

Estimasi Biaya Perangkat Lunak Pada Aplikasi SIBIMA Universitas XYZ dengan Menggunakan Metode Function Point

Annisa Azzar Sandhea, Sri Maulidiyah, Renny Sari Dewi *

Fakultas Teknologi Informasi dan Kreatif, Sistem Informasi, Universitas Internasional Semen Indonesia, Gresik, Indonesia
Email: ¹annisa.sandhea16@student.uisi.ac.id, ²sri.maulidiyah16@student.uisi.ac.id, ^{3,*}renny.dewi@uisi.ac.id

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

Submitted 07-01-2020; Accepted 29-01-2020; Published 15-02-2020

Abstrak

Semakin berkembangnya penggunaan komputer memicu banyaknya perangkat lunak dalam jumlah dan kegunaan yang berbeda, suatu sistem dikatakan baik ketika memulai perencanaan yang dapat memperkirakan biaya, waktu dan jumlah sumber daya yang diperlukan sebagai dasar perencanaan suatu sistem tersebut. Sehingga estimasi biaya dapat menjadi elemen dalam suatu penilaian. Penelitian menjelaskan bahwa Sistem Informasi Bimbingan Akademik (SIBIMA) yang merupakan sistem untuk menjalankan kegiatan konsultasi mahasiswa dengan pembimbing akademik di Universitas XYZ. Dari hasil Estimasi Biaya Perangkat Lunak pada SIBIMA menggunakan Function Point yang merupakan metode perkiraan ukuran perangkat lunak menggunakan pendekatan secara tidak langsung atau indirect approach untuk memperkirakan ukuran perangkat lunak dalam satuan Function Point (FP), untuk nilai Effort pada SIBIMA adalah 379,906 man/hour dan telah diketahui biaya yang dibutuhkan dalam pengembangan SIBIMA sebesar IDR 22.858.705 sehingga dapat dijadikan sebagai acuan apabila institusi pendidikan tinggi ingin mengembangkan suatu sistem perangkat lunak.

Kata Kunci: Sistem Informasi Bimbingan Akademik (SIBIMA), Metode Function Point, Estimasi Biaya Perangkat Lunak, Pengembangan Perangkat Lunak, *Effort*

Abstract

The growing development of computers triggers the amount of software in different amounts and uses. A system is said to be good when starting planning that can estimate the cost, time and amount of resources needed as a basis for planning a system. So that the estimated cost can be an element in an assessment. The research explains that the Academic Guidance Information System (SIBIMA) which is a system for carrying out student consultation activities with academic advisers at XYZ University. From the software cost estimation results at SIBIMA use Function Point which is a method of estimating software size using an indirect approach or indirect approach to estimate software size in Function Point units (FP), for Effort values on SIBIMA is 379,906 man / hour and it is known that the cost needed in developing SIBIMA is IDR 22,858,705 so that it can be used as a reference if the higher education institution wants to develop a software system.

Keywords: Academic Guidance Information System (SIBIMA), Function Point Method, Software Cost Estimation, Software Development, Effort

1. PENDAHULUAN

Perkembangan teknologi saat ini telah memberikan pengaruh yang sangat besar bagi dunia teknologi informasi dan telekomunikasi. Munculnya beragam aplikasi memberikan pilihan dalam peningkatan kinerja suatu pekerjaan [1]. Tak terkecuali dalam bidang informasi. Kini dikenal dengan istilah internet. Dengan adanya internet perubahan informasi atau kejadian terkini yang ada di dunia dapat diakses dengan cepat [2]. Di Indonesia, estimasi usaha sejatinya merupakan perencanaan proyek guna mengalokasikan sumber daya yang mereka butuhkan [3].

Saat ini Sistem Informasi Bimbingan Akademik (SIBIMA) yang berada pada website Sistem Informasi Akademik (SIKAD) Universitas XYZ digunakan untuk menjalankan kegiatan konsultasi yang diselenggarakan oleh masing-masing program studi dan telah didukung dengan aplikasi berbasis komputer untuk membantu antara pembimbing akademik dan mahasiswa dalam merencanakan studi serta membantu menyelesaikan masalah studi yang dialami, agar mahasiswa yang bersangkutan dapat menyelesaikan studinya dengan baik sesuai dengan minat dan kemampuannya.

Suatu sistem dikatakan baik ketika memulai perencanaan yang dapat memperkirakan kebutuhan biaya, waktu dan jumlah sumber daya yang diperlukan sebagai dasar perencanaan dalam pengembangan suatu sistem tersebut, sebab Pengukuran perangkat lunak merupakan hal yang sangat mendasar dalam pembuatan perangkat lunak dan daur/siklus hidup perangkat lunak tersebut. Pengukuran perangkat lunak dan metrik telah banyak digunakan untuk memutuskan bentuk perangkat lunak versi selanjutnya, dalam bentuk pengukuran ataupun penambahan fitur produk. Selain itu Pengukuran Perangkat lunak juga dapat digunakan untuk mengukur faktor kebutuhan dan budget dalam sebuah proyek perangkat lunak, seperti sumber daya manusia dan biaya [4], [5], [6].

Estimasi atau perkiraan ukuran perangkat lunak dalam proses pengembangan perangkat lunak memegang peranan penting untuk menjaga proses pengembangan tetap dalam kendali pengembang [7]. Sehingga estimasi biaya dapat menjadi elemen dalam suatu penilaian [4]. Karena salah satu penentu keberhasilan pengerjaan sebuah proyek adalah adanya estimasi biaya dan estimasi effort suatu proyek [8].

Untuk mengetahui perhitungan dalam perkiraan biaya pada Sistem Informasi Bimbingan Akademik (SIBIMA) maka di perlukannya suatu metode yang tepat, yaitu dengan menggunakan Metode Function Point. Metode Functional Point merupakan salah satu metode perkiraan ukuran perangkat lunak yang menggunakan pendekatan secara tidak langsung atau indirect approach yang memperkirakan ukuran perangkat lunak dalam satuan Function Point (FP) [9]. Metode Function Point Analysis (FPA) memperkirakan ukuran perangkat lunak dalam satuan Function Point (FP) yang

didapatkan dari lima parameter utama yaitu: External Input (EI), External Output (EO), External Inquiry (EQ), Internal Logical File (ILF) serta External Interface File (EIF) yang dikelompokkan dalam bobot Low, Average, dan High [10], yang digunakan untuk menghitung bobot nilai dari komponen-komponen function points yang dikaitkan dengan suatu sistem yang akan dikembangkan. Sebelum menghitung nilai FP perlu adanya perhitungan nilai total Relative Complexity Adjustment Factor (RCAF). RCAF dihitung berdasarkan pada keseluruhan kompleksitas sistem. Cara menghitung RCAF adalah dengan menggunakan 14 (empat belas) GSC (General System Characteristic), dimana masing-masing GSC berskala 0 (nol) sampai 5 (lima). Skala 0 (nol) menunjukkan tidak adanya pengaruh dan skala 5 (lima) menunjukkan adanya pengaruh yang luas terhadap keseluruhan proyek [11].

Tujuan dari perhitungan estimasi biaya perangkat lunak pada Sistem Informasi Bimbingan Akademik (SIBIMA) yaitu sebagai pedoman awal dalam perhitungan rencana untuk mendapatkan perkiraan biaya yang dikeluarkan pada sistem yang akan dikembangkan. Dengan demikian akurasi estimasi biaya suatu proyek merupakan salah satu penentu keberhasilan suatu proyek [12]. Dengan demikian akurasi estimasi biaya suatu proyek merupakan salah satu penentu keberhasilan suatu proyek [13], [14].

Dari penjelasan diatas, penulis menggunakan Metode *Function Point* untuk melakukan estimasi biaya pada pengembangan software FAS (*Financing Analysis System*) yang digunakan pada perusahaan perbankan BPRS Lantabur Tebuireng. *Function Point* sendiri adalah suatu pengukuran standar dari perangkat lunak untuk kuantifikasi fungsionalitas yang ditawarkan program ke pengguna. Metode perhitungan *Function Point* berdasarkan pada identifikasi fungsi yang dilakukan oleh sistem dan memberikan tingkat kompleksitas pada setiap fungsi dan dianggap lebih cepat karena tidak memerlukan hasil analisis sistem dalam bentuk skema pelaporan penggunaan naratif dan konsep basis data.

2. METODE PENELITIAN

Penelitian ini bertujuan untuk mengetahui estimasi biaya dari pengembangan software SIBIMA (Sistem Bimbingan Akademik) yang terdapat pada universitas xyz. Secara garis besar terdapat metode yang perlu dilakukan untuk melakukan penelitian ini yang dijelaskan pada Gambar 1.



Gambar 1. Metode Penelitian

2.1 Studi Lapangan

Metode penelitian dimulai dari studi lapangan yang dilakukan dengan mengamati masalah masalah aktual dalam proyek- proyek pengembangan perangkat lunak (*software*) yang ditujukan pada SIBIMA di Universitas XYZ. Dengan mengumpulkan data-data maupun informasi yang dibutuhkan untuk melakukan penelitian dari sistem perangkat lunak tersebut

2.2 Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan wawancara untuk mendapatkan data kualitatif berupa informasi terkait kebutuhan kebutuhan fungsionalitas sistem informasi serta dilakukan dengan mengamati masalah masalah pada proyek pengembangan perangkat lunak (Software) pada SIBIMA yang telah dilakukan sebelumnya. Pada tahapan ini, dilakukan juga studi literatur dengan membaca jurnal penelitian terkait dengan metode Function Point yang dibutuhkan dalam pengembangan aplikasi. Serta dilakukan pengumpulan dari teori-teori maupun informasi guna mendukung penelitian yang sumbernya dari buku, jurnal penelitian dan internet.

2.3 Analisis dan Pemodelan Dengan Metode FP

Data hasil pengeumpulan data kemudian dianalisis untuk membuat model estimasi menggunakan metode function point untuk mengetahui perhitungan dalam memperkirakan biaya pada SIBIMA di unversitas XYZ. Setelah diketahui hasil perhitungannya maka dapat dijadikan sebagai acuan apabila institusi pendidikan tinggi ingin mengembangkan suatu sistem perangkat lunak. Berikut ini adalah langkah-langkah untuk melakukan perhitungan estimasi biaya perangkat lunak dengan menggunakan metode function point:

a. Menghitung CFP (*Crude Function Points*)

Jumlah dari komponen fungsional sistem pertama kali diidentifikasi dan dilanjutkan dengan mengevaluasi kuantitasi bobot kerumitan dari tiap komponen tersebut. Pembobotan tersebut kemudian dijumlahkan dan menjadi angka CFP. Perhitungan CFP melibatkan 5 tipe komponen sistem software berikut :

1. Jumlah macam aplikasi input.
2. Jumlah macam aplikasi output
3. Jumlah macam aplikasi query online – aplikasi ini berhubungan dengan query terhadap data yang tersimpan
4. Jumlah macam file/Tabel logic yang terlibat

Jumlah macam interface eksternal – output atau input yang dapat berhubungan dengan komputer lewat komunikasi data, CD, disket, dan lain-lain

b. Menghitung *RCAF (Relative Complexity Adjustment Factor)*

RCAF berfungsi untuk menghitung kesimpulan kompleksitas dari sistem software dari beberapa subyek karakteristik. Penilaian berskala 0-5 diberikan pada tiap subyek yang paling berpengaruh terhadap usaha pengembangan yang dibutuhkan. Tabel penilaian RCAF dapat dilihat pada Tabel 1.

Tabel 1. Faktor RCAF

No	Karakteristik	Deskripsi	Skor
1	Tingkat kompleksitas Komunikasi Data	Tingkat kebutuhan komunikasi langsung antara aplikasi dan processor	(0/1/2/3/4/5)
2	Tingkat kompleksitas Pemrosesan Terdistribusi	Tingkat kebutuhan transfer data antara komponen-komponen aplikasi	(0/1/2/3/4/5)
3	Tingkat kompleksitas Performance	Tingkat <i>response time</i> dan <i>throughput</i> yang perlu dipertimbangkan dalam pengembangan aplikasi.	(0/1/2/3/4/5)
4	Tingkat kompleksitas Konfigurasi	Tingkat kebutuhan dimana setting konfigurasi komputer berpengaruh terhadap pengembangan aplikasi.	(0/1/2/3/4/5)
5	Tingkat Frekuensi Penggunaan Software	Tingkat kecepatan transaksi bisnis yang berpengaruh terhadap pengembangan aplikasi.	(0/1/2/3/4/5)
6	Tingkat Frekuensi Input Data	Tingkat kebutuhan peng-input-an data secara interaktif.	(0/1/2/3/4/5)
7	Tingkat Kemudahan Penggunaan Bagi User	Tingkat kemudahan penggunaan aplikasi.	(0/1/2/3/4/5)
8	Tingkat Frekuensi Update Data	Tingkat kebutuhan ILF diupdate secara online.	(0/1/2/3/4/5)
9	Tingkat Kompleksitas Prosesing Data	Tingkat kesulitan logika proses yang mempengaruhi proses development	(0/1/2/3/4/5)
10	Tingkat Kemungkinan Penggunaan Kembali/Reusable Kode Program	Tingkat kebutuhan aplikasi dan kode program aplikasi dirancang dan dikembangkan untuk bisa digunakan pada aplikasi lain.	(0/1/2/3/4/5)
11	Tingkat Kemudahan Dalam Instalasi	Tingkat kemudahan konversi ke sistem baru yang berpengaruh pada proses development.	(0/1/2/3/4/5)
12	Tingkat Kemudahan operasional software (backup, recovery, dsb)	Tingkat kemudahan aplikasi dalam aspek-aspek operasional, seperti <i>start-up</i> , <i>backup</i> , dan proses <i>recovery</i> .	(0/1/2/3/4/5)
13	Tingkat Software dibuat untuk multi organisasi/perusahaan/client	Tingkat kebutuhan aplikasi dapat dioperasionalkan pada lingkungan <i>hardware</i> dan <i>software</i> yang berbeda-beda.	(0/1/2/3/4/5)
14	Tingkat kompleksitas dalam mengikuti perubahan/fleksibel	Tingkat kemudahan aplikasi untuk modifikasi logika proses maupun struktur data.	(0/1/2/3/4/5)

Keterangan Skor:

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

c. Menghitung Nilai FP (*Function Point*)

Pada tahapan ini dapat dilakukan perhitungan untuk mendapatkan nilai FP dari aplikasi SIBIMA yang akan dikembangkan dengan menggunakan rumus dari nilai FP sendiri yang akan dijelaskan pada bab pembahasan

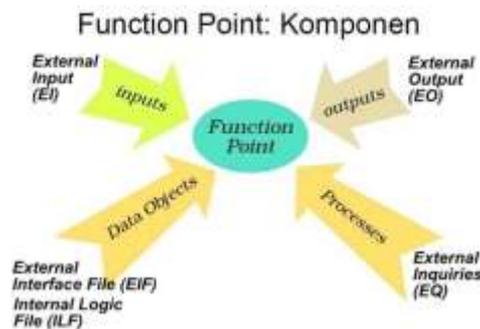
- d. Menghitung Nilai *Effort*
 Pada tahapan ini dapat dilakukan perhitungan dari hasil nilai FP dan angka 8,2 yang merupakan angka ketetapan yang dibuat oleh International Function Point Users Group (IFPUG)
- e. Mengkalkulasi Nilai Kloc
 Pada tahapan ini dapat dilakukan perhitungan dari hasil nilai FP dan angka 56 yang merupakan angka ketetapan yang dibuat oleh International Function Point Users Group (IFPUG)
- f. Menghitung Nilai *Final Effort*
 Pada tahapan *Final Effort* dilakukan perhitungan dari Kloc ke *Effort* yang dapat dilihat pada rumus (5). angka 18 merupakan angka ketetapan yang dibuat oleh International Function Point Users Group (IFPUG)
- g. Mendistribusikan Nilai *Final Effort* ke Presentase Tiap Aktivitas
 Setelah dilakukan beberapa tahapan perhitungan didapatkan *Final Effort* ke presentase tiap aktivitas dengan jumlah presentase dan rincian pada Tabel 4 dan Tabel 5

3. ANALISA DAN PEMBAHASAN

Pada dasarnya pengukuran merupakan kegiatan penentuan angka bagi suatu objek yang bertujuan untuk melakukan perkiraan dan untuk menyatakan kualitas produk. Pada pengembangan perangkat lunak, perlu adanya estimasi atau perkiraan ukuran suatu perangkat lunak yang akan dikembangkan untuk menentukan dan mengetahui berapa besar effort atau usaha yang akan dikeluarkan untuk mengembangkan perangkat lunak.

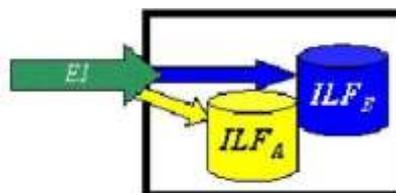
3.1 *Function Point*

Salah satu metode yang digunakan dalam mengetahui perhitungan estimasi biaya perangkat lunak pada SIBIMA adalah *Function Point*. *Function Point* diperkenalkan pertama kali oleh Allan J. Albercht pada tahun 1979 lalu perbaikan metode ini dilakukan pada tahun 1984 oleh Albrecht. Sejak berdirinya International Function Point Counting Practices (IFPUG) pada tahun 1986, menerbitkan beberapa versi dari *Function Point Counting Practices Manual* [15]. Metode *Function Point* adalah sebuah metode yang melakukan pendekatan berorientasi pada fungsionalitas dan kompleksitas perangkat lunak dalam mengestimasi size perangkat lunak yang selanjutnya digunakan untuk mengestimasi effort dan estimasi biaya yang diperlukan untuk pengembangan perangkat lunak [7], [16]. Pengertian lain dari *Function Point* (FP) merupakan sebuah teknik terstruktur dalam memecah suatu sistem perangkat lunak menjadi komponen yang lebih kecil dan menetapkan beberapa karakteristik dari sebuah software sehingga mudah untuk dianalisis. Gunanya (1) untuk memprediksi atau memperkirakan biaya yang dibutuhkan dalam sistem perangkat lunak, (2) untuk memprediksi jumlah error yang mungkin akan terjadi, dan (3) memperkirakan jumlah komponen pada sistem yang diimplementasikan [5].



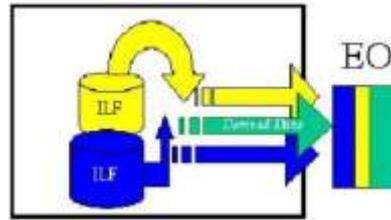
Gambar 2. Komponen *Function Point*

Function Point dihitung dari 5 parameter [19] utama yaitu :



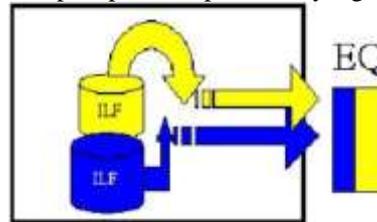
Gambar 3. *External Input*

Pada Gambar 2. Menunjukkan proses *External Input* (EI) yaitu parameter pertama yang merupakan proses dasar yang memproses data dan informasi kontrol yang datang dari luar batasan aplikasi, berkaitan dengan interface yang lakukan pengguna/user dalam memasukan data pada aplikasi. Perhitungan input harus dibedakan dari Inquiry, karena Inquiry akan dihitung secara terpisah



Gambar 4. External Output

Pada Gambar 3. Menunjukkan proses Tipe Output atau External Output (EO) yaitu parameter kedua yang merupakan proses dasar dimana hasil data dilewatkan dari dalam keluar dari batasan aplikasi, berkaitan dengan output yang dihasilkan dari aplikasi untuk pengguna/user yang dapat berupa laporan di print atau yang ditampilkan pada layar.



Gambar 5. External Inquiry

Pada Gambar 4. Menunjukkan proses Tipe Query/Search/View atau External Inquiry (EQ) yaitu parameter ketiga yang berfungsi untuk menyediakan informasi ke user melalui pengambilan atau pemrosesan data atau informasi kontrol dari ILF/EIF, berkaitan dengan query terhadap data yang tersimpan. Selanjutnya parameter yang keempat yaitu Tipe File/Tabel/Database atau Internal Logical Files (ILF) adalah kelompok data atau kelompok informasi kontrol yang digunakan dalam aplikasi. Peran utamanya yaitu menyimpan data yang dipelihara oleh satu atau lebih proses dalam aplikasi, berkaitan dengan logic penyimpanan data yang dapat berupa Tabel, file atau semacam database relational. Selanjutnya parameter yang kelima yaitu Tipe Interface Eksternal atau External Interface Files (EIF) adalah kelompok data berelasi atau informasi kontrol yang dirujuk oleh aplikasi, tapi dipelihara oleh aplikasi lain, berkaitan dengan komunikasi data pada perangkat/mesin yang lain. Sebuah EIF yang dihitung untuk sebuah aplikasi harus merupakan ILF di aplikasi lain. Tahap perhitungan pada metode function point adalah dengan menggunakan rumus sebagai berikut :

$$CFP = \sum(EI, EO, EQ, ILF, ELF) \times \text{Bobot} \quad (1)$$

$$FP = CFP \times (0,65 + (0,01 \times RCAF)) \dots \dots \dots (n) \quad (2)$$

$$\text{Effort} = FP \times 8,2 \quad (3)$$

$$FP \text{ ke Kloc} = FP \times 56 \quad (4)$$

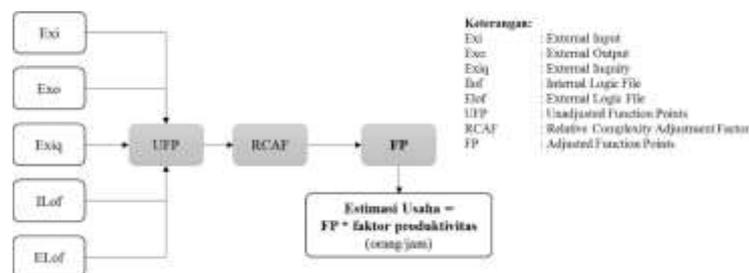
$$\text{Kloc ke Effort} = \frac{\text{Effort}}{56} \times \text{Hari Kerja} \times \text{Jam Kerja} \quad (5)$$

18

$$\text{Effort} = \% \times \text{Final Effort} \quad (6)$$

$$\text{Cost} = \text{Effort} \times \text{Effort Payrate/hr} \quad (7)$$

Secara sistematis, pada Gambar 5. Menjelaskan tentang tahapan-tahapan perhitungan metode FP sampai dengan estimasi dan distribusi usaha dalam satuan orang/jam. Dari perhitungan estimasi usaha tersebut, penulis membandingkannya dengan usaha actual.



Gambar 6. Metode Function Point

3.2 Menghitung CFP (Crude Function Point)

Pada tahap ini *Crude Function Point* (CFP) digunakan untuk menghitung bobot nilai dari komponen-komponen *function points* yang dikaitkan dengan suatu sistem yang akan dikembangkan. Pada Tabel 1. telah dilakukan pengelompokan dalam bobot *Low*, *Average*, dan *High*. Pembobotan tersebut kemudian dijumlahkan dan menjadi angka CFP [11].

Tabel 2. Bobot Kompleksitas

Komponen	Level Kompleksitas		
	<i>Low</i>	<i>Average</i>	<i>High</i>
<i>External Inputs</i> (EI)	3	4	6
<i>Internal Outputs</i> (EO)	4	5	7
<i>External Inquiries</i> (EQ)	3	4	6
<i>Internal Logical Files</i> (ILF)	7	10	15
<i>External Interface File</i> (EIF)	5	7	10

Setelah dilakukan pembobotan yang berdasarkan kompleksitasnya, maka dapat dilakukan perhitungan *Crude Function Point* (CFP) dengan menggunakan rumus (1). Berdasarkan pada Tabel 3. maka dapat diketahui nilai total dari *Crude Function Point* (CFP) sebesar 41.

Tabel 3. Perhitungan CFP

Tipe Komponen	Level Kompleksitas									Total
	Sederhana			Menengah			Kompleks			
	Jumlah	Bobot	Point	Jumlah	Bobot	Point	Jumlah	Bobot	Point	
EI	6	3	18	3	4	12	0	6	0	30
EO	2	4	8	0	5	0	0	7	0	8
EQ	1	3	3	0	4	0	0	6	0	3
ILF	0	7	0	0	10	0	0	15	0	0
ELF	0	5	0	0	7	0	0	10	0	0
Total (CFP = UFP)										41

3.3 Menghitung RCAF (*Relative Complexity Adjustment Factor*)

Relative Complexity Adjustment Factor (RCAF) digunakan untuk menghitung bobot kompleksitas dari suatu sistem yang berdasarkan 14 karakteristik *software*. Terdapat penilaian kompleksitas yang memiliki skala dari 0 sampai 5.

0 = Tidak Pengaruh 1 = Insidental

2 = Moderat

3 = Rata-rata

4 = Signifikan

5 = Essential

Berdasarkan pada Tabel 4. maka dapat diketahui nilai total RCAF sebesar 48, yang didapat dari total skala dalam 14 karakteristik tersebut.

Tabel 4. Perhitungan RCAF

No	Karakteristik	Skala
1	Tingkat keandalan dan pemulihan data	4
2	Tingkat komunikasi data	2
3	Tingkat distribusi pemrosesan data	4
4	Tingkat capaian kerja	5
5	Tingkat konfigurasi lingkungan	3
6	Tingkat kapasitas transaksi	2
7	Tingkat efisiensi pengguna	5
8	Tingkat pembaruan file induk	1
9	Tingkat pembaruan <i>real time</i> online	4
10	Tingkat penggunaan kembali	3
11	Tingkat kemudahan instalasi	4
12	Tingkat kemudahan penggunaan	5
13	Tingkat variasi organisasi pengguna	4
14	Tingkat kerentanan terhadap perubahan	2
TOTAL RCAF		48

Setelah mengetahui nilai *Crude Function Point* (CFP) pada Tabel 3. dan nilai *Relative Complexity Adjustment Factor* (RCAF) pada Tabel 4. maka langkah selanjutnya menentukan nilai *Function Point* (FP)

3.4 Menghitung *Function Point* (FP)

Function Point (FP) yaitu proses melakukan perhitungan untuk mendapatkan nilai *Function Point* dari suatu sistem yang akan dikembangkan. Untuk menghitung *Function Point* (FP) maka dapat menggunakan rumus (2). Angka 0,65 dan 0,01 merupakan angka yang konstanta atau ketetapan yang dibuat oleh *International Function Point Users Group* (IFPUG). Berikut perhitungan dalam menentukan nilai FP :

$$\begin{aligned} \text{FP} &= \text{CFP} \times (0,65 + (0,01 \times \text{RCAF})) \\ &= 41 \times (0,65 + (0,01 \times 48)) \\ &= 41 \times (0,65 + 0,48) \\ &= \mathbf{46,33} \end{aligned}$$

Jadi, nilai FP pada SIBIMA adalah 46,33

3.5 Menghitung Nilai *Effort*

Untuk menghitung *Effort*, maka dapat menggunakan rumus (3). Angka 8,2 merupakan angka yang konstanta atau ketetapan yang dibuat oleh *International Function Point Users Group* (IFPUG). Berikut perhitungan dalam menentukan nilai *Effort* :

$$\begin{aligned} \text{Effort} &= \text{FP} \times 8,2 \\ &= 46,33 \times 8,2 \\ &= \mathbf{379,906 \text{ man/hour}} \end{aligned}$$

Jadi, nilai *Effort* pada SIBIMA adalah 379,906 *man/hour*

3.6 Menghitung Nilai FP ke Kloc

Untuk menghitung Kloc, maka dapat menggunakan rumus (4). Angka 56 merupakan angka yang konstanta atau ketetapan yang dibuat oleh *International Function Point Users Group* (IFPUG). Berikut perhitungan dalam menentukan nilai Kloc:

$$\begin{aligned} \text{FP ke Kloc} &= \text{FP} \times 56 \\ &= 46,33 \times 56 \\ &= \mathbf{2.594,48} \end{aligned}$$

Jadi, nilai Kloc pada SIBIMA adalah 2.594,48

3.7 Menghitung Nilai *Final Effort*

Untuk menghitung *Final Effort*, maka menggunakan rumus (5). Angka 18 merupakan angka yang konstanta atau ketetapan yang dibuat oleh *International Function Point Users Group* (IFPUG). Berikut perhitungan dalam menentukan nilai *Final Effort*:

$$\begin{aligned} \text{Kloc ke Effort} &= \frac{\text{Effort}}{18} \times 22 \text{ Hari Kerja} \times 8 \text{ Jam Kerja} \\ &= \frac{4633}{18} \times 22 \times 8 \\ &= \mathbf{453,004 \text{ man/hour}} \end{aligned}$$

Jadi, nilai *Final Effort* pada SIBIMA adalah 453,004 *man/hour*

3.8 Mendistribusikan Nilai *Effort* Ke Aktivitas

Pada Tabel 4. dihasilkan sebuah presentase tiap aktivitas yang merupakan hasil dari perhitungan final effort yang sudah diketahui dan di distribusikan ke dalam setiap aktivitas pengembangan perangkat lunak. Yang bertujuan untuk mencari estimasi biaya yang dibutuhkan dalam mengembangkan SIBIMA. Untuk menghitung *Effort* pada Tabel 4. maka dapat menggunakan rumus (6).

Contoh perhitungan :

$$\begin{aligned} \text{Requirements} &= 1.60 \% \times 453,004 = 7,248 \\ \text{Angka dalam persentase (\%)} &\text{ didapat dari penelitian terdahulu.} \end{aligned}$$

Sedangkan untuk menghitung *Cost* pada Tabel 5. maka dapat menggunakan rumus (7).

Contoh perhitungan :

$$\begin{aligned} \text{Requirements} &= 7,248 \times 39.773 \\ &= \mathbf{288.276} \end{aligned}$$

Nilai *EffortPayrate/hr* merupakan standart gaji oleh kelly service. Dari perhitungan pada Tabel 4. maka dapat diketahui biaya keseluruhan yang harus disiapkan dalam membangun Sistem Bimbingan Akademik (SIBIMA) sebesar IDR 22.858.705

Tabel 5. Presentase Tiap Aktivitas

No	Activities	%	Effort	EffortPayrate/hr	Cost
1	Requirements	2%	7,248071111	IDR 39.773	IDR 288.276
2	Spesifications	8%	33,97533333	IDR 39.773	IDR 1.351.291
3	Design	6%	27,18026667	IDR 45.455	IDR 1.235.467
4	Implementation	52%	235,5623111	IDR 45.455	IDR 10.707.376
5	Integration Testing	7%	31,71031111	IDR 45.455	IDR 1.441.378
6	Acceptance & Deployment	6%	24,91524444	IDR 45.455	IDR 1.132.511
7	Project Management	4%	17,21416889	IDR 113.636	IDR 1.956.155
8	Configuration Management	4%	19,47919111	IDR 113.636	IDR 2.213.544
9	Quality Assurance	1%	4,07704	IDR 45.455	IDR 185.320
10	Documentation	8%	38,05237333	IDR 45.455	IDR 1.729.653
11	Training & Support	1%	4,530044444	IDR 45.455	IDR 205.911
12	Evaluation and Testing (Warranty)	2%	9,060088889	IDR 45.455	IDR 411.822
TOTAL		100 %	453,004		IDR 22.858.705

4. IMPLEMENTASI

Pada tahap implementasi menjelaskan mengenai biaya keseluruhan yang disesuaikan dengan masing-masing pekerja dalam pengembangan perangkat lunak SIBIMA dengan aktivitas yang dilakukannya. Pada Tabel 6. dapat diketahui jumlah rincian seluruh presentase dan biaya yang didapatkan pada setiap bagian tenaga kerja, meliputi Business Analyst, Programmer, Tester, Project Manager, serta Dokumentator yang memiliki aktivitas pada setiap tenaga kerja.

Tabel 6. Presentase Tenaga Kerja

Tenaga Kerja	Activities	%	Cost Activities	Total Cost Tenaga Kerja
Business Analyst	Requirement	7%	IDR 1.639.567	IDR 1.639.567
	Spesification			
Programmer	Design	52%	IDR 11.942.843	IDR 11.942.843
	Implementation			
Tester	Intregation Testing	12%	IDR 2.759.209	IDR 2.759.209
	Quality Assurance			
Project Manager	Acceptance and Deployment	18%	IDR 4.169.700	IDR 4.169.700
	Project Management			
Dokumentator	Configuration Management	10%	IDR 2.347.386	IDR 2.347.386
	Documentation			
TOTAL		100%	IDR. 22.858.705	IDR. 22.858.705

5. KESIMPULAN

Analisis Function Point dapat digunakan untuk estimasi waktu dan biaya dalam pengerjaan suatu proyek dan juga dapat digunakan untuk mengukur aplikasi yang dikerjakan, atau dalam pengembangan dan pengukuran merupakan faktor penting untuk menentukan produktivitas. Berdasarkan penelitian yang dilakukan bahwa estimasi perangkat lunak dengan *function point* mampu menampilkan *effort* dan estimasi waktu. Estimasi perangkat lunak telah diuji dengan menghasilkan nilai *effort* dari semua fungsi yang ada, yaitu didapatkan dari tahapan perhitungan pada metode function point.

Dari tahapan diatas dapat disimpulkan bahwa untuk menghitung menggunakan metode function point terlebih dahulu menentukan tipe penghitungan FP, apakah pembangunan, perbaikan, atau pemeliharaan kemudian menentukan batas-batas aplikasi lalu mengidentifikasi dan menghitung serta menjumlahkan nilai untuk mencari nilai FP yang belum di-adjust *Unadjusted Function Point* (UFP) atau *Crude Function Point* (CFP), selanjutnya menghitung *effort*, menghitung Kloc, dan menghitung *Final Effort*, dan terakhir mendistribusikan *Final Effort* ke Presentase Tiap Aktivitas. Berikut merupakan hasil dari perhitungan untuk mengetahui biaya yang akan dikeluarkan, meliputi nilai CFP sebanyak 41, nilai RCAF sebanyak 48, dari hasil CFP dan RCAF maka dapat diketahui nilai FP sebanyak 46,33, nilai *effort* 379,906 *man/hour*, nilai Kloc sebanyak 2.594,48, dan nilai Kloc ke *effort* sebanyak 453,004 *man/hour*. Dengan menggunakan analisis *function point* berdasarkan dari hasil dan pembahasan, maka dapat diambil kesimpulan bahwa dalam perhitungan estimasi biaya perangkat lunak pada

Sistem Informasi Bimbingan Akademik (SIBIMA) di Universitas XYZ diketahui nilai *Effort* adalah 379,906 *man/hour* dan biaya yang dibutuhkan sebesar IDR 22.858.705 sehingga dapat dijadikan sebagai acuan apabila institusi pendidikan tinggi ingin mengembangkan suatu sistem perangkat lunak.

Dalam penelitian ini telah dijelaskan secara detail metode pengukuran estimasi perangkat lunak berbasis *Function Point* (FP). Metode FP telah diaplikasikan untuk menghitung estimasi biaya perangkat lunak pada aplikasi berbasis web. Aplikasi ini termasuk aplikasi berukuran kecil karena dapat dilihat dari hasil yang diperoleh menunjukkan keakurasian yang rendah. Penelitian ini menjadi pembelajaran bagi penulis untuk mengembangkan aplikasi-aplikasi lain dikemudian hari. Dengan menghitung nilai estimasi terlebih dahulu maka memungkinkan untuk menghitung estimasi sumber daya yang dibutuhkan, lama pengerjaan serta biaya pengembangan sehingga kegagalan dan kerugian proses pengembangan dapat dihindari.

UCAPAN TERIMA KASIH

Dalam perhitungan estimasi biaya perangkat lunak ini, kami menyadari bahwa pengerjaan jurnal ini tidak akan terselesaikan jika tidak ada dukungan, bimbingan, dan motivasi dari pihak terkait. Oleh karena itu, kami mengucapkan puji syukur dan terimakasih kepada Tuhan Yang Maha Esa serta Ibu Renny Sari Dewi, S.Kom., M.Kom. selaku dosen pembimbing yang telah menyediakan waktu, tenaga untuk menyelesaikan jurnal ini dan pikiran untuk mengarahkan kami dalam penyusunan jurnal yang berjudul “Estimasi Biaya Perangkat Lunak Pada Aplikasi SIBIMA Universitas XYZ Dengan Menggunakan Metode *Function Point*”. Dengan segala hormat kami mengucapkan banyak terimakasih kepada semua pihak yang membantu.

REFERENCES

- [1] C. Riyana, “Teknologi Informasi dan Komunikasi,” *Pus. Perbukuan Kementrian Pendidik. Nas.*, 2010.
- [2] Johan @ Eddy Luanan, “Perkembangan, Cabaran Dan Aplikasi Teknologi Maklumat Dalam Pengajaran Dan Pembelajaran,” *Konf. Pendidik. Univ. Teknol. MARA – Univ. Pendidik. Indones. Bandung*, 2011.
- [3] R. S. Dewi, A. P. Subriadi, and S. Sholiq, “Use Case Point - Activity-Based Costing: Metode Baru untuk Mengestimasi Biaya Pengembangan Perangkat Lunak,” *Sisfo*, vol. 05, no. 03, 2015.
- [4] F. Point, “Pengukuran Fungsionalitas Perangkat Lunak Menggunakan Metode *Function Point* Berdasarkan Dokumentasi Desain,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 7, no. 2, pp. 111–120, 2013.
- [5] R. M. Fikri and J. Sekarsari, “Analisis Estimasi Biaya Proyek Peningkatan Jalan Beton Di Kabupaten Tangerang Dengan Metode *Cost Significant Model MK-337 MK-338*,” *Semin. Nas. Tek. Sipil V Tahun 2015 – UMS*, 2015.
- [6] Andi asnur pranata MH, “Perbandingan Estimasi Anggaran Biaya Antara Metode,” *Proceeding PESAT*, 2011.
- [7] R. Saptono and R. Angrainingsih, “Development of Software Size Estimation Application using *Function Point Analysis (FPA)* Approach with Rapid Application Development (RAD),” *ITsmart J. Ilm. Teknol. dan Inf.*, vol. 5, no. 2, pp. 96–103, 2016.
- [8] I. Bisnis, “Penggunaan model,” vol. 2006, pp. 337–358, 2006.
- [9] Sholiq, R. S. Dewi, and A. P. Subriadi, “A Comparative Study of Software Development Size Estimation Method: UCPabc vs *Function Points*,” in *Procedia Computer Science*, 2017.
- [10] N. Balaji, N. Shivakumar, and V. V. Ananth, “Software cost estimation using *function point* with non algorithmic approach,” *Glob. J. Comput. Sci. Technol.*, vol. 13, no. 8, 2013.
- [11] R. Sarno, J. L. Buliali, and S. Maimunah, “Pengembangan Metode Analogy Untuk Estimasi Biaya Rancang Bangun Perangkat Lunak,” *MAKARA Technol. Ser.*, vol. 6, no. 2, pp. 46–55, 2010.
- [12] A. Maddepungeng, I. Suryani, and A. Mawardani, “ESTIMASI BIAYA PADA PROYEK PERUMAHAN (Studi Kasus Proyek Pembangunan Citra Serang Residence),” *J. Fondasi Jur. Tek. Sipil Univ. Sultan Ageng Tirtayasa*, 2016.
- [13] A. Bakar, “Estimasi Biaya Dengan Menggunakan “*Cost Significant Model*” Pada Pekerjaan Jembatan Rangka Baja Di Proyek Pembangunan Jalan Lintas Selatan Provinsi Jawa Timur,” *Tek. Sipil Untag Surabaya*, 2014.
- [14] D. Longstreet, “*Function Points Analysis Training Course*,” *Longstreet Consult. Inc. Accessed*, vol. 2, p. 15, 2005.
- [15] 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.
- [16] C. J. Lokan, “*Function Points*,” *Advances in Computers*. 2005.