

Estimasi Biaya Perangkat Lunak INLIS LITE 3.0 Perpustakaan Nasional Menggunakan Metode Function Points

Fatimah Chibbatullah, Renny Sari Dewi*

Fakultas Teknologi Industri dan Kreatif, Sistem Informasi, Universitas Internasional Semen Indonesia, Gresik, Indonesia

Email: ¹fatmachibba99@gmail.com, ^{2,*}renny.dewi@uisi.ac.id

Email Penulis Korespondensi renny.dewi@uisi.ac.id

Submitted 08-01-2020; Accepted 04-02-2020; Published 15-02-2020

Abstrak

Dalam mengerjakan suatu proyek perangkat lunak keberhasilan proyek harus dimulai dengan perencanaan yang benar, jika perencanaan salah maka akan menyebabkan kegagalan dan akan terjadi pembengkakan biaya. Oleh karna itu diperlukan estimasi biaya yang tepat untuk meminimalisir resiko over-estimates dalam proyek perangkat lunak. estimasi biaya pada penelitian ini menggunakan metode Function Point (FP) yang merupakan metode yang mengukur perangkat lunak dengan cara mengukur dari prespektif functional dari software yang akan dibangun. Terdapat 5 komponen untuk menghitung matrik FP yaitu External Input (EI), External Output (EO), External Inquiry (EI), Internal Logical File (ILF), External File (EIF).. Dari penelitian yang telah dilakukan, estimasi biaya pengembangan perangkat lunak INLIS LITE 3.0, maka disimpulkan apabila Perpustakaan Nasional ingin mengembangkan perangkat lunak INLIS LITE 3.0, maka dibutuhkan kurang lebih sebesar IDR 798.906.152 atau setara dengan *effort* 18255,91 orang/jam.

Kata Kunci: Estimasi Biaya, Metode *Function Points*, Pengembangan Perangkat Lunak, *Effort*

Abstract

In working on a software project the success of the project must begin with the right planning, if planning is wrong it will cause failure and cost overruns will occur. Therefore it is necessary to estimate the right cost to minimize the risk of over-estimates in software projects. Cost estimation in this study uses the Function Point (FP) method which is a method that measures software by measuring the functional perspective of the software to be built. There are 5 components to calculate the FP matrix, namely External Input (EI), External Output (EO), External Inquiry (EI), Internal Logical File (ILF), External File (EIF). From the research that has been done, the estimated cost of developing INLIS LITE 3.0 software, it is concluded that if Perpustakaan Nasional want to develop INLIS LITE 3.0 software, it will require approximately IDR 798.906.152 or equivalent to effort 18255,91 man / hour.

Keywords: Cost estimation, Function Point Method, Software Development, Effort

1. PENDAHULUAN

Dalam industri 4.0 teknologi informasi dan komunikasi semakin dibutuhkan dan berkembang pesat. Berbagai aplikasi baru banyak bermunculan untuk membantu menjalankan proses bisnis. Baik dalam bidang pendidikan, ekonomi, dan pemerintahan. Perpustakaan Nasional merupakan lembaga pemerintahan yang telah memanfaatkan teknologi untuk mendukung proses bisnis. Aplikasi perangkat lunak (software) telah dikembangkan di Perpustakaan Nasional. dalam mengembangkan aplikasi perangkat lunak maka dibutuhkan estimasi biaya sebagai acuan pengembangan aplikasi kedepanya. Dalam pengembangan perangkat lunak terdapat beberapa metode estimasi biaya yang biasa digunakan, yaitu : Metode *Function Points* yang pertamakali dikembangkan oleh A.J. Albrecht pada tahun 1983 dan resmi dinyatakan oleh *International Function Points User Group* (IFPUG), Metode *Cost Constructive Model* (COCOMO) yang telah dikembangkan oleh B. Boehm, Metode *Use Case Point* (UCP) yang dikembangkan oleh Karner pada tahun 1993, dan Logika fuzzy cara untuk memetakan suatu ruang input kedalam suatu ruang output

Menurut Renny [1], metode FP dapat digunakan sebagai metode estimasi usaha yang cukup baik. Beberapa penelitian sebelumnya telah merekomendasikan metode FP apabila jangka waktu perencanaan proyek terbatas [2], [3]. Oleh karena itu, dalam estimasi biaya perangkat lunak SIMAS ini menggunakan metode *Function Points*.

2. METODE PENELITIAN

Allan J. Albrecht merupakan pengembang metode *Function Points Analysis* sejak tahun 1983 di perusahaan IBM [4]. Metode Function Point adalah metode dengan pendekatan yang berorientasi pada fungsionalitas dan kompleksitas dalam memperkirakan ukuran perangkat lunak dan selanjutnya memperkirakan upaya dan estimasi biaya untuk pengembangan perangkat lunak [5]. Berikut tahapan-tahapan untuk melakukan estimasi biaya dengan menggunakan metode *Function Points Analysis* :

a. Menghitung *Crude Function Points* (CFP)

Dalam metode *Function Points* tersusun dari paramater utama yang memiliki bobot masing-masing Tabel 1. Menurut penelitian Albrecht [4] ada lima parameter utama yaitu EI, EO, EQ, ILF dan ELF. Dalam menghitung UFP setiap nilai dikalikan dengan faktor kali kompleksitas masing-masing Tabel 2.

Tabel 1. Parameter Menghitung Crude Function Points

Information Domain Value	Count	Weighting Factor			=	
		Simple	Average	Complex		
External Input (EI)		×	3	4	6	=
External Output (EO)		×	4	5	7	=
External Inquiry (EQ)		×	3	4	6	=
Internal Logical File (ILF)		×	7	10	15	=
External Logical File (ELF)		×	5	7	10	=
Count Total						

b. Menghitung *Value Adjustment Factor* (VAF)

VAF digunakan untuk menghitung bobot kompleksitas dari perangkat lunak. Penilaian kompleksitas memiliki bobot 0 sampai dengan 5.

- 0 = Tidak Pengaruh
- 1 = Insidental
- 2 = Moderat
- 3 = Rata-rata
- 4 = Signifikan
- 5 = Essential

Sebuah perangkat lunak memiliki 14 karakteristik. Karakteristik tersebut adalah ketetapan atau konstanta yang dibuat oleh IFPUG. Empat belas karakteristik akan dijelaskan pada Tabel 2

Tabel 2. Bobot Karakteristik Value Adjustment Factor

NO	KARAKTERISTIK	BOBOT				
		0	1	2	3	4
1	Tingkat Kompleksitas Komunikasi					
2	Tingkat Kompleksitas Pemrosesan Terdistribusi					
3	Tingkat Kompleksitas Performance					
4	Tingkat kompleksitas Konfigurasi					
5	Tingkat Frekuensi Penggunaan Software					
6	Tingkat Frekuensi Input Data					
7	Tingkat Kemudahan Pengguna Bagi User					
8	Tingkat Frekuensi Input Data					
9	Tingkat Kompleksitas Prosesing Data					
10	Tingkat Kemungkinan Penggunaan Kembali/ Reusable Kode Program					
11	Tingkat Kemudahan dalam Instalasi					
12	Tingkat Kemudahan Dalam Operasional Software (backup, Recovery, dsb)					
13	Tingkat Software dibuat untuk multi Organisasi/Perusahaan/Client					
14	Tingkat Kompleksitas dalam mengikuti Perubahan					

c. Menghitung *Function Points* (FP)

FP merupakan suatu proses melakukan perhitungan untuk mendapat nilai FP dari perangkat lunak yang akan dikembangkan.

Rumus FP:

$$FP = CP \times (0.65 + (0.01 \times VAF)) \quad (1)$$

Angka 0.65 dan 0.01 merupakan ketetapan atau konstanta yg dibutuh oleh *Function Point Internasional User Group* (IFPUG).

d. Menghitung *Lines of Code* (LoC)

Lines of code (LoC) adalah metrik perangkat lunak yang digunakan untuk mengukur ukuran program perangkat lunak dengan menghitung jumlah baris pada teks kode sumber program. LoC biasanya digunakan untuk memprediksi jumlah upaya yang akan

diperlukan untuk mengembangkan suatu program, serta untuk memperkirakan produktivitas atau upaya pemograman begitu perangkat lunak produksi.

Untuk mengubah nilai FP ke dalam LoC dengan rumusan dibawah ini:

$$LoC = FP \times 56 \quad (2)$$

*Nilai 56 adalah ketetapan atau konstanta dengan indeks bahasa pemrograman PHP.

e. Menghitung *Effort*

Nilai *Function Points* yang telah diketahui untuk perangkat lunak yang akan dibangun. Langkah selanjutnya ialah melakukan perhitungan estimasi effort usaha yang diperlukan dalam pengembangan perangkat lunak berdasarkan man/hour.

Dengan rumusan:

$$Effort = FP \times PF \quad (3)$$

Diketahui PF adalah *Productivity Factor* yang bernilai 8,2 angka ini didapatkan dari penelitian Putu Linda Primandari

f. Menghitung Pendistribusian Effort ke Aktivitas

Setiap effort yang sudah diketahui di distribusikan kedalam setiap aktivitas pengembangan perangkat lunak yang ada pada Tabel 3 tujuannya adalah untuk mencari estimasi biaya yang dibutuhkan dalam membangun sebuah perangkat lunak.

Tabel 3. Pendistribusian Effort ke Aktifitas

No	Aktivitas	Percentase (%)
1	Requirements	1,6
2	Specifications	7,5
3	Design	6
4	Implementation	52
5	Integration Testing	7
6	Acceptance & deployment	5,5
7	Project Management	3,8
8	Configuration Management	4,3
9	Quality Assurance	0,9
10	Documentation	8,4
11	Training & Support	1
12	Evaluation & Testing	2
	Total	100

Tabel % effort merupakan konstanta mutlak. Tabel Effort (man hours) merupakan perkalian antara effort × % effort. Payrate/hr didapat dari hasil perhitungan Rp. 7.000.000 standart gaji Kelly (Kelly Service, 2017) dibagi 20 hari (1 bulan = 4 minggu, 1 minggu= 5 hari, senin hingga jumat, jadi total 20 hari) kemudian dibagi 8 (1 hari = 8 jam dari jam 0.00 hingga 16.00 WIB)

3. ANALISA DAN PEMBAHASAN

Untuk mengetahui nilai estimasi biaya pengembangan perangkat lunak INLIS LITE 3.0 maka perlu dilakukan perhitungan.

Berikut tahap-tahap menghitung estimasi biaya dengan menggunakan metode *Function Point* seperti berikut ini :

a. *Crude Function Points* (CFP)

Tabel 4. Crude Function Points INLIS LITE 3.0

Komponen	Level kompleksitas									Total CFP	
	Simple			Average			Complex				
	A	B	C=A xB	D	E	F=DxE	G	H	I=GxH		
EI	513	3	1539	100	4	400	1	6	6	1945	
EO	513	4	2052	100	5	500	1	7	7	2559	
EQ	14	3	42	0	4	0	12	6	72	114	
ILF	2	7	14	0	10	0	1	15	14	14	
EIF	5	0	2	7	14	6	10	60	74		
Total									4706		

b. *Value Adjustment Factor (VAF)*

Tabel 5. Value Adjustment Factor

NO	KARAKTERISTIK	BOBOT				
		0	1	2	3	4
1	Tingkat Kompleksitas Komunikasi					
2	Tingkat Kompleksitas Pemrosesan Terdistribusi					
3	Tingkat Kompleksitas Performance					
4	Tingkat kompleksitas Konfigurasi					
5	Tingkat Frekuensi Penggunaan Software					
6	Tingkat Frekuensi Input Data					
7	Tingkat Kemudahan Pengguna Bagi User					
8	Tingkat Frekuensi Input Data					
9	Tingkat Kompleksitas Prosesing Data					
10	Tingkat Kemungkinan Penggunaan Kembali/ Reusable Kode Program					
11	Tingkat Kemudahan dalam Instalasi					
12	Tingkat Kemudahan Dalam Operasional Software (backup, Recovery, dsb)					
13	Tingkat Software dibuat untuk multi Organisasi/Perusahaan/Client					
14	Tingkat Kompleksitas dalam mengikuti Perubahan					
TOTAL VAF		22				

c. Menghitung *Function Points (FP)*

FP merupakan suatu proses melakukan perhitungan untuk mendapat nilai FP dari perangkat lunak yang akan dikembangkan. Sesuai dengan perhitungan sebagai berikut:

$$\begin{aligned} \text{FP} &= CFP \times (0.65 + (0.01 \times VAF)) \\ &= 2559 \times (0.65 + 0.01 \times 22) \\ &= 2226,33 \end{aligned}$$

d. Menghitung *Lines of Code (LoC)*

LoC biasanya digunakan untuk memprediksi jumlah upaya yang akan diperlukan untuk mengembangkan suatu program, serta untuk memperkirakan produktivitas atau upaya pemograman begitu perangkat lunak produksi.

Untuk mengubah nilai FP ke dalam LoC dengan rumusan dibawah ini:

$$\begin{aligned} \text{LoC} &= \text{FP} \times 56 \\ &= 2226,33 \times 56 \\ &= 124674,48 \end{aligned}$$

e. Menghitung *Effort*

Nilai *Function Points* yang telah diketahui untuk perangkat lunak yang akan dibangun. Langkah selanjutnya ialah melakukan perhitungan estimasi effort usaha yang diperlukan dalam pengembangan perangkat lunak berdasarkan man/hour.

$$\begin{aligned} \text{Effort} &= \text{FP} \times \text{PF} \\ &= 2226,33 \times 8,2 \\ &= 18255,91 \end{aligned}$$

f. Menghitung Pendistribusian *Effort* ke Aktivitas

Tabel 6. Distribusi Effort ke Aktivitas SIMAS

No	Aktivitas	Percentase (%)	Effort = SLoC x %	Payrate/hr (IDR)	Cost = effort x payrate (IDR)
1	Requirements	1.6	348,29696	39,773	15.237.992
2	Specifications	7.5	1632,642	39,773	71.428.088

3	Design	6	1306,1136	45,455	57.142.470
4	Implementation	52	11.319,65	45,455	353.739.100
5	Integration Testing	7	1523,7992	45,455	47.618.725
6	Acceptance & deployment	5.5	1197,2708	45,455	37.414.713
7	Project Management	3.8	827,20528	113,636	103.400.660
8	Configuration Management	4.3	936,04808	113,636	29.251.503
9	Quality Assurance	0.9	195,91704	45,455	6.122.408
10	Documentation	8.4	1828,55904	45,455	57.142.470
11	Training & Support	1	217,6856	45,455	6.802.675
12	Evaluation & Testing	2	435,3712	45,455	13.605.350
Total		100	21.768,56		798.906.152

4. KESIMPULAN

Dari penelitian yang telah dilakukan, estimasi biaya pengembangan perangkat lunak INLIS LITE 3.0, maka disimpulkan apabila bahwa Perpustakaan Nasional ingin mengembangkan perangkat lunak INLIS LITE 3.0 maka dibutuhkan kurang lebih sebesar IDR 798.906.152 atau setara dengan *effort* 18255,91 orang/jam. Penulis mengabaikan faktor-faktor eksternal terutama kesiapan insfrakstruktur dalam mengimplementasikan INLIS LITE 3.0. Oleh karena itu, perhitungan estimasi *effort* dan biaya juga disesuaikan dengan kondisi pemilik perangkat lunak

REFERENCES

- [1] R. S. Dewi, “Implementasi Metode Function Points Untuk Mengestimasi Usaha Pada Proyek Pembangunan Aplikasi Layanan Publik,” *J. Sist. Inf. Indones.*, vol. 2, no. 1, pp. 1–8, 2017.
- [2] R. S. Dewi, A. P. Subriadi, and Sholiq, “A Modification Complexity Factor in Function Points Method for Software Cost Estimation Towards Public Service Application,” in *Procedia Computer Science*, 2017.
- [3] R. S. Dewi, T. W. Andari, M. B. A. Rasyid, and R. Candra A.P., “Ekstraksi Faktor Kompleksitas Game Menggunakan Metode Function Points,” *J. Teknol. dan Sist. Inf.*, 2018.
- [4] A. J. Albrecht and J. E. Gaffney, “Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation,” *IEEE Trans. Softw. Eng.*, 1983.
- [5] A. Yhurinda, P. Putri, and A. P. Subriadi, “Software Cost Estimation Using Function Point Analysis,” *IPTEK J. Proc. Ser.*, vol. 0, no. 1, pp. 79–83, 2019.
- [6] P. L. Primandaria and Sholiq, “Effort Distribution to Estimate Cost in Small to Medium Software Development Project with Use Case Points,” in *Procedia Computer Science*, 2015.