

## Meningkatkan Akurasi Perkiraan Waktu Proyek Perangkat Lunak Dalam COCOMO II Dengan Mengubah Nilai Parameter

Rahmi Rizkiana Putri<sup>1</sup>, Weny Mistarika Rahmawati<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya

Email : [rahmirizkianaputri@gmail.com](mailto:rahmirizkianaputri@gmail.com)

**Abstract** Good management of software projects can be obtained through accurate time estimates. When estimating less accurate time it will affect the lack of effective management of the software project and the entire process during project development. Barry Boehm, an inventor of COCOMO, has developed the COCOMO I cost driver that has an effect on the accuracy of the estimated time results. But if you only use the COCOMO II cost driver, it is still far from the accuracy of the desired results. Therefore it is necessary to change the values of parameters C and D for estimated time. Changes in parameter values are done by decreasing the initial gradation by 0.1 so that the approximate results become more optimal and close to the original values. Based on the implementation of the proposed method, the results show that the error decreases to 0.08% when compared to using only the COCOMO II and COCOMO II cost drivers without changing parameter values. So that the accuracy of the estimated project time can increase.

**Keywords:** Time, cocomo II, software project, accuracy.

**Abstrak** Manajemen yang baik pada proyek perangkat lunak bisa didapatkan salah satunya melalui perkiraan waktu yang akurat. Ketika dalam memperkirakan waktu kurang akurat maka hal tersebut akan berpengaruh pada kurang efektifnya manajemen pada proyek perangkat lunak serta seluruh proses saat pengembangan proyek. Barry Boehm seorang penemu dari COCOMO telah mengembangkan *cost driver* COCOMO I yang berpengaruh terhadap keakuratan hasil perkiraan waktu. Tetapi jika hanya menggunakan *cost driver* COCOMO II hal tersebut masih jauh dari keakuratan hasil yang diinginkan. Oleh sebab itu diperlukan adanya perubahan nilai parameter C dan D untuk perkiraan waktu. Perubahan nilai parameter dilakukan dengan menurunkan gradual awal sebanyak 0,1 agar hasil perkiraan menjadi lebih optimal dan mendekati nilai aslinya. Berdasarkan implementasi dari metode yang diusulkan maka didapatkan hasil bahwa kesalahan berkurang menjadi 0,08% bila dibandingkan hanya menggunakan *cost driver* COCOMO I dan COCOMO II tanpa perubahan nilai parameter. Sehingga keakuratan dari waktu proyek yang diperkirakan dapat meningkat.

Kata kunci: Waktu, Cocomo ii, Proyek Perangkat Lunak, Akurasi.

### 1. Pendahuluan

Perkiraan waktu proyek perangkat lunak adalah satu dari masalah utama proyek perangkat lunak, dan sangat penting dalam manajemen dan pengembangan proyek untuk membantu manajemen perusahaan menjadi lebih baik ketika dikembangkan. Dalam proyek terdapat banyak fakta tentang aktivitas dari proyek perangkat lunak. Seperti pernyataan Boehm, setidaknya yang digunakan untuk memenuhi kebutuhan pada awal adalah 20% dari waktu proyek, 8% dari total sumber daya. Hal ini menunjukkan bahwa kebanyakan proyek berhasil jika menggunakan 7% dari 15% total sumber daya proyek perangkat lunak yang dibutuhkan. Atribut pertanyaan proyek adalah deskripsi dari karakteristik hubungan dengan proyek perangkat lunak. Ada 7 atribut, yaitu pengukuran, kompleksitas, kebutuhan, kategori, level keamanan, waktu dan biaya. Tiap atribut memiliki jumlah yang telah menjadi ketentuan. Contohnya ukuran proyek memiliki rentang nilai lebih dari 4000 yang berarti proyek sangat

besar, antara 1000 dan 4000 yang berarti proyek besar, antara 500 dan 1000 yang berarti proyek menengah, antara 100 dan 500 yang berarti proyek kecil, sedangkan kurang dari 100 yang berarti proyek sangat kecil. Dalam suatu kondisi, manajemen proyek yang baik adalah yang

memiliki ukuran presisi yang baik dalam perkiraan waktu dan sumber daya. Metode untuk memperkirakan waktu proyek terdapat dalam COCOMO (*Constructive Cost Model*) yang sangat populer dan banyak digunakan dalam hal perhitungan perkiraan sumber daya termasuk waktu atau durasi proyek perangkat lunak. Pada tahun 2000, Barry Boehm memperkenalkan COCOMO II yang mana memiliki atribut lebih banyak daripada COCOMO I. Dengan bertambahnya atribut maka akan meningkatkan keakuratan saat memperkirakan waktu proyek berlangsung. Beberapa penelitian sebelumnya masih menggunakan metode COCOMO I yang mana masih memiliki 12 atribut. Sehingga hasil perkiraan waktu untuk proyek perangkat lunak masih terdapat banyak kesalahan yang mengakibatkan nilai akurasi masih rendah. Oleh sebab itu, pada penelitian ini menggunakan atribut COCOMO II memiliki 17 atribut dengan cakupan yang lebih luas untuk memperkirakan usaha dan waktu proyek, menggunakan dataset yang berasal dari *Turkish Software Industry* dan terdiri dari 12 proyek serta *Effort Multiplier* seperti yang ditunjukkan pada tabel dibawah ini.

**Tabel 1 Nilai effort multiplier**

No.	Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
1.	rely	0,82	0,92	1	1,1	1,26	-
2.	data	-	0,9	1	1,14	1,28	-
3.	resu	-	0,95	1	1,07	1,15	1,24
4.	docu	0,81	0,91	1	1,11	1,23	-
5.	time	-	-	1	1,11	1,29	1,63
6.	stor	-	-	1	1,05	1,17	1,46
7.	pvol	-	0,87	1	1,15	1,3	-
8.	acap	1,42	1,19	1	0,85	0,71	-
9.	pcap	1,34	1,15	1	0,88	0,76	-
10.	pcon	1,29	1,12	1	0,9	0,81	-
11.	apex	1,22	1,1	1	0,88	0,81	-
12.	plex	1,19	1,09	1	0,91	0,85	-
13.	ltex	1,2	1,09	1	0,91	0,84	-
14.	tool	1,17	1,09	1	0,9	0,78	-
15.	site	1,22	1,09	1	0,93	0,86	0,8
16..	sced	1,43	1,14	1	1	1	-
17.	cplx	0,73	0,87	1	1,17	1,34	1,74

Pada tabel 1 diatas menunjukkan nilai *effort multiplier* secara umum, adapun *cost driver rely* dapat dijelaskan pada tabel dibawah ini.

**Tabel 2 Deskripsi Rely**

Rely Descriptor	Slight Inconvenience	Low, easily recoverable losses	Moderate, easily recoverable losses	High Financial Loss	Risk to Human Life	Extra High
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0,82	0,92	1,00	1,10	1,26	n/a

Pada tabel 2 diatas menunjukkan tingkatan *reliability* yaitu *Required Software Reliability (RELY)* dari perangkat lunak yang akan dibangun. Sedangkan nilai *database* dapat dilihat pada tabel dibawah ini.

**Tabel 3 Deskripsi Data**

Data Descriptor	Testing DB bytes/pgm sloc <10	10<D/P<100	100>D/P<1000	D/P ≥ 1000	Extra High	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0,92	1,00	1,10	1,26	n/a

Pada tabel 3 diatas menunjukkan *Database size* yaitu besaran data yang disimpan pada penyimpanan. Sedangkan nilai *reusability* dapat dilihat pada tabel dibawah ini.

**Tabel 4 Deskripsi Resu**

Resu Descriptor		none	Across project	Across program	Across product line	Across multiple product lines
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0,95	1,00	1,07	1,15	1,24

Pada tabel 4 diatas menunjukkan *Develop of reusability*, yaitu memperkirakan usaha yang dibutuhkan dalam rangka mengembangkan komponen yang akan digunakan kembali dalam proyek. Sedangkan nilai *documentation* dapat dilihat pada tabel dibawah ini.

**Tabel 5 Deskripsi Docu**

Docu Descriptor	Many life cycle needs uncovered	Some life cycle needs uncovered	Right sized to life cycle needs	Excessive for life cycle needs	Very excessive for life cycle needs	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0,81	0,91	1,00	1,11	1,23	n/a

Pada tabel 5 diatas menunjukkan *Documentation match to life cycle needs*, yaitu penjelasan mengenai kesesuaian dokumentasi proyek dengan kebutuhan. Sedangkan nilai ketepatan waktu dapat dilihat pada tabel dibawah ini.

**Tabel 6 Deskripsi Time**

Time Descriptor			≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1,00	1,11	1,29	1,63

Pada tabel 6 diatas menunjukkan *Execution time constraint*, yaitu penjelasan ketepatan waktu eksekusi perintah yang diharapkan. Sedangkan nilai *storage* dapat dilihat pada tabel dibawah ini.

**Tabel 7 Deskripsi Stor**

Time Descriptor			≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1,00	1,11	1,17	1,48

Pada tabel 7 diatas menunjukkan *Main storage constraint*, yaitu merepresentasikan besaran penyimpanan yang digunakan. Sedangkan nilai kompleksitas dapat dilihat pada tabel dibawah ini.

**Tabel 8 Deskripsi Pvol**

Time Descriptor		Major change every 12 month minor change every 1 month	Major: 6 month Minor: 2 weeks	Major: 2 month Minor: 1 week	Major: 2 weeks Minor: 2 days	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0,87	1,00	1,15	1,30	n/a

Pada tabel 8 diatas menunjukkan *Platform volatility*, yaitu menjelaskan kompleksitas perangkat lunak dan keras yang digunakan seperti OS dan DBMS. Sedangkan nilai *capability* dapat dilihat pada tabel dibawah ini.

**Tabel 9 Deskripsi Acap**

Acap Descriptor	15 <sup>th</sup> percentile	35 <sup>th</sup> percentile	55 <sup>th</sup> percentile	75 <sup>th</sup> percentile	90 <sup>th</sup> percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,42	1,19	1,00	0,85	0,71	n/a

Pada tabel 9 diatas menunjukkan *Analyst capability*, yaitu parameter yang menjelaskan kemampuan dari analis dalam mengobservasi kebutuhan serta desain. Sedangkan nilai *programmer* dapat dilihat pada tabel dibawah ini.

**Tabel 10 Deskripsi Pcap**

Pcap Descriptor	15 <sup>th</sup> percentile	35 <sup>th</sup> percentile	55 <sup>th</sup> percentile	75 <sup>th</sup> percentile	90 <sup>th</sup> percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,42	1,19	1,00	0,85	0,71	n/a

Pada tabel 10 diatas menunjukkan *programmer capability*, yaitu parameter yang menjelaskan kemampuan *programmer* sebagai sebuah tim yang mampu bekerjasama. Sedangkan nilai personal dapat dilihat pada tabel dibawah ini.

**Tabel 11 Deskripsi Pcon**

Pcon Descriptor	15 <sup>th</sup> percentile	35 <sup>th</sup> percentile	55 <sup>th</sup> percentile	75 <sup>th</sup> percentile	90 <sup>th</sup> percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,42	1,19	1,00	0,85	0,71	n/a

Pada tabel 11 diatas menunjukkan *personnel continuity*, yaitu parameter yang mendeskripsikan persentase pergantian pegawai dalam setahun. Sedangkan nilai pengalaman *programmer* dapat dilihat pada tabel dibawah ini.

**Tabel 12 Deskripsi Apex**

Apex Descriptor	≤ 2 months	6 months	1 year	3 years	6 years	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,22	1,10	1,00	0,88	0,81	n/a

Pada tabel 12 diatas menunjukkan *application experience*, yaitu parameter yang menjelaskan tingkat pengalaman tim pengembang dalam mengembangkan perangkat lunak. Sedangkan nilai pengalaman *platform* dapat dilihat pada tabel dibawah ini.

**Tabel 13 Deskripsi Plex**

Plex Descriptor	≤ 2 months	6 months	1 year	3 years	6 years	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,19	1,09	1,00	0,91	0,85	n/a

Pada tabel 13 diatas menunjukkan *platform experience*, yaitu parameter yang menjelaskan tingkat pengalaman tim pengembang dalam mengembangkan antar muka, antar muka, jaringan dan distribusi sistem. Sedangkan nilai kompleksitas alat dapat dilihat pada tabel dibawah ini.

**Tabel 14 Deskripsi Ltex**

Ltex Descriptor	≤ 2 months	6 months	1 year	3 years	6 years	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,20	1,09	1,00	0,91	0,84	

Pada tabel 14 diatas menunjukkan *Language and tool experience*, yaitu parameter yang menjelaskan tingkat pengalaman tim pengembang dalam bahasa pemrograman dan penggunaan alat. Sedangkan nilai pengalaman bahasa pemrograman dapat dilihat pada tabel dibawah ini.

**Tabel 15 Deskripsi Tool**

Tool Descriptor	Edit, code, debug	Sample frontend, backend, case, little integration	Basic lifecycle tools, moderately integrated	Strong, mature lifecycle tools, moderately	Strong, mature, proactive lifecycle tools,	
-----------------	-------------------	--	--	--	--	--

				integrated	well integrated with processes methods, reuse	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,17	1,09	1,00	0,91	0,76	n/a

Pada tabel 15 diatas menunjukkan *use of software tools*, yaitu parameter yang menjelaskan kompleksitas alat yang digunakan dalam pengembangan. Sedangkan nilai penunjang komunikasi dapat dilihat pada tabel dibawah ini.

**Tabel 16 Deskripsi Site**

Site: collocation descriptors	Some phone, mail	Individual phone, fax	Narrow band email	Wideband electronic communication	Wideband electronic communicasional video conf.	Interactive multimedia
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,22	1,09	1,00	0,93	0,86	0,80

Pada tabel 16 diatas menunjukkan *multisite development*, yaitu parameter yang merepresentasikan dua hal berupa distribusi data dan penunjang komunikasi. Sedangkan nilai jadwal pengembangan dapat dilihat pada tabel dibawah ini.

**Tabel 17 Deskripsi Sced**

Sced descriptors	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,43	1,14	1,00	1,00	1,00	n/a

Pada tabel 17 diatas menunjukkan *required development schedule*, yaitu parameter yang menjelaskan persentasi dari akselerasi penyelesaian proyek terhadap waktu yang diberikan. Sedangkan nilai kompleksitas produk dapat dilihat pada tabel dibawah ini.

**Tabel 18 Deskripsi Cplx**

Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1,20	1,09	1,00	0,91	0,84	

Pada tabel 18 diatas menunjukkan *product complexity*, yaitu parameter yang dibagi dalam lima wilayah yaitu operasi pengendalian, komputasional, kemandirian perangkat, manajemen data, dan manajemen antarmuka.

Penelitian ini tidak hanya menggunakan atribut COCOMO II namun juga dengan metode perubahan pada nilai parameter perkiraan waktu proyek. COCOMO II memiliki 4 model yaitu aplikasi komposisi model, awal desain model, model yang digunakan kembali, dan model pos arsitektur. Persamaan pada COCOMO II adalah sebagai berikut (Sharma, 2011).

*Konstanta Effort:*

$$A=2.94; B=0.91 \quad (1)$$

$$PM_{NS} = A \times \text{Size}^E \times \prod_{i=1}^n EM_i \quad (2)$$

$$\text{dimana } E = B + 0.01 \times \sum_{j=1}^5 SF_j \quad (3)$$

*Konstanta Schedule:*

$$C=3.67; D=0.28 \quad (4)$$

$$TDEV_{NS} = C \times (PM_{NS})^F \quad (5)$$

$$\begin{aligned} \text{dimana } F &= D + 0.2 \times 0.01 \times \sum_{j=1}^5 SF_j \\ &= D + 0.2 \times (E - B) \end{aligned} \quad (6)$$

## 2. Tinjauan Pustaka

### 2.1 Proyek Perangkat Lunak

Proyek perangkat lunak merupakan suatu proyek yang sangat berbeda apabila kita bandingkan dengan beberapa proyek lain, sebab tujuannya adalah untuk menghasilkan produk yang jelas, berwujud, dan bisa dipahami. Kerumitan manajemen proyek perangkat lunak pun melebihi beberapa proyek lain. Melalui fakta ini, maka siklus produksi menjadi sangat rumit dalam proyek perangkat lunak. Ada begitu banyak alasan ketika proyek perangkat lunak tidak dapat diselesaikan tepat waktu (Khatibi.2012), yaitu ketika manajer proyek perangkat lunak dihadapkan dengan produksi yang tidak pasti dan tidak stabil yang sulit untuk di kontrol. Terdapat pula masalah komunikasi dapat menjadi faktor dalam keterlambatan dan kegagalan proyek (Potdar dan Potdar.2014). Sebagai solusi dapat dilakukan analisis faktor yang efektif pada perkiraan waktu pengembangan yang berdampak pada pengurangan masalah yang ada. Investigasi atribut proyek, keterbatasan, isu-isu manajemen dan pengetahuan pengembangan dapat berguna untuk menarik kesimpulan dalam hal faktor yang efektif pada manajemen perkiraan waktu proyek. Sebuah survei terbaru mengatakan bahwa lebih dari 60% dari proyek perangkat lunak gagal karena berbagai alasan. Maka dari itu, dengan diterapkannya beberapa prinsip manajemen proyek perangkat lunak yang tepat diharapkan proyek dapat selesai tepat waktu (Dasari, Kavya, Jasti.2015).

### 2.2 Perkiraan Perangkat Lunak

Proses memprediksi jumlah usaha yang dibutuhkan untuk membangun sistem perangkat lunak. Dalam beberapa tahun, banyak cara untuk memperkirakan usaha proyek yang telah diusulkan untuk evaluasi kinerja perkiraan (Reddy dan Raju.2009).

### 2.3 Cocomo II

Secara umum COCOMO memiliki tiga kelas proyek perangkat lunak, yaitu proyek organik yang memiliki tim “kecil” dengan pengalaman bekerja yang baik dengan kebutuhan yang tidak terlalu “ketat” (Malik, Pandey, Kaushik.2013) selain itu proyek organik termasuk proyek yang relatif kecil, mirip proyek yang dikembangkan sebelumnya, membutuhkan sedikit inovasi (Jaglan.2016). Kemudian proyek semi yang memiliki tim “menengah” dengan beragam pengalaman kerja dan kebutuhan yang kurang “ketat”, proyek ini termasuk juga dalam level intermediate dalam ukuran dan kerumitan (Jaglan.2016). Serta proyek terintegrasi, yang dikembangkan dalam sekumpulan hambatan yang sulit. Kombinasi antara perangkat keras, perangkat lunak, operasional (Malik, Pandey, Kaushik.2013).

## 3. Metode Penelitian

Permasalahan utama dalam penelitian ini adalah bagaimana cara untuk meningkatkan akurasi perkiraan waktu dari proyek perangkat lunak dengan menggunakan COCOMO II. Peningkatan akurasi dilakukan dengan mengubah nilai parameter waktu yang terdapat pada COCOMO II. Data yang digunakan pada uji coba ini didapatkan dari Industri Perangkat Lunak Turki yaitu Cocomo Sdr. Data tersebut terdiri atas 12 proyek dengan 25 atribut, 17 driver, 5 faktor skala, SLOC, nilai sebenarnya. Dengan beberapa tahap skenario.

- a. Implementasi *dataset* cocomo sdr menggunakan atribut dari COCOMO II dengan nilai parameter yang telah pada COCOMO II tanpa perubahan apapun.
- b. Implementasi *dataset* cocomo sdr menggunakan atribut dari COCOMO II yang nilai parameter waktu C dan D telah diubah menjadi gradual 0,0001
- c. Membandingkan hasil dari kedua cara diatas.

Adapun perubahan nilai parameter waktu yaitu C dan D diubah secara bertahap, yaitu dimulai dari gradual 0,1 sampai gradual 0,0001. Yang mana hasil akurasi optimal akan dicapai melalui proses perubahan gradual seperti dibawah ini.

**Tabel 19 Perubahan Gradual 0,1**

Solusi Parameter C	Solusi Parameter D	MMRE (%)
3,57	0,18	124,81

Pada tabel 3.1 merupakan tahapan perubahan gradual 0,1 pada nilai parameter waktu. Total nilai kesalahan adalah 124,81%. Sedangkan perubahan gradual 0,01 dapat ditunjukkan seperti dibawah ini.

**Tabel 20 Perubahan Gradual 0,01**

Solusi Parameter C	Solusi Parameter D	MMRE (%)
3,66	0,27	175,14

Pada tabel 3.2 merupakan tahapan perubahan gradual 0,01 pada nilai parameter waktu. Total nilai kesalahan adalah 175,14%. Sedangkan perubahan gradual 0,001 dapat ditunjukkan seperti dibawah ini.

**Tabel 21Perubahan Gradual 0,001**

Solusi Parameter C	Solusi Parameter D	MMRE (%)
3,66	0,27	181,71

Pada tabel 3.3 merupakan tahapan perubahan gradual 0,001 pada nilai parameter waktu. Total nilai kesalahan adalah 181,71%. Sedangkan perubahan gradual 0,0001 dapat ditunjukkan seperti dibawah ini.

**Tabel 22 Perubahan Gradual 0,0001**

Solusi Parameter C	Solusi Parameter D	MMRE (%)
3,6699	0,2799	182,38

Pada tabel 3.4 merupakan tahapan perubahan gradual 0,0001 pada nilai parameter waktu. Total nilai kesalahan adalah 182,38%.

#### 4. Hasil dan Pembahasan

Pada tahapan ini uji coba pada penelitian ini melakukan implementasi COCOMO II dengan mengubah nilai parameter waktu yaitu C dan D dengan *dataset* cocomo sdr dan membandingkan dengan penelitian sebelumnya yang menggunakan COCOMO II tanpa mengubah nilai parameter C dan D dengan *dataset* cocomo sdr. Hasil terbaik yang di dapat dari uji coba yaitu pada gradual 0,1 dengan MMRE yang didapat yaitu 124,81. Solusi parameter C yaitu 3,57 sedangkan solusi parameter D yaitu 0,18. Seperti yang terlihat pada tabel dibawah ini.

**Tabel 23 Perubahan Gradual 0,1**

Solusi Parameter C	Solusi Parameter D	MMRE (%)
3,57	0,18	124,81

#### 5. Penutup

Pada penelitian ini, COCOMO II telah dapat memberikan hasil akurasi yang baik. Hanya saja masih belum dapat optimal, dan masih terdapat kesalahan yang cukup besar jika dibandingkan dengan waktu proyek perangkat lunak yang sebenarnya. Oleh sebab itu penelitian ini menggunakan perubahan nilai parameter dengan beberapa tahap penurunan gradual. Dari uji coba tersebut didapatkan hasil bahwa kesalahan (MMRE) dapat berubah menjadi 124,81% yang artinya menurun sebanyak 0,08%.

#### Referensi

- Jaglan, V. (2016) 'Apply Fuzzy Optimization in Proficient Managing COCOMO Model Cost Drivers', *International Journal of Recent Aspects*, 3(1), pp. 20–26.
- Khatibi, E. (2012) 'Efficient Indicators to Evaluate the Status of Software Development Effort Estimation inside the Organizations', *International Journal of Managing Information Technology*, 4(3), pp. 23–32. doi: 10.5121/ijmit.2012.4303.

- L.Dasari, M.A.Kavya, V.Jasti, M. A. (2015) ‘An Analysis on the Role of Project Management in the Success Chronicle of Software Project’, *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(9), pp. 91–94.
- Malik, A., Pandey, V. and Kaushik, A. (2013) ‘An analysis of fuzzy approaches for COCOMO II’, *International Journal of Intelligent Systems and Applications*, 5(5), pp. 68–75. doi: 10.5815/ijisa.2013.05.08.
- Reddy, C. S. and Raju, K. (2009b) ‘Improving the accuracy of effort estimation through fuzzy set combination of size and cost drivers’, *WSEAS Transactions on Computers*, 8(6), pp. 926–936. Available at: <http://www.wseas.us/e-library/transactions/computers/2009/31-479.pdf>.
- Sharma, T. N. (2011) ‘Analysis of Software Cost Estimation using COCOMO II’, *International Journal of Science & Engineering Research*, 2(6), pp. 1–5. Available at: [http://www.ijser.org/researchpaper/Analysis\\_of\\_Software\\_Cost\\_Estimation\\_using\\_COCOMO\\_II.pdf](http://www.ijser.org/researchpaper/Analysis_of_Software_Cost_Estimation_using_COCOMO_II.pdf).