



ESTIMASI USAHA PENGEMBANGAN PERANGKAT LUNAK APLIKASI CERDAS DESAKU MENGGUNAKAN METODE USE CASE POINTS DAN COSMIC

Amelia Hayyu Astiarasanti¹, Nur Alfiyah Miftakhul Jannah², Renny Sari Dewi³

Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Kreatif
Universitas Internasional Semen Indonesia

E-mail: amelia.astiarasanti17@student.uisi.ac.id, nur.jannah17@student.uisi.ac.id,
renny.dewi@uisi.ac.id

ABSTRACT

In the era of industrial revolution 4.0, technology played an important role in all sectors, especially government agencies. Government agencies are expected to be quickly adaptive to apply technology in business processes, one of which is in the village government that has now begun to be required to develop the technology. One form of technology that began to be developed to support the progress of the village is Cerdas Desaku application. The software development of Cerdas Desaku application certainly required business estimation which is the most important activity and is a supporting factor in the success of software development. The use of the business estimation method in this research uses use case points and COSMIC method that can be a reference in estimating software development efforts. The result of this research is an estimated effort using the use case points method obtained results of 637,300 man/hour and cosmic method obtained results of 4,765.92 man/hour.

Keywords: *COSMIC; Effort Estimates; Software; Use Case Points*

ABSTRAK

Pada era revolusi industri 4.0, teknologi sangat berperan penting bagi semua sektor khususnya pada instansi pemerintahan. Instansi pemerintahan diharapkan dapat secara cepat adaptif untuk menerapkan teknologi dalam proses bisnis, salah satunya pada pemerintahan desa yang saat ini sudah mulai dituntut untuk mengembangkan teknologi. Salah satu bentuk teknologi yang mulai dikembangkan untuk menunjang kemajuan desa adalah aplikasi cerdas desaku. Dalam pengembangan perangkat lunak aplikasi cerdas desaku tentu dibutuhkan estimasi usaha yang merupakan aktivitas yang paling penting dan merupakan faktor pendukung dalam keberhasilan pengembangan perangkat lunak. Penggunaan metode estimasi usaha pada penelitian ini menggunakan metode use case points dan COSMIC yang dapat menjadi acuan dalam melakukan estimasi usaha pengembangan perangkat lunak. Hasil dari penelitian ini adalah estimasi usaha menggunakan metode use case points didapatkan hasil sebesar 637,300 man/hour dan metode COSMIC didapatkan hasil sebesar 4.765,92 man/hour.

Kata Kunci: *COSMIC; Estimasi Usaha; Perangkat Lunak; Use Case Points*

I. PENDAHULUAN

Pada era revolusi industri 4.0, teknologi sangat berperan penting bagi semua sektor khususnya pada instansi pemerintahan. Instansi pemerintahan diharapkan dapat secara cepat adaptif untuk menerapkan teknologi dalam proses bisnis, salah satunya pada pemerintahan desa yang saat ini sudah mulai dituntut untuk mengembangkan teknologi guna meningkatkan kualitas pelayanan masyarakat [1]. Salah satu bentuk teknologi yang mulai dikembangkan untuk menunjang kemajuan desa adalah aplikasi cerdas desaku.

Aplikasi cerdas desaku merupakan aplikasi berbasis mobile yang memuat informasi mengenai acara, data instansi pendidikan, data panen lokal, dan juga topik diskusi yang membahas tentang permasalahan yang ada di desa. Aplikasi ini dikembangkan dengan harapan dapat meningkatkan pariwisata desa. Namun dalam pengembangannya dibutuhkan estimasi untuk mencegah adanya over-estimate dan under- estimate [2][3]. Apabila under-estimate terjadi dari seharusnya maka akan berakibat pada tidak terselesaikannya proyek dalam pengembangan perangkat lunak dari segi waktu, biaya maupun kualitas produk. Tetapi jika terjadi over-estimate, maka akan menyebabkan proyek dalam pengembangan perangkat lunak menjadi tidak realistis untuk dilaksanakan, oleh karena itu sangat dibutuhkannya kegiatan estimasi usaha [4].

Estimasi usaha merupakan aktivitas yang paling penting dalam proses pengembangan perangkat lunak dan menjadi faktor pendukung dalam keberhasilan suatu proyek. Estimasi usaha pengembangan perangkat lunak merupakan suatu proses memprediksi dalam pengaturan sumber daya agar proyek dapat berjalan sesuai dengan apa yang telah ditargetkan yakni tepat waktu, sesuai dengan biaya dan terpenuhinya standart kualitas produk [5]. Penelitian terdahulu yang dilakukan oleh Kurniadi dkk pada tahun 2017 [6], dimana penelitian tersebut melakukan pengukuran *size* dan *effort* pada *software* Student Information Terminal (S-IT) dengan menggunakan metode *use case points*. Hasil penelitian menunjukkan bahwa nilai *size* 96.767 dan nilai *effort* 1,451.505 person/hours dari nilai tersebut digunakan

untuk menentukan pengambilan keputusan dalam mengimplementasikan pengembangan aplikasi ditinjau berdasarkan waktu, biaya dan sumber daya manusia. Pada penelitian ini akan melakukan pengembangan metode estimasi usaha dari penelitian sebelumnya untuk mengetahui besaran *effort* dalam pengembangan aplikasi perangkat lunak dengan menggunakan metode Use Case Points dan COSMIC.

II. TINJAUAN PUSTAKA

A. Software

Software (Perangkat Lunak) merupakan suatu perintah (instruksi) yang digunakan untuk memproses sebuah informasi atau data [7]. Pengertian lain dari *Software* adalah program komputer yang digunakan sebagai sarana interaksi antara pengguna dan perangkat keras serta dapat mengubah instruksi dengan mudah. *Software* juga bisa dikatakan sebagai "interpreter" dari perintah-perintah yang dijalankan oleh pengguna komputer untuk diteruskan atau diproses oleh perangkat hardware [8]. Terdapat dua jenis *Software* yaitu *System Software* (melakukan tugas dasar yang dibutuhkan oleh seluruh pengguna komputer) dan *Application Software* (perangkat lunak yang dirancang khusus untuk kebutuhan tertentu contohnya Ms.Office) [9].

Pada *Sytem Software* memiliki tiga jenis dasar yaitu : [9]

1. Sistem operasi: berperan untuk mengaktifkan segala fitur yang terpasang pada komputer agar dapat saling berkomunikasi. Contoh: DOS, Linux, Windows, Mac OS dll.
2. Bahasa pemrograman: Program yang digunakan untuk menerjemahkan instruksi yang ditulis dalam bahasa pemrograman ke dalam bahasa mesin dengan aturan atau prosedur tertentu, sehingga dapat diterima oleh komputer. Contoh: Basic, Visual basic, visual C++ dll.
3. Program utilitas: berperan untuk membantu ataupun mengisi kekurangan dan kelemahan pada sistem operasi Contoh: Program Antivirus, Spyware, Pengompres File, DVD Burning.

Setiap Software akan dilakukan pengembangan dalam peningkatan kualitas penggunaan dan kebermanfaatan software. Dalam pengembangan software diperlukan analisa daftar kebutuhan yang diperlukan. Tingkat keberhasilan software dipengaruhi oleh dukungan eksekutif, keterlibatan pengguna, jumlah tenaga kerja, biaya, pengalaman manager proyek, tujuan bisnis yang jelas, infrastruktur perangkat lunak dan penggunaan metodologi pengembangan yang formal. Untuk mengukur tingkat keberhasilan software perlu dilakukan estimasi usaha sebelum melakukan pengembangan software [10].

B. Estimasi Usaha

Estimasi usaha pengembangan perangkat lunak merupakan suatu proses memprediksi dalam pengaturan sumber daya agar proyek dapat berjalan sesuai dengan apa yang telah ditargetkan yakni tepat waktu, sesuai dengan biaya dan terpenuhinya standart kualitas produk [5]. Estimasi usaha merupakan salah satu faktor terpenting dalam upaya pencegahan kegagalan suatu proyek. Ketepatan estimasi usaha merupakan penentu keberhasilan seorang manager proyek dalam mengatur dan menyelesaikan sebuah proyek. Demikian halnya dengan terjadinya kegagalan proyek diakibatkan dari estimasi usaha yang buruk [11].

Metode-metode pada estimasi usaha dikelompokkan menjadi 2 metode yaitu metode *Machine Learning* dan *Non-Machine Learning*. Contoh Metode *Machine Learning* adalah *case-based reasoning* (CBR), *k-nearest neighbor* (kNN), *neural networks* (NN), *support vector machine* (SVM), dan *linier regression* (LR). Kemudian untuk contoh metode *Non-Machine Learning* adalah *slim*, *expert judgement*, *cocomo*, *function points* (FP), dan *use case points* (UCP) [10].

C. Use Case Points (UCP)

Metode *Use Case Points* (UCP) merupakan salah satu metode yang digunakan untuk melakukan estimasi usaha dan tergolong pada metode *non-machine learning* [10]. UCP adalah salah satu metode estimasi ukuran perangkat lunak diperkenalkan oleh Gustav Karner pada 1993 yang memiliki kemampuan dalam memberikan estimasi effort berdasarkan jumlah dan kompleksitas *use case* yang dimiliki

software tersebut. Dan metode ini merupakan pengembangan dari metode *function points analysis* [12]. Terdapat empat tahapan dalam melakukan perhitungan estimasi biaya pengembangan perangkat lunak, diantaranya: menghitung unadjusted actor weight (UAW), menghitung unadjusted use case weight (UUCW), menghitung technical complexity factor (TCF) dan environment complexity factor (ECF) yang sebelumnya sudah diberikan bobot serta nilai yang telah ditentukan [4].

D. Common Software Measurement International Consortium (COSMIC)

COSMIC merupakan metode baru yang dikembangkan oleh sekelompok ahli pengukuran perangkat lunak dengan tujuan mengembangkan metode baru untuk mengukur ukuran *software*. Metode ini telah ditetapkan menjadi standart internasional ISO/IEC 19671, yang digunakan untuk menghitung fungsi-fungsi mulai dari memasukkan data, menyimpan, mengambil data, sampai dengan mengeluarkan data [13]. Dalam penggunaan COSMIC ini difokuskan pada kebutuhan fungsional *software* yang mana setelah mendapatkan daftar kebutuhan fungsional *software* akan diukur nilai fungsionalnya dan melakukan identifikasi kecacatannya [14].

III. METODOLOGI

A. Use Case Points (UCP)

Penelitian ini menggunakan metode penelitian UCP untuk melakukan perhitungan estimasi usaha pada aplikasi Cerdas Desaku. Langkah yang dilakukan dalam perhitungan adalah menghitung unadjusted actor weight (UAW), menghitung unadjusted use case weight (UUCW), menghitung technical complexity factor (TCF) dan environment complexity factor (ECF) yang sebelumnya sudah diberikan bobot serta nilai yang telah ditentukan [4].

Setelah mendapatkan masing-masing nilai pada empat tahapan, untuk mendapatkan nilai UCP dengan mengalikan nilai-nilai yang telah didapatkan sebelumnya. Kemudian, untuk mendapatkan nilai usaha didapatkan dari nilai UCP dikalikan terhadap effort rate dari penelitian sebelumnya [15][16][17]. Pada tahapan menghitung unadjusted actor weight (UAW) yaitu memberikan justifikasi score

berdasarkan hak akses masing-masing aktornya, dapat dilihat pada Table 1.

Tabel 1. Unadjusted Actor Weight

Kategori Aktor	Deskripsi	Bobot
Simple	Actor with API interaction through system	1
Average	Actor with standard communication protocols (eg TCP / IP, FTP, HTTP, database) through system	2
Complex	Human	3

Source: [4]

Demikian dengan unadjusted use case weight (UUCW). Didapatkan berdasarkan jumlah transaksi, penjelasan kategorisasi dapat dilihat pada Table 2.

Table 2. Unadjusted Use Case Weight

Kategori Use Case	Deskripsi	Bobot
Simple	Having transaction between 1-3	5
Average	Having transaction between 4-7	10
Complex	Having more than 7 transactions	15

Source: [4]

Terdapat 13 faktor kompleksitas teknis (TCF) dan 8 factor kompleksitas lingkungan (ECF). Rincian bobot untuk 13 TCF dapat dilihat pada Table 3.

Tabel 3. Technical Complexity Factor

No.	Deskripsi	Bobot
1	Distributed system	2.0
2	Response time / performance objectives	1.0
3	End-user efficiency	1.0
4	Internal processing complexity	1.0
5	Code reusability	1.0
6	Easy to install	0.5
7	Easy to use	0.5
8	Portability to other platforms	2.0
9	System maintenance	1.0
10	Concurrent / parallel processing	1.0
11	Security features	1.0
12	Access for third parties	1.0
13	End user training	1.0

Source:[4]

Terdapat 8 faktor lingkungan (ECF) dengan bobot masing-masing telah ditentukan dapat dilihat pada table 4. Rumus operasi matematika untuk TCF dan ECF dapat dilihat pada penelitian sebelumnya [18].

Tabel 4. Environment Complexity Factor

No.	Deskripsi	Bobot
1	Familiarity with the development process used	1.5
2	Application experience	0.5
3	Object-oriented experience of team	1.0

4	Lead analyst capability	0.5
5	Motivation of the team	1.0
6	Stability of requirements	2.0
7	Part-time employee	-1.0
8	Difficult programming language	-1.0

Source: [4]

B. Common Software Measurement International Consortium (COSMIC)

Penelitian ini menggunakan metode penelitian COSMIC untuk melakukan perhitungan estimasi usaha pada aplikasi Cerdas Desaku. Pada metode COSMIC setiap pergerakan tipe transaksi adalah satu CFP, berikut tipe transaksi yang digunakan untuk menghitung ukuran perangkat lunak [19][20] dapat dilihat pada Table 5.

Tabel 5. Tipe Transaksi COSMIC

Tipe Transaksi	Deskripsi
Entry (E)	Pergerakan data yang memindahkan grup data dari pengguna fungsional ke dalam proses fungsional
Read (R)	Pergerakan data yang memindahkan grup data dari penyimpanan persiten ke proses fungsional
Write (W)	Pergerakan data yang memindahkan grup data dari proses fungsional ke penyimpanan persiten
Exit (X)	Pergerakan data yang memindahkan data dari proses fungsional kepada pengguna.

Source: [16]

IV. HASIL DAN PEMBAHASAN

A. Perhitungan Usaha dengan Metode Use Case Points

Tahapan Perhitungan Usaha dengan Metode Use Case Points:

1. Mengklasifikasikan aktor untuk mendapatkan nilai *Unadjusted Actor Weight* (UAW)

Actor pada aplikasi cerdas desaku ini dibagi menjadi dua, dimana kedua actor dalam aplikasi cerdas desaku merupakan manusia yang berinteraksi dengan interface sistem, sehingga digolongkan ke dalam complex actor yang memiliki bobot tiga. Detail perhitungan *Unadjusted Actor Weight* (UAW) ditunjukkan pada tabel 6.

Tabel 6. Perhitungan Unadjusted Actor Weight

Kategori Aktor	Deskripsi	Bobot	Nilai
Simple	Actor with API interaction through	1	0

	system		
Average	Actor with standard communication protocols (eg TCP / IP, FTP, HTTP, database) through system	2	0
Complex	Human	3	6

2. Mengklasifikasikan use case diagram untuk menghasilkan nilai Unadjusted Use Case Weighting (UUCW).

Use case dibagi menjadi tiga kategori yaitu simple, average dan complex. Kategori tersebut berdasarkan jumlah transaksi yang ada di dalamnya untuk mendapatkan Unadjusted Use Case Weight (UUCW). Detail dari pembagian use case berdasarkan kompleksitasnya ditunjukkan pada tabel 2 dan hasil UUCW ditunjukkan pada tabel 7.

Tabel 7. Perhitungan Unadjusted Use Case Weight

Kategori Use Case	Deskripsi	Bobot	Nilai
Simple	Having transaction between 1-3	5	75
Average	Having transaction between 4-7	10	0
Complex	Having more than 7 transactions	15	0

3. Menghitung Technical Complexity Factor (TCF)

Proses selanjutnya menghitung Technical Complexity Factor (TCF), diberikan nilai pembobotan pada faktor kompleksitas dari TCF dengan rentang 0,1,2,3,4, dan 5, dimana semakin besar pengaruh dari faktor yang bersangkutan dalam pengerjaan proyek pengembangan perangkat lunak maka makin besar pula nilai estimasi pengaruhnya. Detail dari pembobotan TCF dapat dilihat pada tabel 3 dan hasil ditunjukkan pada tabel 8.

Tabel 8. Perhitungan Technical Complexity Factor

No.	Technical Complexity Factor	Bobot	Nilai (0-5)	Jumlah
T1	Distribution System	2,0	2	4
T2	Response time or throughput performance objectives	1,0	3	3
T3	End-user online efficiency	1,0	3	3
T4	Complex internal processing	1,0	3	3
T5	Reusability of code	1,0	1	1
T6	Easy to install	0,5	4	2

T7	Ease of use	0,5	2	1
T8	Portability	2,0	2	4
T9	Ease of change	1,0	3	3
T10	Concurrency	1,0	2	2
T11	Special security objectives included	1,0	2	2
T12	Direct access for third parties	1,0	3	3
T13	Special user training required	1,0	4	4
Technical Complexity Factor (TCF)				0,95

4. Menghitung Environment Complexity Factor (ECF)

Proses selanjutnya menghitung Environment Complexity Factor (ECF), diberikan nilai pembobotan pada faktor kompleksitas dari ECF dengan rentang 0,1,2,3,4, dan 5, dimana semakin besar pengaruh dari faktor yang bersangkutan dalam pengerjaan proyek pengembangan perangkat lunak maka makin besar pula nilai estimasi pengaruhnya. Detail dari pembobotan ECF dapat dilihat pada tabel 4 dan hasil ditunjukkan pada tabel 9.

Tabel 9. Perhitungan Environment Complexity Factor

No.	Environment Complexity Factor	Bobot	Nilai (0-5)	Jumlah
E1	Familiar with system development process being used	1,5	3	4,5
E2	Application Experience	0,5	3	1,5
E3	Object-Oriented experience	1,0	2	2
E4	Lead analysis capability	0,5	2	1
E5	Motivation	1,0	3	3
E6	Requirement stability	2,0	2	4
E7	Part time staff	-1,0	2	-2
E8	Difficult of programming language	-1,0	1	-1
Environment Complexity Factor (ECF)				13

5. Menghitung Use Case Points

Untuk menghitung UCP, harus dilakukan perhitungan UUCP dikalikan dengan TCF dan ECF. Sedangkan, nilai UAW dan UUCW dijumlahkan menjadi nilai Unadjusted Use Case Points (UUCP), hasil ditunjukkan pada tabel 10. Formula untuk menghitung dapat dilihat pada penelitian sebelumnya [12][6]

Tabel 10. Perhitungan Use Case Points

Unadjusted Use Case Points (UUCP)	81
Use Case Points	77,7195

Nilai UAW dan UUCW dijumlahkan menjadi nilai Unadjusted Use Case Points (UUCP), sehingga UUCP dari aplikasi cerdas desaku adalah 81. Hasil perhitungan UCP digunakan untuk menghitung usaha, dengan cara mengalikan nilai UCP terhadap effort rate. Diperoleh hasil estimasi usaha sebesar 637,300 man/hour.

B. Perhitungan Usaha dengan Metode COSMIC

Jumlah transaksi COSMIC, diukur dalam cfsu (COSMIC functional size unit), dihitung dengan menjumlahkan semua Entri (E), Exit (X), Read (R), dan Write (W) [21],[22]. Hasil perhitungan size COSMIC sebesar 206, digunakan untuk menghitung usaha dan diperoleh hasil estimasi usaha dengan menggunakan metode COSMIC sebesar 4.765,92 man/hour, ditunjukkan pada tabel 11.

Table 11. Perhitungan Ukuran COSMIC

Entry	Exit	Read	Write	Size	Total Effort
112	67	9	18	206	4.756,92

V. PENUTUP

Berdasarkan penelitian tentang perhitungan estimasi usaha pengembangan perangkat lunak aplikasi cerdas desaku untuk menunjang kemajuan desa dan meningkatkan kualitas layanan dengan menggunakan metode use case points dan COSMIC. Dapat disimpulkan bahwa estimasi perhitungan estimasi usaha aplikasi cerdas desaku, menggunakan metode *use case points* didapatkan hasil sebesar 637,300 man/hour. Sedangkan dengan menggunakan metode COSMIC didapatkan hasil sebesar 4.765,92 man/hour.

DAFTAR PUSTAKA

[1] A. . Rahadian, “Revitalisasi Birokrasi Melalui Transformasi Birokrasi Menuju E-Governance Pada Era Revolusi Industri 4.0,” *Pros. Semin. Stiami*, 2019.

[2] S. Suhesti, “Analisis Alokasi Sumber Daya Manusia, Waktu, Dan Biaya Menggunakan Metode Cocomo Ii

Dalam Manajemen Proyek Sistem Informasi Bpkb Online Pada Pt Rhys Auto Galery Surabaya,” Universitas Brawijaya, 2017.

- [3] R. Adhitama, “Effort estimation menggunakan metode use case point untuk pengembangan perangkat lunak,” *J. Informatics, Inf. Syst. Softw. Eng. Appl.*, vol. 1, no. 1, 2018.
- [4] W. H. N. Putra dan A. R. Perdanakusuma, “Estimasi Biaya Proyek Perangkat Lunak Menggunakan Use Case-Based Effort Estimation,” *J. TECNOSCENZA*, vol. 4, no. 2, hal. 283–300, 2020.
- [5] N. Rachmat dan S. Saparuddin, “Estimasi Ukuran Perangkat Lunak Menggunakan Function Point Analysis-Studi Kasus Aplikasi Pengujian dan Pembelajaran Berbasis Web,” in *Annual Research Seminar (ARS)*, 2017, vol. 3, no. 1, hal. 57–60.
- [6] D. Kurniadi, H. L. H. S. Warnars, F. L. Gaol, dan others, “Software size measurement of student information terminal with use case point,” in *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, 2017, hal. 164–169.
- [7] G. Y. Swara dan Y. Pebriadi, “Rekayasa Perangkat Lunak Pemesanan Tiket Bioskop Berbasis Web,” *J. TEKNOIF*, vol. 4, no. 2, hal. 27–39, 2016.
- [8] I. K. SIMATUPANG, “Pengenalan Perangkat Lunak Pada Komputer,” *Osf.Io*.
- [9] Shella, “Perangkat Lunak,” *1 September*, 2020. .
- [10] E. K. Adhitya, R. S. Wahono, dan H. Subagyo, “Komparasi Metode Machine Learning dan Metode Non Machine Learning untuk Estimasi Usaha Perangkat Lunak,” *IlmuKomputer. com J. Softw. Eng.*, vol. 1, no. 2, hal. 109–113, 2015.
- [11] E. Prayitno, “Penggunaan Metode Estimasi Use Case Points (UCP) Dalam

- Proyek Software Domain Bisnis,” *J. Inform.*, vol. 4, no. 2, 2017.
- [12] G. Karner, “Resource estimation for objectory projects,” *Object. Syst. SFAB*, hal. 1–9, 1993, [Daring]. Tersedia pada: http://si.lopesgazzani.com.br/docentes/marcio/gcm/p_Karner_ResourceEstimationForObjectoryProjects.pdf.
- [13] “COSMIC Sizing,” 2020. .
- [14] Prima Arifandi, “Fungsional Dengan Functional Size Measurement Berdasarkan Standar Cosmic Metode Refactoring (Studi Kasus : Wikibudaya) Requirement Using Functional Size Measurement Based On COSMIC ISO 19761 Standard And Refactoring Method (Case Study : Wikibudaya),” 2014.
- [15] R. S. Dewi, G. F. Prassida, A. P. Subriadi, dan others, “UCPabc as an integration model for software cost estimation,” in *2016 2nd International Conference on Science in Information Technology (ICSITech)*, 2016, hal. 187–192.
- [16] Sholiq, R. S. Dewi, dan A. P. Subriadi, “A comparative study of software development size estimation method: UCPabc vs function points,” *Procedia Comput. Sci.*, vol. 124, hal. 470–477, 2017.
- [17] A. P. Subriadi dan P. A. Ningrum, “Critical Review Of The Effort Rate Value In Use case Point Method For Estimating Software Development Effort,” vol. 59, no. 3, hal. 735–744, 2014.
- [18] K. Sangeetha dan P. P. Dalal, “Software Sizing with Use case point,” *Int. J. Innov. Sci. Eng. Technol.*, vol. 3, no. 8, hal. 146–150, 2016.
- [19] S. Di Martino, F. Ferrucci, C. Gravino, dan F. Sarro, “Web effort estimation: function point analysis vs. COSMIC,” *Inf. Softw. Technol.*, vol. 72, hal. 90–109, 2016.
- [20] C. J. Lin dan D. M. Yeh, “A Software Maintenance Project Size Estimation Tool Based on Cosmic Full Function Point,” in *Proceedings - 2016 International Computer Symposium, ICS 2016*, 2017, doi: 10.1109/ICS.2016.0115.
- [21] G. Kumar, P. K. Bhatia, dan others, “A detailed analysis of software cost estimation using COSMIC-FFP,” *PAK Publ. Gr. J. Rev Comput. Eng. Res*, vol. 2, no. 2, hal. 39–46, 2015
- [22] Andini, Dwi Yana Ayu. "PENERAPAN METODE LOGIKA FUZZY DALAM MENGETAHUI KECERDASAN MENGATASI MASALAH PELAMAR KERJA MELALUI TES ADVERSITY QUOTIENT (AQ)." *Aisyah Journal Of Informatics and Electrical Engineering* 1.1 (2019): 11-17.