

Sistem Penentu Estimasi Biaya Proyek Pengembangan Perangkat Lunak Menggunakan Metode COCOMO II

¹⁾Eko Febri Harsono ²⁾Anjik Sukmaaji ³⁾Arifin Puji Widodo ⁴⁾Sholih

Program Studi/Jurusan Sistem Informasi
Institut Bisnis dan Informatika Stikom Surabaya
Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email : 1)16410100003@dinamika.ac.id, 2)anjik@dinamika.ac.id, 3)arifin@dinamika.ac.id, 4)sholih99@gmail.com

Abstract: *Estimated cost of the software project is very important for the government before the project tendered. The government must determine the HPS (Harga Perkiraan Sendiri) by surveying the price of the project in each region. Because there is no specific regulation related to the HPS itself in the presidential regulation. Perpres No. 70 of 2012 explains that HPS is only for the procurement of goods or services and there are no specific regulations related to HPS for software projects. Therefore, the government has difficulty determining the estimated total value of software development projects. based on these problems, the application of determining the cost of a software development project was built using the COCOMO II method. The application built can determine the estimated total value of a software development project using the COCOMO II method before the project is tendered. So the government is easy to determine the HPS in each software development project.*

Keywords: COCOMO II, HPS, Estimation Cost, Software Project.

Pada umumnya dalam proyek pemerintahan Indonesia. Sebelum proyek ditenderkan perlu mengetahui estimasi biaya dalam pembuatan proyek perangkat lunak sebelum proyek tersebut ditenderkan. estimasi biaya tersebut adalah HPS (Harga Perkiraan Sendiri) (Ambari, 2015). HPS sangat dibutuhkan pada setiap pengembangan proyek perangkat lunak. Supaya pihak pemerintah mendapatkan pedoman sebagai acuan khusus atau estimasi biaya proyek (Purwanto, 2017).

HPS disusun atau dibuat dengan melakukan survei harga pasar lokal pada tiap daerah. Tidak ada acuan khusus terkait HPS, setiap daerah bisa membuat HPS yang berbeda-beda karena setiap daerah mempunyai harga pasar yang berbeda - beda dengan proyek pengembangan perangkat lunak yang sama. Selain itu penentuan HPS untuk proyek pengembangan perangkat juga masih berpedoman pada Peraturan Presiden No 70 Tahun 2012 (Perpres, 2012). Tetapi di pada Peraturan Presiden tersebut fokus terhadap pengadaan barang atau jasa, tidak ada peraturan khusus terkait HPS untuk proyek pengembangan perangkat lunak (Sholih, 2019).

Salah satu metode algoritma yang paling populer dalam hal menentukan estimasi biaya proyek pengembangan perangkat lunak adalah metode COCOMO II (*Constructive Cost Model*). Metode COCOMO II memiliki perhitungan yang

lebih detail terkait tentang menentukan estimasi biaya atau HPS pada proyek pengembangan perangkat lunak serta bisa digunakan dalam lingkup besar atau kecil (Boehm, 1981). Metode COCOMO II mempunyai 3 sub model yaitu, *early design*, *application composition*, dan *post architecture*. Dalam penelitian ini menggunakan 1 sub model yaitu *post architecture* (Amru Nizar Maqdam, 2019).

Sub model *post architecture* adalah sub model yang paling cocok dalam menyelesaikan masalah pada penelitian ini. Sub model *post architecture* dapat menentukan estimasi biaya proyek pengembangan perangkat lunak sebelum proyek ditenderkan dengan lingkup proyek yang besar maupun kecil.

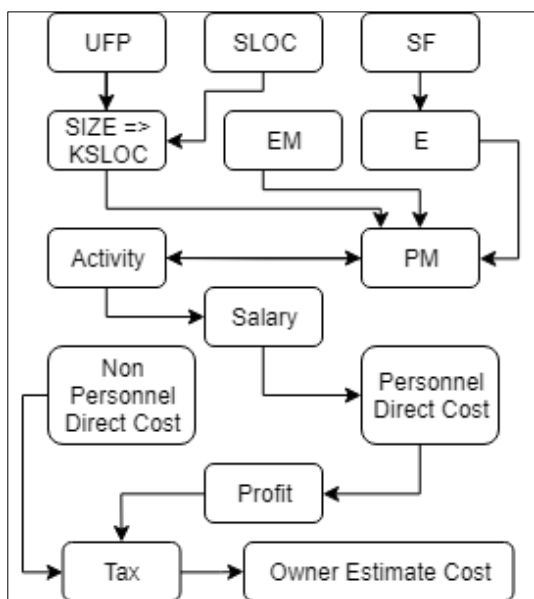
Terdapat beberapa jurnal yang membahas tentang implementasi COCOMO II pada sebuah proyek pihak pemerintahan. Tetapi fokus terhadap hitungan dan menghitung manual dengan excel. Salah satu jurnal tersebut adalah "*Implementing COCOMO II as personnel direct cost in an owner estimate cost model for software project*" jurnal tersebut menghitung nilai total proyek dengan 4 dataset dari pihak pemerintahan dan masih menggunakan cara yang manual.

Berdasarkan permasalahan diatas maka peneliti membuat sistem penentu estimasi biaya proyek pengembangan perangkat lunak menggunakan metode COCOMO II dengan sub model *post architecture*. Hasil akhir yang

diharapkan dari penelitian ini berupa aplikasi yang nantinya dapat membantu pihak pemerintah atau perusahaan *software house* dalam menentukan HPS pada proyek pengembangan perangkat lunak yang akan datang.

METODE

Dalam penelitian judul ini, untuk menentukan estimasi nilai total proyek pihak pemerintahan adalah menggunakan metode perhitungan COCOMO II. Dalam metode COCOMO II dibutuhkan beberapa parameter yang harus dipenuhi dan langkah – langkah yang harus dilakukan oleh *user* untuk mendapatkan *owner estimate cost*. Lebih jelasnya terdapat pada gambar 1.



Gambar 1. Diagram COCOMO II (Oleh Penulis)

Menghitung UFP

Unjusted Function Point (UFP) adalah langkah pertama atau parameter pertama yang digunakan oleh metode COCOMO II. UFP merupakan sebuah matrik untuk mengukur kebutuhan dari perangkat lunak dengan cara mengkuantisasi pengolahan informasi seperti *Internal Logic File* (ILF), *External Interfaces File* (EIF), *External Input* (EI), *External Output* (EO) dan *External Inquiry* (EQ) (Febrina Putri Laksamana, 2018). 5 komponen tersebut mempunyai bobot tertentu seperti tabel 1.

Tabel 1. Bobot *Unjusted Function Point*

Function Type	Complexity Weight		
	Low	Average	High
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

Sumber : (Boehm, 2000)

Terdapat beberapa langkah untuk menghitung nilai total UFP sebagai berikut :

1. Menentukan batas – batas aplikasi
2. Mengidentifikasi dan menghitung nilai ILF dan EIF
3. Mengidentifikasi dan menghitung nilai EI, EO dan EQ.
4. Menjumlahkan nilai ILF, EIF, EI, EO dan EQ (Pradani, 2019).

$$\begin{aligned}
 UFP = & (ILF \times \text{bobot ILF}) \\
 & + (EIF \times \text{bobot EIF}) \\
 & + (EI \times \text{bobot EI}) \\
 & + (EO \times \text{bobot EO}) \\
 & + (EQ \times \text{bobot EQ})
 \end{aligned}$$

Menghitung SIZE

Source Line Of Code (SLOC) adalah ukuran / matrik yang umumnya digunakan dalam kode program, demikian juga COCOMO II menggunakan SLOC tetapi dalam satuan Kilo SLOC sebagai parameter perhitungan COCOMO II. Nilai ukuran didapat dari perkalian UFP dengan rasio konversi SLOC dari bahasa pemrograman yang digunakan dalam pengembangan perangkat lunak. rasio konversi SLOC sebagai contoh salah satu bahasa programan PHP seperti tabel 2. Untuk lebih detail ada pada dokumentasi *Model Definision Manual COCOMO II* oleh Dr.Barry Boehm.

$$SIZE = UFP \times SLOC \text{ Conversion Ratios}$$

Tabel 2. Konversi Rasio SLOC

Bahasa	Default SLOC
Third generation Language (PHP)	80

Sumber : (Boehm, 2000)

Menghitung Exponent E

Eksponent E diperoleh dengan menggabungkan lima *Scale Factor* (SF). Formula

ini dibangun di atas satu nilai konstan, yaitu B (B = 0,91) dan penjumlahan dari 5 nilai *Scale Factor*. Definisi 5 *Scale Factor* dan Nilai *default Scale Factor* bisa dilihat pada tabel 3 dan 4.

$$SF = PREC + FLEX + RESL + TEAM + PMAT$$

$$E = B + 0.01 \times \prod SF.$$

Tabel 3. Diskripsi 5 *Scale Factor*

Scale Factor	Diskripsi
<i>Precedentedness</i> (PREC)	Mencerminkan organisasi sebelumnya
<i>Flexibility of development</i> (FLEX).	Mencerminkan tingkat fleksibilitas dalam proses pengembangan
<i>Risk resolution</i> (RESL)	Mencerminkan tingkat resiko analisis
<i>Team of cohesion</i> (TEAM)	Mencerminkan seberapa baik tim pengembang dalam kerja sama
<i>Project maturity</i> (PMAT).	Mencerminkan kematangan dalam oarganisasi

Sumber : (Boehm, 2000)

Tabel 4. *Default Value 5 Scale Factor*

SF	Bobot				
	VL	L	N	H	VH
PREC	6.20	4.96	3.72	2.48	1.24
FLEX	5.07	4.96	3.04	2.03	1.01
RESL	7.07	5.65	4.24	2.83	1.41
TEAM	5.48	4.38	3.29	2.19	1.10
PMAT	7.80	6.24	4.68	3.12	1.56

Sumber : (Boehm, 2000)

Keterangan:

- SF : *Scale Factor*
- VL : *Very Low*
- L : *Low*
- N : *Nominal*
- H : *High*
- VH : *Very Hight*

Menghitung *Person-Month*

Person-month adalah jumlah waktu yang dihabiskan satu orang bekerja pada proyek pengembangan perangkat lunak selama satu bulan. jumlah *person-month* bisa saja berbeda dengan kebutuhan setiap proyek (Boehm, 2000).

Nilai EM didapat dari menjumlahkan 17 faktor *effort multiplier*. 17 faktor tersebut bisa dilihat pada tabel 5. Tabel 5 hanya menunjukkan nilai *default* 17 faktor *effort multipliers*. Untuk detail setiap faktor dan definisinya bisa melihat dokumentasi *Model Definision Manual COCOMO II* oleh Dr.Barry Boehm A = (2.94) adalah *default value* dari rumus COCOMO II, sedangkan E didapat dari perhitungan sebelumnya yaitu Exponent E.

$$EM = rely \times data \times cplx \times ruse \times docu \times time \times stor \times pvol \times acap \times pcap \times pcon \times apex \times plex \times ltex \times tool \times site$$

$$PM = A \times SIZE^E \times \prod EM$$

Tabel 5. *Default Value 17 Effort Multiplier*

EM	Bobot					
	VL	L	N	H	VH	EH
rely	0.82	0.92	1	1.10	1.26	-
data	-	0.90	1	1.14	1.28	-
cplx	0.73	0.87	1	1.17	1.34	1.74
ruse	-	0.95	1	1.07	1.15	1.24
docu	0.81	0.91	1	1.11	1.23	-
time	-	-	1	1.11	1.29	1.63
stor	-	-	1	1.05	1.17	1.46
pvol	-	0.87	1	1.15	1.30	-
acap	1.42	1.19	1	0.85	0.71	-
pcap	1.34	1.15	1	0.88	0.76	-
pcon	1.29	1.12	1	0.90	0.81	-
apex	1.22	1.10	1	0.88	0.81	-
plex	1.19	1.09	1	0.91	0.85	-
ltex	1.20	1.09	1	0.91	0.84	-
tool	1.17	1.09	1	0.90	0.78	-
site	1.22	1.09	1	0.93	0.86	-

Sumber : (Boehm, 2000)

Keterangan:

- EM : *Effort Multiplier*
- VL : *Very Low*
- L : *Low*
- N : *Nominal*
- H : *High*
- VH : *Very Hight*
- EH : *Extra High*

Activity

Untuk melihat *activity* apa saja itu bergantung pada metode pengembangan perangkat lunak. yang telah digunakan saat ini

adalah metode pengembangan perangkat lunak menggunakan *waterfall*. Dan bisa saja menggunakan metode *Prototype*. Untuk nilai dari metode pengembangan *waterfall* bisa dilihat pada tabel 6. Setiap *activity* harus dihitung kemudian hasil setiap *activity* harus dijumlahkan. Sebagai contoh rumus dibawah ini adalah menghitung 2 *activity* saja, kemudian menjumlahkan semua *activity*. Hasil penjumlahan akan dibandingkan hasil PM (*person-month*) dan harus sama.

$$totalreq = medium\ or\ small \times PM$$

$$totalspec = medium\ or\ small \times PM$$

$$totalallactivity = totalreq + PMtotalspec$$

Keterangan:

totalreq : Total Activity Requirements
totalspec : Total Activity Specification
totalallactivity : Total Semua Activity

Tabel 6. Value of Waterfall

No	Activity	Distribution per project size (%)	
		Medium - Large	Small
1	Requirements	7.50	1.17
2	Specifications	7.50	6.75
3	Design	10.00	5.57
4	Implementation	10.00	55.65
5	Integration Testing	7.50	6.42
6	Acceptance & deployment	7.50	5.60
7	Project Management	8.34	2.55
8	Configuration management	4.16	3.58
9	Quality Assurance	8.34	2.55
10	Documentation	4.16	9.76
11	Training & support	4.16	0.60
12	Evaluation & testing	20.84	1.67

Sumber : (Sholih, 2019)

Menghitung Personnel Direct Cost

Personnel direct cost menjabarkan siapa saja yang terlibat dalam peroyek pengembangan perangkat lunak dan berapa gaji orang yang terlibat. Pada pihak *internal* pemerintahan bisa mendapatkan gaji 100% . sedangkan pada pihak luar mendapatkan 70% dari gaji seperti pada tabel 7. Contoh : (*internal*) Project Manager (IDR) 10.000.000. (*software house*) Project Manager (IDR) 7.000.000.

Tabel 8. Payment Rate Personnel

Position in Project Team	Minimum Requirement
Project Manager	Undergraduate and experience > 4 years
System Analyst	Undergraduate and experience > 3 years
Programmer	Experience 1 -2 years
System Testing	Experience 1 -2 years
Technical Support	Experience 1 -2 years
Documenter	Experience 1 -2 years

Sumber : (Sholih, 2019)

Menghitung Non Personnel Direct Cost

Non personnel direct cost menjabarkan kebutuhan apa saja dalam mendukung proses pengembangan perangkat lunak tersebut. Atau bisa disebut dengan biaya diluar personil seperti pada tabel 9. sebagai contoh biaya transport ke tempat *meeting*, biaya seminar, *workshop*, sosialisasi terkait perangkat lunak yang telah dikembangkan dan lain sebagainya. Formula yang digunakan untuk menghitung *non personnel direct* adalah sebagai berikut. Formula tersebut sangat dinamis, boleh menambahkan biaya sesuai dengan kebutuhan proyek.

$$CNONPER = TT + CP + SEM + SUR + LAB + COP$$

Keterangan:

CNONPER : Cost Non Personnel
TT : Transportation Ticket
CP : Computer & printer
SEM : Seminar
SUR : Survei
LAB : Laboratory Test
COP : Copyright

Tabel 9. Payment Rate Non Personnel

No	Cost driver of non personnel direct cost	Cost
1	Transportation ticket	-
2	Computer & printer	-
3	Seminar, workshop, socialization, training	-
4	Survey	-
5	Laboratory test	-
6	Copyright	-

Sumber : (Sholih, 2019)

Menghitung Cost Per Activity

Menghitung biaya total per *activity waterfall* untuk mendapatkan total biaya semua *activity* dalam metode *waterfall*. Semua *activity* harus dihitung dan semua hasil perhitungan *activity* tersebut harus dijumlahkan. Sebagai contoh hanya menghitung 2 *activity* dengan formula dibawah ini.

$$CAR = \text{payrate} \times 0,923 \times \text{totalreq}$$

$$CAS = \text{payrate} \times 0,923 \times \text{totalspec}$$

$$CAW = CAR + CAS$$

Keterangan:

- CAR : Cost Activity Requirement
- CAS : Cost Activity Specification
- CAW : Cost All Waterfall

Menghitung Profit

Suatu proyek pasti mempunyai *profit* tersendiri. Minimal profit dari pihak pemerintahan yaitu 10%. Profit diambil dari nilai total biaya *activity* (*cost personnel*). Tidak diambil dari biaya *non personnel*. Formula untuk menghitung *profit* yaitu sebagai berikut :

$$\text{profit} = 10\% \times \text{cost personnel}$$

$$CPWP = \text{cost personnel} + \text{profit}$$

Keterangan:

- CPWP : Cost Personnel With Profit

Menghitung Tax

Suatu proyek pasti mempunyai pajak tersendiri. Minimal pajak dari pihak pemerintahan yaitu 10%. tax diambil dari nilai total biaya *activity personnel* dengan *profit* dan *non personnel*. Formula untuk menghitung *tax* yaitu sebagai berikut.

$$TAX = 10\% \times (CPWP + NONPER)$$

$$CAWT = CPWP + NONPER + TAX$$

Keterangan:

- CPWP : Cost Personnel With Profit
- NONPER : Cost Non Personnel
- CAWT : Cost All With Tax

Menghitung Total Estimasi

Menghitung semua biaya yg telah dikeluarkan. *Owner estimate* dihitung dengan cara menjumlahkan semua biaya dari biaya

personnel, non personnel, dan total pajak. Formula untuk menghitung tersebut yaitu sebagai berikut.

$$\text{Owner Estimate} = TAX + CAWT$$

Keterangan:

- CAWT : Cost All With Tax

HASIL DAN PEMBAHASAN

Fokus penelitian adalah menghitung 2 dataset dari pihak pemerintahan. Dataset bisa dilihat pada tabel 10.

Tabel 10. Dataset

Modul-ID	Deskripsi Proyek	Provinsi	Bahasa Pemrograman
1	E-SDM DPRKPCCK TR. Website of Public Housing and Residential Areas, Creative Works and Spatial Planning in Surabaya	East Java	PHP, MySQL
2	Archive of DLH. Website of Public Environmental Services in Surabaya	East Java	PHP, MySQL

Sumber : (Sholih, 2019)

Tabel 11 adalah tabel *result input* untuk mencari nilai total UFP. Nilai input tersebut akan dikalikan dengan bobot UFP untuk mendapatkan hasil nilai total UFP.

Tabel 11. Input UFP

UFP	Level Bobot	Modul - ID	
		1	2
ILF	Low	11	7
	Average	0	0
	High	0	0

EIF	Low	47	18
	Average	0	0
	High	0	0
EI	Low	54	44
	Average	7	2
	High	0	0
EO	Low	12	11
	Average	4	8
	High	0	0
EQ	Low	10	7
	Average	0	0
	High	0	0

Sumber : Oleh Penulis

Tabel 12 adalah hasil dari *input* yang telah dihitung dengan bobot UFP . hasil UFP akan diteruskan lagi pada langkah selanjutnya yaitu menghitung nilai size in SLOC.

Tabel 12. Hasil Pembobotan UFP

UFP	Level Bobot	Modul - ID	
		1	2
ILF	Low	77	49
	Average	0	0
	High	0	0
EIF	Low	235	90
	Average	0	0
	High	0	0
EI	Low	162	132
	Average	28	8
	High	0	0
EO	Low	48	44
	Average	20	40
	High	0	0
EQ	Low	30	21
	Average	0	0
	High	0	0
Total UFP		600	384

Sumber : Oleh Penulis

Tabel 13 adalah hasil pilihan *user* sesuai kebutuhan bahasa dalam pengembanga perangkat lunak. untuk pilihan bahasa pemrogramman bisa dilihat pada tabel 2. Nilai SLOC akan dikonversikan menjadi KSLOC. KLOC akan digunakan pada langkah selanjutnya.

Tabel 13. Nilai total KSLOC

Modul-ID	UFP	SLOC	SIZE	KSLOC
1	600	80	48000	48.00
2	384	80	30720	30.72

Sumber : Oleh Penulis

Keterangan :

UFP : Nilai UFP didapat dari tabel 12.

SLOC : Pilihan Bahasa Pemrogramman pada tabel 2

SIZE : Formula pada tahap menghitung size

KSLOC : *Convert size to KSLOC*

Tabel 14. *Input Scale Factor*

Scale Factor	Modul – ID	
	1	2
<i>Precedentedness</i>	4,96	3,72
<i>Flexibility of development</i>	2,03	3,04
<i>Risk resolution</i>	4,24	2,83
<i>Team of cohesion</i>	3,29	4,38
<i>Project maturity</i>	4,68	4,68

Sumber : Oleh Penulis

Tabel 14. Adalah *result input scale factor* oleh *user* dengan bobot yang telah dipilih oleh *user*. pemilihan bobot *scale factor* dapat dilihat pada tabel 4.

Tabel 15. *Result Exponent Effort*

Modul - ID	Effort
1	1,1020
2	1,0965

Sumber : Oleh Penulis

Nilai *Effort* didapat dari perhitungan tahap menghitung *exponent E*. Dan dengan formula dari COCOMO II. Nilai E akan digunakan dalam langkah selanjutnya, yaitu menghitung nilai *Person-Month (PM)*.

Tabel 16. *Result Effort Multiplier*

Effort Multiplier	Modul – ID	
	1	2
RELY	1	1
DATA	0,9	1,28
CPLX	1	1,17
RUSE	1,07	1
DOCU	0,81	1
TIME	1	1,11
STOR	1	1
PVOL	1,15	1
ACAP	0,85	0,85
PCAP	1	0,88
PCON	0,81	0,81
APEX	0,81	0,88
PLEX	0,85	0,85
LTEX	0,84	0,91
TOOL	0,9	0,9

SITE	1	1,,09
SCED	1	1
Total EM	0,3215	0,6725

Sumber : Oleh Penulis

Nilai total *Effort Multiplier* (EM) didapat dari perhitungan tahap menghitung *effort multiplier*. Dan dengan formula dari COCOMO II. Pilihan bobot EM dapat dilihat pada tabel 5. Nilai EM akan digunakan dalam langkah selanjutnya, yaitu menghitung nilai *Person-Month* (PM).

Tabel 17. Total *Person Month*

Modul-ID	KSLOC	E	EM	PM
1	48	1.102	0,3215	67,33
2	30,72	1,0965	0,6725	84,53

Sumber : Oleh Penulis

Keterangan :

KSLOC : Nilai KSLOC didapat dari tabel 13.

E : Nilai E didapat dari dari tabel 15

EM : Formula pada tahap menghitung size

KSLOC : *Convert size to KSLOC*

Nilai total *Person Month* (PM) didapat dari tahap menghitung person month. Dan dengan formula dari COCOMO II. Nilai PM akan digunakan sebagai acuan pada langkah selanjutnya.

Tabel 18. *Activity Waterfall*

No	Activity	%	Modul – ID	
			1	2
1	<i>Requirement</i>	1,17	0,79	0,99
2	<i>Spesifications</i>	6,75	4,54	5,71
3	<i>Design</i>	5,57	3,75	4,71
4	<i>Implementatio n</i>	55,65	37,47	47,04
5	<i>Integrating & testing</i>	6,42	4,32	5,43
6	<i>Acceptance & development</i>	5,6	3,77	4,73
7	<i>Project Management</i>	2,55	1,72	2,16
8	<i>Configuration Management</i>	3,58	2,41	3,03
9	<i>Quality Assurance</i>	0,66	0,44	0,56
10	<i>Documentatio n</i>	9,76	6,57	8,25

11	<i>Training & support</i>	0,62	0,42	0,52
12	<i>Evaluastion & testing</i>	1,67	1,12	1,41
Total		100	67,33	84,53

Sumber : Oleh Penulis

Nilai total *Activity* didapat dari tahap menghitung *activity*. Dan dengan formula dari COCOMO II. Nilai *Activity* akan digunakan sebagai acuan sebagai pembanding dengan nilai PM. Jika total *activity* kurang dari nilai PM maka tambahkan beberapa nilai pada *activity* hingga seperti nilai PM. Dan jika nilai *activity* sudah sama dengan PM. Nilai total per *activity* akan digunakn sebagai parameter pada langkah selanjutnya.

Tabel 19. *Personnel Direct Cost*

<i>Position in Project Team</i>	<i>Minimum Requirement</i>	<i>Salary (IDR)</i>
<i>Project Manager</i>	<i>Undergraduate and experience > 4 years</i>	18,250,000
<i>System Analyst</i>	<i>Undergraduate and experience > 3 years</i>	12,000,000
<i>Programmer</i>	<i>Experience 1 -2 years</i>	11,000,000
<i>System Testing</i>	<i>Experience 1 -2 years</i>	9,300,000
<i>Technical Support</i>	<i>Experience 1 -2 years</i>	9,300,000
<i>Documenter</i>	<i>Experience 1 -2 years</i>	5,000,000

Sumber : Oleh Penulis

Tabel 19 adalah hasil *input salary* dari user untuk setiap modul. *Salary* untuk pihak internal pemerintahan tidak ada potongan (100%). Jika digunakan untuk pihak luar seperti perusahaan *software house* mendapat potongan 70% dari gaji asli tiap posisi.

Tabel 20. *Direct Cost Per Activity*

No	Activity	Use d	Modul - ID	
			1	2
1	<i>Requirements</i>	SA	8,750,000	10,965,240
2	<i>Specifications</i>	SA	50,285,040	63,243,960
3	<i>Design</i>	SA	41,535,000	52,167,960

No	Activity	Use d	Modul - ID	
			1	2
4	Implementation	P	380,432,910	477,597,120
5	Integration Testing	ST	37,082,448	46,610,577
6	Acceptance & deployment	ST	32,361,303	40,601,847
7	Project Management	PM	28,972,970	36,384,660
8	Configuration management	SA	26,693,160	33,560,280
9	Quality Assurance	ST	3,776,916	4,806,984
10	Documentation	D	30,320,550	38,073,750
11	Training & support	TS	3,605,238	4,463,628
12	Evaluation & testing	ST	9,613,968	4,463,628
Total			653,429,543	820,579,305

Sumber : Oleh Penulis

Keterangan :

- SA : System Analyst
- P : Programmer
- ST : System Testing
- TS : Technical Support
- D : Documentation

Tabel 20 adalah hasil dari perhitungan pada tahap menghitung *Cost Per Activity* dan menggunakan formula dari COCOMO II. Data *salary* didapat dari tabel 19.

Tabel 21. *Personnel Direct Cost With Profit*

Modul - ID	PDC (IDR)	P (IDR)	PDCP (IDR)
1	653,429,543	65,342,954	718,772,497
2	820,579,305	82,057,931	902,637,236

Sumber : Oleh Penulis

Keterangan :

- PDC (*Personnel Direct Cost*) : Nilai PDC didapat dari tabel 20
- P (*Profit*) : Nilai *profit* didapat dari metode tahap perhitungan *profit*.

PDCP (*Personnel Direct Cost & Profit*) : Nilai PDCP yaitu menjumlahkan hasil dari *profit* dan PDC

Tabel 21 adalah hasil dari perhitungan pada tahap menghitung *profit*. *Profit* yang digunakan yaitu sebesar 10%.

Tabel 22. *Direct Cost Non Personnel*

No	Name	Modul - ID	
		1	2
1	Transport	3,500,000	2,760,000
2	Seminar	3,800,000	5,000,000
Total		7,400,000	7,760,000

Sumber : Oleh Penulis

Tabel 22 adalah kebutuhan dari setiap proyek yang dikerjakan. Nilai *total non personnel direct cost* didapat dari tahap menghitung *non personnel direct cost*. *Total non personnel direct cost* akan ditambahkan dengan *personnel direct cost* setelah ditambahkan oleh *profit*.

Tabel 23 *Owner Estimate Cost*

Id	PDCP	NONP	TAX	total
1	718,772,497	7,400,000	72,612,249	798,734,746
2	902,637,236	7,760,000	91,039,723	1,001,436,959
Total				1,800,171,705

Sumber : Oleh Penulis

Keterangan :

- PDCP (*Personnel Direct Cost & Profit*) : Nilai PDCP didapat dari tabel 21
- NONP (*Non Personnel*) : Nilai NONP didapat dari tabel 22
- TAX : Hasil dari metode pada tahapan menghitung TAX
- Total : Total ada menjumlah seluruh biaya yang telah dihitung. Formula menghitung total ada pada metode tahap menghitung *owner estimate cost*

Tabel 23 adalah langkah terakhir yaitu menggabungkan biaya *personnel* dengan *profit*, *non personnel* dan *tax* sebesar 10%. Untuk

menghitung *tax* ada pada tahap metode menghitung *tax*. Hasil yang didapatkan yaitu Rp. 1,800,171,705 2 project dengan menggunakan metode COCOMO II

Estimate Cost Model For Software Project. -, -.

RUJUKAN

- Ambari, P. L. (2015). *Estimasi Biaya Proyek Pengembangan Perangkat Lunak Pemerintahan Bersekala Small-Medium Dengan Metode Use Case Point (UCP)*. Surabaya: Institut Teknologi Sepuluh November.
- Amru Nizar Maqdam, A. R. (2019). Implementasi Metode COCOMO II Untuk Estimasi Biaya Pengembangan Perangkat Lunak Di CV. Profile Image Studio. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6238-6247.
- Boehm, B. (- de - de 1981). *COCOMO II MODEL DEFINITION MANUAL*. Obtenido de -: <https://pdfs.semanticscholar.org/3efa/fad7124e658bad3ccc7678d9f25ead1ca0a9.pdf>
- Boehm, B. (2000). *COCOMO II Model Definition Manual*. California: University Of California.
- Febrina Putri Laksamana, A. R. (2018). Evaluasi Biaya Pengembangan Sistem Informasi Pengelolaan Arsip Surat (SIPAS) Menggunakan Function Point dan Object Point (Studi Kasus : PT Sekawan Media Informatika). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* , 2692-2701.
- Perpres. (- de - de 2012). *Peraturan Presiden Republik Indonesia No 70 Tahun 2012*. Obtenido de Perpres: <http://www.bpkp.go.id/uu/filedownload/5/91/2253.bpkp>
- Pradani, W. (2019). Kajian Metode Perhitungan Metrik Function-Point dan Penerapannya pada Dua Perangkat Lunak yang Dipilih. *Jurnal AL-AZHAR INDONESIA SERI SAINS DAN TEKNOLOGI*, Vol. 2, No.1.
- Purwanto, D. (2017). *Aplikasi Penentu Harga Perkiraan Sendiri Proyek Pengembangan Perangkat Lunak Pemerintahan Berbasis Web*. Surabaya: Universitas Dinamika.
- Sholiq, A. P. (2019). Implementing COCOMO II As Personnel Direct Cost In An Owner