

PREDIKSI BIAYA DAN DURASI PENGEMBANGAN PERANGKAT LUNAK MENGGUNAKAN FUZZY COCOMO

Yulis Rizal Fauzan ¹, Fahmi Auliya Tsani ², Devi Ayu Rachmawati ³

¹ Program Studi: Kecerdasan Buatan dan Robotik, Politeknik Artificial Intelligence, Yogyakarta, Indonesia

^{2,3} Program Studi: Rekayasa Keamanan Siber, Politeknik Artificial Intelligence, Yogyakarta, Indonesia
yulisrf@plai.ac.id

ABSTRAK

Perkembangan teknologi informasi yang semakin pesat mendorong meningkatnya kebutuhan pengembangan perangkat lunak yang harus diselesaikan tepat waktu dan sesuai anggaran. Permasalahan yang sering terjadi dalam proyek perangkat lunak adalah kurang optimalnya estimasi biaya dan durasi pengembangan. Oleh karena itu, penelitian ini bertujuan merancang metode estimasi biaya dan durasi pengembangan perangkat lunak menggunakan model *Fuzzy COCOMO*. Sistem yang dikembangkan mengotomatisasi penentuan model *COCOMO Basic* dan *COCOMO Intermediate* serta membandingkan kinerja metode inferensi *Fuzzy Mamdani*, *Sugeno*, dan *Tsukamoto* untuk membantu project manager dalam memperoleh estimasi yang lebih akurat. Penelitian menggunakan metode *design research* yang terdiri atas tahap konseptualisasi masalah, perancangan solusi, dan validasi empiris. Pengujian dilakukan menggunakan 63 data proyek pada dataset *COCOMO 81*, dengan metode evaluasi berupa perbandingan nilai estimasi terhadap data aktual berdasarkan tingkat akurasi model. Hasil penelitian menunjukkan bahwa metode *Sugeno* dan *Tsukamoto* menghasilkan akurasi rata-rata sebesar 50,0% pada *COCOMO Basic* dan 63,2% pada *COCOMO Intermediate*. Sementara itu, metode *Mamdani* memperoleh akurasi sebesar 50,5% pada *COCOMO Basic* dan 63,4% pada *COCOMO Intermediate*. Berdasarkan hasil tersebut, metode *Mamdani* memberikan kinerja terbaik dibandingkan metode lainnya, sedangkan *COCOMO Intermediate* terbukti lebih akurat dibandingkan *COCOMO Basic* dalam memprediksi biaya dan durasi pengembangan perangkat lunak sehingga berpotensi mengurangi risiko keterlambatan proyek dan pembengkakan biaya.

Kata kunci : *COCOMO, Logika Fuzzy, Estimasi Biaya*

1. PENDAHULUAN

Perkembangan teknologi serta perangkat lunak yang semakin hari semakin pesat menyebabkan banyaknya perusahaan (*Software Engineering*) jasa konsultan yang menyajikan layanan pembuatan dan pengembangan perangkat lunak sesuai dengan permintaan pengguna (*user*). Banyak sekali perusahaan yang bergerak di bidang informasi dan teknologi digital [1], dimana semakin banyak permintaan klien mengakibatkan meningkatnya jumlah proyek pembuatan perangkat lunak yang disetujui dan diselesaikan tepat waktu.

Permasalahan yang sering terjadi saat proses pengerjaan proyek perangkat lunak adalah tidak optimalnya estimasi durasi waktu pengerjaan dan perkiraan biaya proyek perangkat lunak (*Software*) yang dilakukan oleh *project manager* (PM). Pembagian *jobdesk* oleh project manager dilakukan dengan cara berbicara langsung dan itu kurang efektif karena anggota memiliki kemampuan yang berbeda terutama dalam pemahaman [2].

Tidak adanya representasi pengerjaan proyek dan *schedule* sebagai penunjang sehingga project manager tidak mengetahui pasti proses perkembangan proyek yang akibatnya adalah saat monitoring pengerjaan terhadap proyek sering terjadi kesalahan sehingga proyek banyak yang tidak selesai tepat waktu dan mengalami pembengkakan biaya pelaksanaan karena melebihi anggaran awal yang sudah disepakati [3].

Oleh sebab itu, tujuan penelitian ini adalah merancang sebuah metode estimasi waktu dan perkiraan biaya pada pengembangan perangkat lunak. Sistem ini digunakan untuk membantu dalam otomatisasi model *COCOMO* dan membandingkan hasil akurasi dengan *COCOMO* menggunakan algoritma *Fuzzy Sugeno* dan *Fuzzy Mamdani* dalam mengetahui dan mengukur estimasi biaya dan waktu (durasi) [4].

Proses identifikasi biaya dan waktu yang mudah dapat membantu mengurangi resiko kegagalan dalam pembangunan perangkat lunak sehingga akan memberikan keuntungan sekaligus meningkatkan citra bagi perusahaan [5].

Penelitian ini menggunakan metode *Fuzzy Mamdani*, *Sugeno*, dan *Tsukamoto* untuk membandingkan kinerja ketiga metode inferensi fuzzy dalam mengoptimasi model *COCOMO*. Ketiga metode dipilih karena merupakan metode inferensi yang umum digunakan dalam sistem logika fuzzy dan memiliki mekanisme inferensi yang berbeda, sehingga perbandingan di antara ketiganya diharapkan dapat memberikan gambaran mengenai metode yang menghasilkan estimasi biaya dan durasi pengembangan perangkat lunak yang paling akurat.

2. TINJAUAN PUSTAKA

Beberapa penelitian telah dilakukan untuk meningkatkan akurasi estimasi biaya dan durasi dalam pengembangan perangkat lunak. Malik [6]

mengombinasikan model COCOMO II dengan pendekatan *Neural Network* (NN) dan *Support Vector Machine* (SVM), yang menunjukkan bahwa integrasi metode cerdas dapat meningkatkan akurasi estimasi dibandingkan model konvensional. Sementara itu, Zakaria et al.

Mengoptimasi parameter COCOMO menggunakan *Particle Swarm Optimization* (PSO), yang terbukti mampu memperbaiki hasil estimasi meskipun masih bergantung pada pendekatan numerik [2].

Pendekatan lain yang berkembang adalah penggunaan logika *fuzzy* untuk menangani ketidakpastian dalam parameter estimasi. Gouda dan Mehta [7] mengembangkan model estimasi berbasis *Fuzzy C-Means* yang menunjukkan peningkatan fleksibilitas dan akurasi dalam proses prediksi biaya perangkat lunak.

Berdasarkan penelitian-penelitian tersebut, dapat disimpulkan bahwa meskipun berbagai metode telah digunakan untuk meningkatkan akurasi COCOMO, masih terdapat kebutuhan akan pendekatan yang mampu mengakomodasi ketidakpastian secara lebih intuitif dan mudah diimplementasikan. Oleh karena itu, penelitian ini mengusulkan penggunaan metode *Fuzzy COCOMO* dengan pendekatan *Fuzzy Mamdani* dan *Fuzzy Sugeno* untuk membandingkan tingkat akurasi dalam estimasi biaya dan durasi pengembangan perangkat lunak.

2.1. Constructive Cost Model (COCOMO)

Tujuan utama COCOMO adalah untuk membantu memahami konsekuensi biaya dari keputusan yang akan dibuat dalam menjalankan, mengembangkan, dan mendukung proyek perangkat lunak [5]. Selain menyediakan kemampuan estimasi biaya perangkat lunak, COCOMO juga menyediakan banyak materi yang menjelaskan dengan tepat berapa biaya yang diestimasi oleh model, dan mengapa model tersebut menghasilkan estimasi tersebut COCOMO memiliki tiga jenis model yaitu *Basic Model*, *Intermediate Model* dan *Detailed Model*.

2.2. COCOMO Basic

Parameter utama dalam analisis harga perangkat lunak disebut *effort* (E) yang dinyatakan dalam satuan

orang-bulan dan *time* (T) yang dinyatakan dalam satuan bulan. Estimasi harga perangkat lunak dapat diketahui dengan mengalikan nilai *effort* dengan gaji programmer di suatu kondisi tertentu dan dikalikan dengan nilai *time* atau waktu pengerjaan proyek tersebut [6].

Ukuran produk perangkat lunak dinyatakan sebagai KLOC (*Kilo Line Of Codes*) yaitu jumlah baris *source code* dalam ribuan. Metode COCOMO memiliki persamaan sebagai berikut:

$$\begin{aligned} effort(E) &= a(KLOC)^b \\ time(T) &= c(E)^d \end{aligned} \quad (1)$$

$$Person\ Required = E / T$$

Keterangan :

$KLOC$ = Jumlah baris *source code* dalam ribuan

E = Usaha pengembangan perangkat lunak (orang-bulan)

T = Waktu Pengembangan perangkat lunak (bulan)

a, b, c, d = konstanta jenis perangkat lunak

Konstanta jenis perangkat lunak dibagi menjadi tiga jenis yaitu *Organic*, *Semi-Detached* dan *Embedded*. Proyek perangkat lunak *Organic* berukuran kecil, memerlukan tim kecil, permasalahan dipahami dengan baik, dan anggota tim berpengalaman dalam mengembangkan proyek serupa. Proyek *Semi-Detached* berukuran sedang, dengan karakteristik seperti ukuran tim, pengalaman, dan pengalaman tim dalam mengembangkan proyek terletak di antara *Organic* dan *Embedded*. Proyek perangkat lunak *Embedded* berukuran besar, memiliki tingkat kompleksitas yang tinggi, terkadang merupakan pengembangan perangkat lunak jenis baru dan dipadukan dengan perangkat keras yang kompleks [8]. Konstanta dari masing-masing jenis perangkat lunak menurut definisi Boehm dinyatakan dalam table berikut:

Tabel 1. Konstantan jenis perangkat lunak

Jenis COCOMO	a	b	c	d
<i>Organic</i>	2,4	1,05	2,5	0,38
<i>Semi-Detached</i>	3	1,12	2,5	0,35
<i>Embedded</i>	3,6	1,2	2,5	0,32

Penentuan jenis perangkat lunak dilakukan dengan mempertimbangkan beberapa aspek. Metode COCOMO membagi aspek-aspek tersebut dalam table berikut [9]:

Tabel 2. Mode pengembangan perangkat lunak COCOMO

Fitur	Mode		
	Organic	Semi Detached	Embedded
Pemahaman organisasi tentang tujuan produk	menyeluruh	Besar	Umum
Pengalaman dalam bekerja dengan sistem perangkat lunak terkait	Luas	Besar	Sedang
Kebutuhan akan kesesuaian perangkat lunak dengan persyaratan yang telah ditetapkan sebelumnya	Dasar	Besar	Full
Kebutuhan akan kesesuaian perangkat lunak dengan spesifikasi antarmuka eksternal	Dasar	Besar	Full
Pengembangan bersamaan perangkat keras baru dan prosedur operasional terkait	Beberapa	Sedang	Luas
Kebutuhan akan arsitektur pemrosesan data yang inovatif, algoritma	Minimal Rendah	Rentang Sedang	Cukup Tinggi
Premis pada penyelesaian awal rentang ukuran produk	<50 KLOC	<300 KLOC	>300 KLOC

2.3. COCOMO Intermediate

COCOMO *Intermediate* adalah kelanjutan dari COCOMO *Basic* dengan menambahkan variabel *cost driver* pada perhitungan untuk mendapat hasil yang lebih baik. Parameter *cost driver* disebut *effort multipliers* dengan skala *very low* hingga *extra high* [9].

Tabel 3. kategori *cost driver* COCOMO

Kategori	Cost Driver
Product Attributes	RELY—Required software reliability
	DATA—Database size
	CPLX—Product complexity constraint
Computer Attributes	TIME—Execution time constraint
	STOR—Main storage constraint
	VIRT—Virtual machine volatility*
	TURN—Computer turnaround time
Personnel Attributes	ACAP—Analyst capability
	AEXP—Applications experience
	PCAP—Programmer capability
	VEXP—Virtual machine experience*
	LEXP—Programming language experience
Project Attributes	MODP—Use of modern programming practices
	TOOL—Use of software tools
	SCED—Required development schedule

Tabel 4. Nilai *effort multipliers* COCOMO

Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
RELY	0.75	0.88	1	1.15	1.4	-
DATA	-	0.94	1	1.08	1.16	-
CPLX	0.7	0.85	1	1.15	1.3	1.65
TIME	-	-	1	1.11	1.3	1.66
STOR	-	-	1	1.06	1.21	1.56
VIRT	-	0.87	1	1.15	1.3	-
TURN	-	0.87	1	1.07	1.15	-
ACAP	1.46	1.19	1	0.86	0.71	-
AEXP	1.29	1.13	1	0.91	0.82	-
PCAP	1.42	1.17	1	0.86	0.7	-
VEXP	1.21	1.1	1	0.9	-	-
LEXP	1.14	1.07	1	0.95	-	-
MODP	1.24	1.1	1	0.91	0.82	-
TOOL	1.24	1.1	1	0.91	0.83	-
SCED	1.23	1.08	1	1.04	1.1	-

Persamaan untuk COCOMO *Intermediate* adalah sebagai berikut:

$$\begin{aligned}
 effort(E) &= a(KLOC)^b * EM \\
 time(T) &= c(E)^d \\
 Person\ Required &= E / T
 \end{aligned}
 \quad (2)$$

Keterangan :

EM = Jumlah perkalian *effort multipliers*

Berdasarkan karakteristik kedua model tersebut, penelitian ini menggunakan COCOMO *Basic* dan COCOMO *Intermediate* sebagai objek pengujian. COCOMO *Basic* dipilih karena memberikan estimasi dasar berdasarkan ukuran perangkat lunak, sedangkan

COCOMO *Intermediate* dipilih karena mempertimbangkan faktor *cost driver* yang dapat meningkatkan representasi kondisi proyek. Dengan menggunakan kedua model tersebut, penelitian ini dapat membandingkan pengaruh optimasi logika *fuzzy* terhadap model estimasi yang sederhana maupun yang lebih kompleks.

2.4. Fuzzifikasi Logika Fuzzy

Fuzzifikasi adalah proses yang dilakukan untuk mengubah variabel nyata (*crisp*) menjadi bentuk variabel *fuzzy* [10]. Hal ini ditujukan agar masukan kontroler *fuzzy* bisa dipetakan menuju jenis yang sesuai dengan himpunan *fuzzy*. Penentuan jumlah *fuzzy* subset dan pemilihan bentuk fungsi keanggotaan (*membership function*) untuk input dan output *fuzzy* sangat penting dalam proses fuzzifikasi [11].

Fungsi keanggotaan dapat berbentuk segitiga, trapesium, maupun bentuk lainnya, tergantung pada kebutuhan system [12].

Pemilihan bentuk dan jumlah fungsi keanggotaan sangat berpengaruh terhadap hasil akhir sistem *fuzzy*.

2.5. Inferensi Logika Fuzzy

Inferensi *fuzzy* merupakan proses penalaran yang digunakan untuk memperoleh output berdasarkan aturan-aturan (*rule base*) yang telah ditentukan. Aturan *fuzzy* biasanya berbentuk pernyataan *IF-THEN* yang menghubungkan variabel input dan output [13].

Terdapat beberapa metode inferensi *fuzzy* yang umum digunakan, di antaranya adalah metode *Mamdani* dan *Sugeno*. Metode *Mamdani* menggunakan pendekatan berbasis himpunan *fuzzy* untuk menghasilkan output, sedangkan metode *Sugeno* menggunakan fungsi output berupa konstanta atau persamaan linear sehingga lebih mudah dalam proses komputasi [10].

2.6. Defuzzifikasi Logika Fuzzy

Proses defuzzifikasi digunakan untuk mendapat nilai rekomendasi. Proses defuzzifikasi untuk metode *Sugeno*, akan dilakukan menggunakan metode rata-rata berbobot (*Weight Average*). Untuk defuzzifikasi variabel Harga dan Waktu adalah persamaan sebagai berikut.

$$WA = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 + \dots + \alpha_9 z_9}{\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_9} \quad (9)$$

Keterangan :

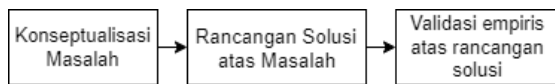
WA = Nilai *Weight Average*

α = Nilai predikat aturan

Z = Indeks nilai output (konstanta)

3. METODE PENELITIAN

Pada penelitian ini menggunakan metode penelitian perancangan (*design research*). Terdapat tiga fase dalam penelitian perancangan sistem informasi, yaitu konseptualitas masalah, rancangan solusi, dan validasi empiris.

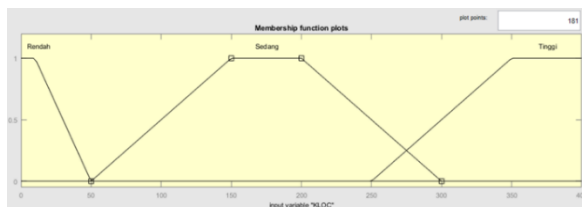
Gambar 3. Alur *design research*

Tahap pertama *design research* adalah konseptualisasi masalah, yaitu proses untuk mengidentifikasi area fokus penelitian, merinci batasan-batasan yang ada, serta membentuk dasar bagi perancangan solusi yang efektif. Tahap kedua adalah rancangan solusi atas masalah, yang mencakup perumusan pendekatan atau strategi yang akan digunakan untuk memecahkan masalah yang telah diidentifikasi.

Pada tahap ini dilakukan pengembangan konsep solusi, pembuatan prototipe, serta penyusunan rencana implementasi. Tahap terakhir adalah validasi empiris atas rancangan solusi, yang bertujuan untuk memastikan bahwa solusi yang diusulkan dapat berfungsi secara efektif dalam praktik, sesuai dengan tujuan penelitian, serta mampu memberikan hasil yang dapat diandalkan terhadap permasalahan yang telah diidentifikasi [13].

3.1. Perancangan Sistem Logika Fuzzy

Pada penelitian ini terdapat dua variabel input yaitu KLOC dan Kompleksitas. Variabel Jumlah KLOC dibuat 3 buah fungsi keanggotaan yaitu fungsi keanggotaan Rendah menggunakan kurva trapesium bahu kiri, fungsi keanggotaan Sedang menggunakan kurva trapesium dan fungsi keanggotaan Tinggi menggunakan kurva trapesium bahu kanan [14].

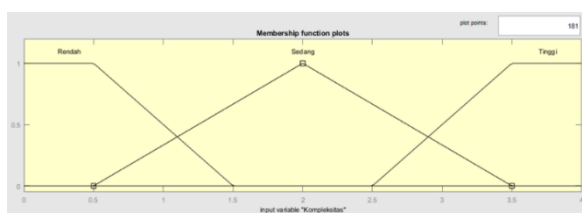


Gambar 1. Fungsi keanggotaan variabel KLOC

$$\mu_{\text{Rendah}}(x) = \begin{cases} 1 & x \leq 10 \\ \frac{40 - (x - 10)}{40} & 10 < x \leq 50 \\ 0 & x > 50 \end{cases} \quad (3)$$

$$\mu_{\text{Sedang}}(x) = \begin{cases} 0 & x \leq 50 \\ \frac{x - 50}{100} & 50 < x \leq 150 \\ 1 & 150 < x \leq 200 \\ \frac{100 - (x - 200)}{100} & 200 < x \leq 300 \\ 0 & x > 300 \end{cases} \quad (4)$$

$$\mu_{\text{Tinggi}}(x) = \begin{cases} 0 & x \leq 250 \\ \frac{x - 250}{100} & 250 < x \leq 350 \\ 1 & x \geq 90 \end{cases} \quad (5)$$



Gambar 2. Fungsi keanggotaan variabel Kompleksitas

Variabel Jumlah Kompleksitas dibuat 3 buah fungsi keanggotaan yaitu fungsi keanggotaan Rendah menggunakan kurva trapesium bahu kiri, fungsi keanggotaan Sedang menggunakan kurva segitiga dan fungsi keanggotaan Tinggi menggunakan kurva trapesium bahu kanan [15].

$$\mu_{\text{Rendah}}(x) = \begin{cases} 1 & x \leq 0,5 \\ \frac{1 - (x - 0,5)}{1} & 0,5 < x \leq 1,5 \\ 0 & x > 1,5 \end{cases} \quad (6)$$

$$\mu_{\text{Sedang}}(x) = \begin{cases} 0 & x \leq 0,5 \\ \frac{x - 0,5}{1,5} & 0,5 < x \leq 2 \\ \frac{1,5 - (x - 2)}{1,5} & 2 < x \leq 3,5 \\ 0 & x > 3,5 \end{cases} \quad (7)$$

$$\mu_{\text{Tinggi}}(x) = \begin{cases} 0 & x \leq 2 \\ \frac{x - 2,5}{1} & 2,5 < x \leq 3,5 \\ 1 & x \geq 3,5 \end{cases} \quad (8)$$

Proses inferensi pada penelitian ini menggunakan metode *Sugeno Orde-0* dan *Tsukamoto* dimana fungsi keanggotaan keluaran adalah sebuah konstanta tegas sebagai kusekuen. Proses fuzifikasi ini akan mencari nilai fungsi keanggotaan dari variabel output mode, yaitu konstanta jenis perangkat lunak model COCOMO. Fungsi keanggotaan variabel mode adalah Organik dengan konstanta 0, *Semi-Detached* dengan konstanta 1, *Ebedded* dengan konstanta 2.

Berdasarkan fungsi keanggotaan input dan output yang telah ditentukan, maka langkah selanjutnya adalah membuat aturan-aturan (*rule base*) sebagai proses penalaran *Fuzzy* [12]. Terdapat 9 aturan *fuzzy* sebagai berikut:

Tabel 5. Aturan *Fuzzy*

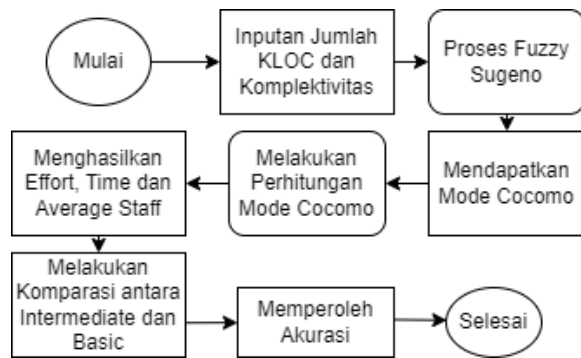
No.	KLOC	Kompleksitas	Mode
1	Rendah	Rendah	<i>Organic</i>
2	Rendah	Sedang	<i>Organic</i>
3	Rendah	Tinggi	<i>Embedded</i>
4	Sedang	Rendah	<i>Organic</i>
5	Sedang	Sedang	<i>Semi-Detached</i>
6	Sedang	Tinggi	<i>Embedded</i>
7	Tinggi	Rendah	<i>Semi-Detached</i>
8	Tinggi	Sedang	<i>Semi-Detached</i>
9	Tinggi	Tinggi	<i>Embedded</i>

Proses inferensi untuk metode *Mamdani* menggunakan metode *Mamdani Min-Max*. Proses ini melibatkan penerapan aturan *fuzzy* yang telah ditetapkan untuk menentukan tingkat keanggotaan output *fuzzy*. Setelah menerapkan semua aturan *fuzzy*, hasil *fuzzy* untuk setiap kesimpulan (mode COCOMO) dikombinasikan. Operasi *max* digunakan untuk mengambil nilai tertinggi dari setiap kesimpulan *fuzzy* yang sama.

Prose defuzzifikasi yang digunakan dalam Metode *Mamdani* adalah metode *Maximum Membership Principle* atau dikenal juga sebagai metode *Winner-Takes-All*. Metode ini memilih nilai crisp output berdasarkan nilai keanggotaan tertinggi dari hasil *fuzzy* yang telah dihitung. Dalam konteks ini, mode COCOMO yang memiliki nilai *fuzzy* tertinggi dipilih sebagai hasil akhir.

4. HASIL DAN PEMBAHASAN

4.1. Diagram Alir Sistem



Gambar 4. Diagram Alir Sistem

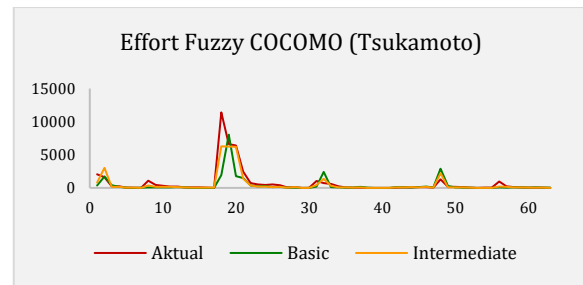
Diagram alir sistem diatas menjelaskan bahwa ketika menginputkan jumlah *KLOC* dan Kompleksitas, maka sistem akan mengotomatisasi menggunakan *fuzzy Sugeno* dalam menentukan mode *COCOMO*. Setelah itu mode *COCOMO* akan menghitung *effort*, *time* dan *Average Staff* yang dibutuhkan. Kemudian melakukan komparasi dan didapatkan akurasi mana yang lebih baik antara 2 mode *COCOMO*.

4.2. Implementasi/Pengujian

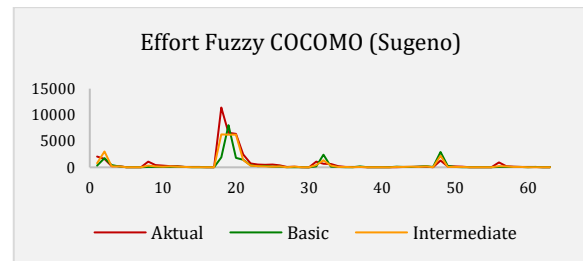
Pada penelitian ini sistem akan diuji menggunakan dataset *COCOMO 81*. Dataset ini merupakan dataset resmi yang digunakan untuk menciptakan pemodelan *COCOMO*. Dataset ini berisi 63 data dari proyek perusahaan pengembang perangkat lunak bidang kedirgantaraan [7]. Kemudian dataset diuji menggunakan model *Fuzzy COCOMO Basic*, *Intermediate* dan *Detailed* untuk melihat tingkat akurasi masing-masing model. Hasil pengujian dari masing-masing model *Fuzzy COCOMO* adalah sebagai berikut:

Tabel 6. Akurasi *Fuzzy COCOMO basic & intermediate*

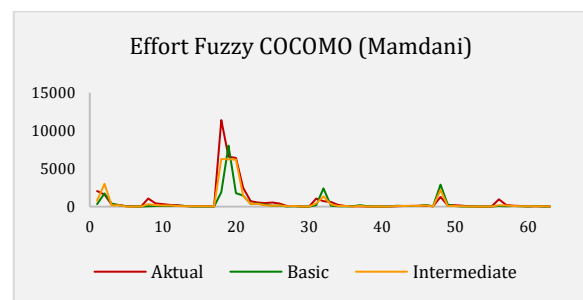
Metode	Akurasi		
	<i>Tsukamoto</i>	<i>Sugeno</i>	<i>Mamdani</i>
<i>Fuzzy COCOMO Basic</i>	50,00%	50,00%	50,50%
<i>Fuzzy COCOMO Intermediate</i>	63,20%	63,20%	63,60%



Gambar 5. Grafik perbandingan hasil *Effort Actual*, *Basic* dan *Intermediate* metode *Tsukamoto*



Gambar 6. Grafik perbandingan hasil *Effort Actual*, *Basic* dan *Intermediate* metode *Sugeno*



Gambar 7. Grafik perbandingan hasil *Effort Actual*, *Basic* dan *Intermediate* metode *Mamdani*

4.3. Pembahasan

Dari hasil pengujian dapat dilihat bahwa model *Fuzzy COCOMO* metode *Tsukamoto* dan *Sugeno* memiliki hasil yang sama, pada mode *COCOMO Basic* untuk seluruh dataset memiliki rata-rata tingkat akurasi sebesar 50,0% dan *intermediate* sebesar 63,2%. Sedangkan untuk model *Fuzzy COCOMO* metode *Mamdani* pada mode *COCOMO Basic* untuk seluruh dataset memiliki rata-rata tingkat akurasi sebesar 50,5% dan *intermediate* sebesar 63,4%.

Untuk melihat gambaran kinerja model *Fuzzy COCOMO* untuk masing-masing metode diambil 20 sampel data testing pertama sebagai berikut :

Tabel 7. Hasil kinerja model *Fuzzy COCOMO*

No	AKTUAL		TSUKAMOTO		SUGENO		MAMDANI	
			BASIC	INTER	BASIC	INTER	BASIC	INTER
	KLOC	Effort	Effort	Effort	Effort	Effort	Effort	Effort
1	113	2040	343.5	786.0	343.5	786.0	343.5	786.0
2	293	1600	1737.9	3004.5	1737.9	3004.5	1737.9	3004.5
3	132	243	404.4	149.0	404.4	149.0	404.4	149.0
4	60	240	176.7	176.4	176.7	176.4	176.7	176.4
5	16	33	44.1	28.9	44.1	28.9	44.1	28.9
6	4	43	10.3	19.2	10.3	19.2	10.3	19.2

No	AKTUAL		TSUKAMOTO		SUGENO		MAMDANI	
			BASIC	INTER	BASIC	INTER	BASIC	INTER
	KLOC	Effort	Effort	Effort	Effort	Effort	Effort	Effort
7	6.9	8	18.2	7.3	18.2	7.3	18.2	7.3
8	22	1075	61.6	339.5	61.6	339.5	61.6	339.5
9	30	423	85.3	171.9	85.3	171.9	85.3	171.9
10	29	321	82.4	142.5	82.4	142.5	82.4	142.5
11	32	218	91.3	158.0	91.3	158.0	91.3	158.0
12	37	201	106.4	99.6	106.4	99.6	106.4	99.6
13	25	79	70.5	60.1	70.5	60.1	70.5	60.1
14	3	60	7.6	37.6	7.6	37.6	7.6	37.6
15	3.9	61	10.0	30.5	10.0	30.5	10.0	30.5
16	6.1	40	16.0	38.1	16.0	38.1	16.0	38.1
17	3.6	9	9.2	17.9	9.2	17.9	9.2	17.9
18	320	11400	1918.2	6274.7	1918.2	6274.7	1918.2	6274.7
19	1150	6600	8037.2	6240.7	8037.2	6240.7	8037.2	6240.7
20	299	6400	1777.8	6200.7	1777.8	6200.7	1777.8	6200.7

Berdasarkan Tabel 7 terlihat bahwa hasil estimasi menggunakan COCOMO *Intermediate* cenderung lebih mendekati nilai aktual dibandingkan COCOMO *Basic* pada sebagian besar data uji. Hal ini menunjukkan bahwa penambahan parameter *cost driver* pada COCOMO *Intermediate* mampu memperbaiki estimasi *effort* dibandingkan model yang hanya mempertimbangkan ukuran KLOC. Pada proyek berukuran kecil hingga sedang, perbedaan estimasi relatif tidak terlalu besar, sedangkan pada proyek berukuran besar seperti data ke-18 sampai data ke-20 terlihat bahwa estimasi *Intermediate* lebih mampu mengikuti pola data aktual dibandingkan *Basic*.

Selain itu, ketiga metode inferensi fuzzy menghasilkan nilai estimasi yang relatif serupa karena menggunakan basis aturan (*rule base*) yang sama. Perbedaan hanya muncul pada mekanisme inferensi dan proses defuzzifikasi. Metode Mamdani menghasilkan akurasi sedikit lebih tinggi dibandingkan Sugeno dan Tsukamoto, yang menunjukkan bahwa pendekatan inferensi berbasis himpunan fuzzy memberikan representasi hubungan antarvariabel yang lebih baik pada data COCOMO.

Dari hasil estimasi pada Tabel 7 juga terlihat bahwa masih terdapat selisih (*error*) antara nilai *effort* aktual dengan nilai hasil prediksi pada seluruh metode. Besarnya selisih tersebut menunjukkan bahwa model estimasi belum mampu merepresentasikan seluruh karakteristik proyek perangkat lunak secara sempurna. Namun demikian, pada sebagian besar data uji, nilai estimasi COCOMO *Intermediate* memiliki selisih yang lebih kecil terhadap data aktual dibandingkan COCOMO *Basic*. Hal ini menunjukkan bahwa penambahan parameter *cost driver* pada COCOMO *Intermediate* mampu mengurangi *error* estimasi sehingga menghasilkan prediksi yang lebih mendekati kondisi sebenarnya.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, metode COCOMO mampu memberikan gambaran awal mengenai estimasi biaya dan durasi pengembangan perangkat

lunak sehingga dapat membantu project manager dalam mengurangi risiko keterlambatan proyek maupun pembengkakan biaya. Penerapan logika *fuzzy* juga berhasil mengotomatisasi proses penentuan konstanta jenis perangkat lunak berdasarkan nilai KLOC dan kompleksitas proyek, sehingga pengguna tidak perlu menentukan mode COCOMO secara manual. Hasil pengujian menggunakan 63 data pada dataset COCOMO 81 menunjukkan bahwa metode *Fuzzy Tsukamoto* dan *Sugeno* menghasilkan rata-rata akurasi sebesar 50,0% pada COCOMO *Basic* dan 63,2% pada COCOMO *Intermediate*, sedangkan metode *Mamdani* memperoleh akurasi sebesar 50,5% pada COCOMO *Basic* dan 63,4% pada COCOMO *Intermediate*. Hasil tersebut menunjukkan bahwa metode *Mamdani* memberikan kinerja terbaik dibandingkan *Tsukamoto* dan *Sugeno*, sementara COCOMO *Intermediate* memberikan estimasi yang lebih akurat dibandingkan COCOMO *Basic* karena mempertimbangkan faktor *cost driver*. Meskipun demikian, penelitian ini masih memiliki keterbatasan karena pengujian hanya dilakukan menggunakan satu dataset, yaitu COCOMO 81, serta evaluasi hanya difokuskan pada perbandingan akurasi ketiga metode inferensi *fuzzy*. Oleh karena itu, penelitian selanjutnya disarankan menggunakan dataset yang lebih beragam, menerapkan metode optimasi lain untuk meningkatkan performa model *Fuzzy COCOMO*, serta menambahkan metrik evaluasi yang lebih komprehensif agar hasil estimasi dapat dianalisis secara lebih mendalam.

DAFTAR PUSTAKA

- [1] Y. Syafitri, R. Astika, and L. S. E. E. Rahayu, "PENGEMBANGAN APLIKASI PELELANGAN MENGGUNAKAN FRAMEWORK CODEIGNITER BERBASIS WEB," no. 2, 2022.
- [2] N. A. Zakaria, A. R. Ismail, N. Z. Abidin, N. H. M. Khalid, and A. Y. Ali, "Optimization of COCOMO model using particle swarm optimization," *Int. J. Adv. Intell. Informatics*, vol. 7, no. 2, pp. 177–187, 2021, doi:

- 10.26555/ijain.v7i2.583.
- [3] Kiki Kusumawati, Nurul Chafid, and Sri Ariyani, "Analisis Dan Perancangan Manajemen Proyek Perangkat Lunak Studi Kasus : Pt. Ekiosku," *Prosiding*, vol. 2, no., pp. 117–129, 2022, doi: 10.59134/prosiding.v2i-.120.
 - [4] A. M. Nuris, A. Maharani, and R. N. Rachmadita, "Analisis Risiko Proyek Pengembangan Perangkat Lunak Menggunakan Kerangka Kerja ISO 31000," *J. METRIS*, vol. 22, no. 02, pp. 73–81, 2022, doi: 10.25170/metris.v22i02.2800.
 - [5] L. Damayanti, "Proyek Rekayasa Perangkat Lunak Sistem Permintaan Perubahan Program Pada Pt Prima," *Akselerator J. Sains Terap. Dan Teknol.*, vol. 2, no. 2, pp. 84–99, 2021.
 - [6] S. Malik, "Software Quality Evaluation by COCOMO II With NN and SVM," *Math. Stat. Eng. Appl.*, vol. 71, Jan. 2022, doi: 10.17762/msea.v71i1.34.
 - [7] S. K. Gouda and A. K. Mehta, "Software cost estimation model based on fuzzy C-means and improved self adaptive differential evolution algorithm," *Int. J. Inf. Technol.*, vol. 14, no. 4, pp. 2171–2182, 2022, doi: 10.1007/s41870-022-00882-4.
 - [8] O. Fatonah and Y. Afrizal, "MODEL ESTIMASI BIAYA PERANGKAT LUNAK MENGGUNAKAN COCOMO II (STUDI KASUS PT. X)," *J. Tata Kelola dan Kerangka Kerja Teknol. Inf.*, vol. 2, pp. 54–61, 2016, doi: 10.34010/jtk3ti.v2i1.302.
 - [9] S. Balakrishna, "Is COCOMO and Putnam relevant to e-Government? Software development efforts estimation in e-Government in the Indian state of Andhra Pradesh," *Digit. Policy, Regul. Gov.*, vol. 25, no. 3, pp. 267–287, 2023, doi: 10.1108/DPRG-11-2021-0147.
 - [10] T. I. Nasrul, Amri, and M. Sayuti, "Application of Fuzzy Mamdani Method to Predict the Number of Blood Bags Based on Demand and Supply Data Using Matlab," *Int. J. Eng. Sci. Inf. Technol.*, vol. 4, no. 4, pp. 29–37, 2024, doi: 10.52088/ijesty.v4i4.567.
 - [11] M. N. Aslam, S. Surorejo, E. Unggul, and S. Utami, "Penerapan Metode Fuzzy Mamdani dalam Prediksi Cuaca di Tegal," vol. 4, no. 3, pp. 2574–2582, 2025.
 - [12] D. K. Y. Hutapea, H. Sachi, W. Kusmanto, D. Herianto, N. Fernanto, and K. O. Bachri, "Simulasi Pengendalian Kecepatan Putar Motor DC Menggunakan Metode Logika Fuzzy," *Metris J. Sains dan Teknol.*, vol. 24, no. 02, pp. 59–66, 2023, doi: 10.25170/metris.v24i02.4725.
 - [13] K. Gericke, C. Eckert, and M. Stacey, "Elements of a design method – a basis for describing and evaluating design methods," *Des. Sci.*, vol. 8, p. e29, 2022, doi: 10.1017/dsj.2022.23.
 - [14] S. Sholiq, R. Sarno, \Endang Siti\ Astuti, and \Muhammad Ainul\ Yaqin, "Implementation of COSMIC Function Points (CFP) as Primary Input to COCOMO II: Study of Conversion to Line of Code Using Regression and Support Vector Regression Models," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 5, pp. 92–103, 2023, doi: 10.22266/ijies2023.1031.09.
 - [15] E. N. Udo, O. U. Obot, and P. G. Obike, "A Comparative Analysis Of Adaptive Neuro-Fuzzy Inference System Back Propagation And Hybrid Learning For Software Development Effort Evaluation," *Quest Journals J. Softw. Eng. Simul.*, vol. 8, no. 7, pp. 2321–3809, 2022, doi: 10.35629/9795-08072338.