rata biaya yaitu 89% dan waktu mencapai 84%.

DAFTAR PUSTAKA

- [1] S. Jayatilleke and R. Lai, "A systematic review of requirements change management," *Journal Information and Software Technology*, vol. 93, pp. 163–185, 2017, doi: 10.1016/j.infsof.2017.09.004.
- [2] IFPUG, "Function point counting practices manual, release 4.3.1," USA, 2010.
- [3] ISO 21502, "Project, programme and portfolio management-Guidance on project management Management," 2020.
- [4] N. Mehmood Minhas, Q. Ul ain, Z. Ul Islam, and A. Zulficar, "An Improved Framework for Requirement Change Management in Global Software Development," *Journal of Software Engineering and Applications*, vol. 07, no. 09, pp. 779–790, 2014, doi: 10.4236/jsea.2014.79072.
- [5] A. A. Khan, S. Basri, and P. D. D. Dominic, "A propose framework for requirement change management in global software development," in *2012 International Conference on Computer and Information Science, ICCIS 2012 - A*

- Conference of World Engineering, Science and Technology Congress, ESTCON 2012 - Conference Proceedings*, 2012, pp. 944–947. doi: 10.1109/ICCISci.2012.6297161.
- [6] Atlassian, “Jira, Best Project Management Tool,” 2024, Sydney. Accessed: Jan. 16, 2026. [Online]. Available: <https://www.atlassian.com/software/jira>
- [7] Atlassian, “About Trello,” 2024. Accessed: Jan. 16, 2026. [Online]. Available: <https://trello.com/about>
- [8] A. Yhurinda, P. Putri, and A. P. Subriadi, “Software Cost Estimation Using Function Point Analysis,” in *International Seminar on Science and Technology*, Surabaya: Postgraduate Program Institut Teknologi Sepuluh Nopember, 2018, pp. 79–83.
- [9] N. Rachmat, “Estimasi Ukuran Perangkat Lunak Menggunakan Function Point Analysis-Studi Kasus Aplikasi Pengujian dan Pembelajaran Berbasis Web,” in *Seminar ilkom unsri*, Universitas Sriwijaya, 2017. [Online]. Available: <https://www.researchgate.net/publication/320825809>
- [10] A. J. Albrecht and J. E. Gaffney, “Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation,” *IEEE Transactions On Software Engineering*, vol. 9, no. 6, pp. 639–648, 1983.
- [11] N. Ali and R. Lai, “A method of requirements change management for global software development,” *Inf. Softw. Technol.*, vol. 70, pp. 49–67, 2016.
- [12] C. Jones, “Software Economics and Function Point Metrics: Thirty years of IFPUG Progress,” Namcook Analytics LLC, United State, 2017. Accessed: Jan. 14, 2026. [Online]. Available: <https://www.ifpug.org/wp-content/uploads/2017/04/IYSM.-Thirty-years-of-IFPUG.-Software-Economics-and-Function-Point-Metrics-Capers-Jones.pdf>
- [13] I. Andika, S. Nevile, R. Satya, and A. Farisi, “Analisis Sistem Informasi Manajemen Proyek: Systematic Literature Review Information System Management Project Analysis: Systematic Literature Review,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 11, no. 1, pp. 220–230, 2024.
- [14] S. Atin, T. Harihayati, and U. D. Widiyanti, “Utilization of function point method for measuring software project complexity,” in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Sep. 2018. doi: 10.1088/1757-899X/407/1/012086.
- [15] R. K. Hapsari, D. M. Jauhari, and H. Wp, “Estimasi Kualitas Perangkat Lunak Berdasarkan Pengukuran Kompleksitas Menggunakan Metrik Function Oriented,” in *Seminar Nasional Sains dan Teknologi Terapan III 2015*, Institut Teknologi Adhi Tama Surabaya, 2015, pp. 425–434.
- [16] R. R. Mahendra *et al.*, “Pengukuran Kualitas Perangkat Lunak Website Pendataan Ekskul Siswa Menggunakan Function Point,” *Jurnal Ilmiah Informatika*, vol. 11, no. 1, pp. 1–14, 2023.
- [17] A. Sidqi, A. Yuswanto, and M. A. Yaqin, “Analisis Pengukuran Kompleksitas Website SOC Menggunakan Function Point,” *Journal Automation Computer Information System*, vol. 4, no. 2, pp. 67–75, Jul. 2024, doi: 10.47134/jacis.v4i2.82.
- [18] C. Lenny Dwi Rizka, F. Shinta Dewi, and S. Rizky Wicaksono, “Pengukuran dan Kualitas Perangkat Lunak Website ‘LINKEDIN’ Menggunakan Metode Function Point Analysis,” *JISKA*, vol. 3, no. 2, pp. 79–83, 2018.
- [19] H. Rohayani, F. Lumban, G. Benfano, S. Harco, and L. Hendric, “Estimated Measurement Quality Software On Structural Model Academic System With Function Point Analysis,” in *International Conference on Applied Computer and Communication Technologies (ComCom)*, IEEE, 2017, pp. 1–5.
- [20] A. P. Satria, K. Widodo, and A. Indrati, “Analysis Yummy App Site Functionality on Costs Use Method Function Point,” *International Research*

- Journal of Advanced Engineering and Science*, vol. 6, no. 1, pp. 164–168, 2021.
- [21] T. Wahyono, B. Soewito, S. W. H. L. Hendric, and F. L. Gaol, “Software complexity measurement of water poverty mapping application with function point method,” in *2017 International Conference on Applied Computer and Communication Technologies (ComCom)*, IEEE, May 2017, pp. 1–5. doi: 10.1109/COMCOM.2017.8167102.
- [22] A. P. Nugroho, S. Y. Muliana, Murtiningrum, and S. S. Arif, “Economic Analysis Based on Software Cost Estimation Model on the Development of Telemetry Equipment to Support the Irrigation Modernization,” in *Advances in Biological Sciences Research*, Atlantic Press, 2022, pp. 114–121. doi: 10.2991/absr.k.220305.017.
- [23] A. R. Irawati and K. Mustofa, “Measuring Software Functionality Using Function Point Method Based On Design Documentation,” *International Journal of Computer Science Issues*, vol. 9, no. 3, pp. 124–130, 2012.
- [24] A. P. Subriadi, Sholiq, V. Lukitosari, and R. Permatasari, “The Comparison Analysis Of Estimation Effort Among Software Development Using Function Point Method,” in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, IEEE, Nov. 2018, pp. 204–209. doi: 10.1109/ISRITI.2018.8864481.
- [25] S. Atin, Z. Priyanto, I. Afrianto, and V. Pratiwi, “Integration of Function Point and Earned Value Management Methods in Information Systems Project Development Evaluation,” in *2023 International Conference on Informatics Engineering, Science & Technology (INCITEST)*, IEEE, Oct. 2023, pp. 1–7. doi: 10.1109/INCITEST59455.2023.10395922.