

# Estimasi Biaya Software FAS (Financing Analysis System) Menggunakan Metode Function Point (Studi Kasus Pada PT BPRS Lantabur Tebuireng)

Juliana Kristi, Siti Nur Aisah, Renny Sari Dewi

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

Email: <sup>1</sup>\*juliana.kristi16@student.uisi.ac.id, <sup>2</sup>siti.aisah16@student.uisi.ac.id, <sup>3</sup>renny.dewi@uisi.ac.id

Email Penulis Korespondensi juliana.kristi16@student.uisi.ac.id

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

## Abstrak

Estimasi biaya perangkat lunak merupakan proses memprediksi usaha pengembangan perangkat lunak. Input dasar estimasi biaya perangkat lunak adalah metrik pengukuran. Proyek perangkat lunak sering mengalami penundaan, over budget, dan tidak selesai karena kegagalan memperkirakan biaya pengembangan perangkat lunak. PT BPRS (Bank Perkreditan Rakyat Syariah) Lantabur Tebuireng menentukan perkiraan biaya berdasarkan jumlah sumber daya manusia, fitur yang dibutuhkan, dan dana yang dimiliki. Penelitian ini menjelaskan estimasi biaya pada perangkat lunak FAS (*Financing Analysis System*) pada PT BPRS Lantabur Tebuireng menggunakan metode *Function Point*. *Function Point* merupakan metode pengukuran fungsionalitas perangkat lunak berdasarkan tipe fungsi pengguna yaitu *External Input*, *External Output*, *External Inquire*, *Internal Logic File*, dan *External Interface File* serta perhitungan teknis dari pengembangan perangkat lunak. Hasil dari penelitian FAS (*Financing Analysis System*) membutuhkan biaya sekitar Rp 94.797.120.

**Kata Kunci:** Analisis Function Point, Estimasi Perangkat Lunak, Estimasi Biaya, Usaha, Financing Analysis System, Perbankan

## Abstract

Software cost estimation is the process of predicting software development efforts. The basic input of software cost estimation is the measurement metric. Projects often experience delays, over-budget, and are not completed due to failure to estimate software development costs. PT BPRS (Bank Perkreditan Rakyat Syariah) Lantabur Tebuireng determines the estimated cost based on the amount of human resources, features needed, and funds owned. This study explains the estimated costs of the FAS (*Financing Analysis System*) software at PT BPRS Lantabur Tebuireng using the *Function Point* method. *Function Point* is a method of measuring software functionality based on the type of user function that is *External Input*, *External Output*, *External Inquire*, *Internal Logic File*, and *External Interface File* as well as technical calculations of software development. The final of the FAS (*Financing Analysis System*) study cost around IDR 94,797,120.

**Keywords:** Function Point Analysis, Software Estimates, Cost Estimates, Effort, Financing Analysis System, Banking

## 1. PENDAHULUAN

Tahap perencanaan dan penyusunan yang benar merupakan tahapan awal dalam keberhasilan suatu proyek sistem informasi. Hal tersebut perlu disadari oleh beberapa pelaku bisnis dalam segala bidang. Kegagalan suatu proyek dapat terjadi karena memiliki perencanaan yang buruk. Salah satu aktifitas dari perencanaan proyek perangkat lunak yaitu estimasi sumber daya, biaya serta jadwal proyek. Dalam membangun perangkat lunak sebaiknya dimulai dengan perencanaan proyek [4].

PT. BPRS (Bank Pembiayaan Rakyat Syariah) Lantabur atau yang lebih dikenal dengan Bank Syariah Lantabur Tebuireng merupakan Bank yang menjalankan kegiatannya berdasarkan Prinsip Syariah. Nama “Lantabur” sendiri bermakna “tidak akan merugi”, makna ini sekaligus menjadi doa penyemangat dalam pengelolaan perusahaan. Dalam perkembangan selanjutnya, didorong semangat kebersamaan untuk membangun ekonomi. Untuk mewujudkan makna Bank dalam sebuah proyek maka diperlukan estimasi biaya perangkat lunak agar membantu dalam efisiensi biaya.

Ketika estimasi dilakukan, perlu dilakukan prediksi pada masa depan serta ketidakpastian yang akan dilalui dalam menjalankan sebuah proyek perangkat lunak [2]. Estimasi biaya perangkat lunak dan usaha proyek mempunyai kesulitan tersendiri karena karakteristik-karakteristik perangkat lunak yang berbeda dengan proyek fisik. Kesulitan-kesulitan yang sering dihadapi dalam estimasi biaya perangkat lunak sangat berkaitan dengan sifat alami perangkat lunak khususnya kompleksitas dan keabstrakan. Untuk memperoleh estimasi biaya, maka harus diketahui estimasi waktu yang diperlukan selama tahap pengembangan perangkat lunak. Estimasi biaya dan waktu yang diperoleh kemudian dialokasikan ke dalam fase-fase yang terjadi selama pengembangan, sehingga menghasilkan penjadwalan [3].

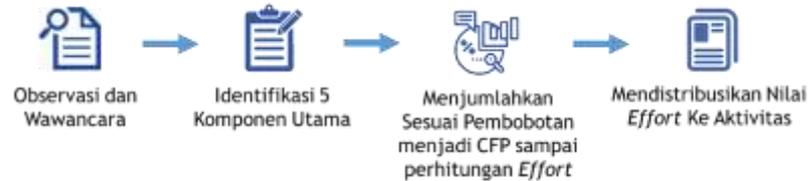
Terdapat beberapa metode dalam estimasi biaya perangkat lunak, di antaranya adalah: *Constructive Cost*, Model (COCOMO) II, *Expert Judgment*, *Function Point* (FP), *Analogi*, *Neural Network*, *Fuzzy*, dan lain-lain. Masing masing metode mempunyai kelebihan dan kekurangan untuk pengembangan perangkat lunak [7]. Dalam penelitian ini, kami mengajukan sebuah teknik paling umum dan familiar untuk diujicobakan pada data proyek yang telah berjalan di lapangan yaitu menggunakan metode *Function Point Analysis*. Metode estimasi biaya ini telah diteliti oleh Albrecht di perusahaan IBM sejak 1983 [1]. *Function Point* adalah pendekatan yang berorientasi fungsi mengukur fungsionalitas perangkat lunak untuk mengestimasi *size* perangkat lunak dan selanjutnya digunakan untuk estimasi biaya serta estimasi effort yang diperlukan untuk mengembangkan sistem [6]. Metode *Function Point* dapat digunakan untuk estimasi Effort dan estimasi biaya perangkat lunak berdasarkan jumlah dan kompleksitas dari fungsi yang ada. Metode *Function Point* (FP) dianggap lebih cepat karena tidak memerlukan hasil analisis sistem dalam bentuk skema pelaporan penggunaan naratif dan basis data konsep. Pada dasarnya metode *Function Point* (FP) merupakan indikator pengukuran yang tidak bisa ditinggalkan.

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 FAS (Financing Analysis System) yang terdapat pada PT. BPRS Lantabur Tebuireng. Secara garis besar terdapat metode yang perlu dilakukan untuk melakukan penelitian ini yang dijelaskan pada Gambar 1.



Gambar 1. Metode Penelitian

### 2.1 Pengumpulan Data

Dalam objek penelitian ini, penulis melakukan penelitian pada software FAS (Financing Analysis System) yang berada di PT. BPRS Lantabur Tebuireng. Metode penelitian yang digunakan penulis dalam mengumpulkan data ini adalah observasi dan wawancara.

### 2.2 Identifikasi 5 Komponen Utama

Terdapat 5 komponen dalam *Function Point* yang dapat digunakan secara efektif sebagai alat untuk memprediksi ukuran sebuah perangkat lunak/sistem yang akan dihasilkan. Komponen dan tipe fungsi pengguna berdasarkan pada karakteristik kompleksitas yang dimiliki meliputi sebagai berikut :

- External Input* (EI), adalah proses dasar yang memproses data dan informasi kontrol yang datang dari luar batasan aplikasi, baik dari user maupun sistem lainnya yang selanjutnya digunakan untuk mengupdate Internal Logical Files.
- External Output* (EO), adalah sebuah proses dasar dimana hasil data dilewatkan dari dalam ke keluar dari batasan aplikasi untuk menyediakan Informasi kepada user, baik dalam bentuk laporan, tampilan di layar, pesan error, dst.
- Internal Logical File* (ILF), adalah kelompok data atau kelompok informasi kontrol yang digunakan dalam aplikasi, merupakan data yang dikelompokkan secara logis, disimpan secara internal dan didapat dari input eksternal.
- External Interface File* (EIF), adalah kelompok data berelasi atau informasi kontrol yang dirujuk oleh aplikasi, tapi dipelihara oleh aplikasi lain, yang dikelompokkan secara logis namun berada diluar aplikasi yang menyediakan Informasi yang dibutuhkan aplikasi.
- External Inquiry* (EQ), fungsi utamanya adalah menyediakan informasi ke user melalui pengambilan/pemrosesan data atau informasi kontrol dari ILF/EIF, sebagai input online yang memicu respon dari software untuk menghasilkan output online.

Nilai kompleksitas akan berbeda dari satu perangkat lunak ke perangkat lunak lainnya. Nilai *Function Point* ini akan merupakan petunjuk terhadap besarnya ukuran perangkat lunak. Semakin besar nilai *Function Point* akan semakin besar pula ukuran perangkat lunaknya. Setiap komponen diklasifikasikan berdasarkan tingkat kompleksitas antara lain mudah, sedang, dan tinggi Nilai dari masing masing komponen kompleksitas dikalikan dengan bobot kompleksitas yang telah ditentukan pada Tabel 1 berikut ini :

Tabel 1. Bobot Kompleksitas

Komponen	Mudah	Sedang	Kompleks
<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

### 2.3 Menjumlahkan Sesuai Pembobotan menjadi CFP sampai perhitungan Effort

Langkah selanjutnya adalah pembobotan CFP dan kemudian dilanjutkan pada tahapan selanjutnya untuk sampai mencapai perhitungan *effort*. Adapun tahapan adalah sebagai berikut :

- Menghitung CFP (*Crude Function Points*)

Setelah mengidentifikasi tingkat kompleksitas selanjutnya adalah mengalikan dengan bobot kompleksitas dan menjumlahkannya menjadi CFP. Setelah mengidentifikasi tingkat kompleksitas maka selanjutnya adalah menghitung nilai CFP (*Crude Function Points*) dengan mengalikan masing-masing nilai dengan jumlah fitur yang ada di setiap kompleksitas. Kemudian hasil yang didapat kelima fungsi dijumlahkan untuk mendapatkan nilai CFP (*Crude Function Points*).

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

Tidak hanya memerlukan perhitungan jumlah kompleksitas dari beberapa fitur yang diberikan kepada pengguna saja, namun juga kepada operasional dari lingkungan sistem. Pada tahapan ini dilakukan perhitungan terhadap *Relative Complexity Adjustment Factor* (RCAF) pada sebuah proyek perangkat lunak. Terdapat 14 faktor yang dibuat mempengaruhi tingkat kesulitan yang berhubungan dengan implementasi sistem. Setiap faktor diberi nilai dari 0 sampai 5. Nilai 0 jika faktor tersebut tidak menimbulkan efek apapun dan 5 jika faktor tersebut sangat penting pada perangkat lunak yang diukur. Table penilaian RCAF dapat dilihat pada Tabel 2.

**Tabel 2.** 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>troughtput</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 *Function Point* (FP)

Dalam menghitung FP menggunakan rumus hasil dari penghitungan (Crud Function Point) CFP yang dikalikan dengan hasil perkalian nilai kontanta dan juga (Relative Complexity Adjustment Factor) RCAF menggunakan rumus sebagai berikut :

$$FP = CFP \times (0.65 + (0.01 \times RCAF)) \quad (1)$$

d) Menghitung Nilai *Effort*

Nilai *function point* telah diketahui langkah selanjutnya ialah melakukan perhitungan nilai *effort*, dengan rumus sebagai berikut :

$$Effort = FP \times 8.2 \quad (2)$$

e) Mengkalkulasi Nilai Kloc

Nilai FP dapat digunakan untuk mengukur Kloc (Kilo Lines of Code) perangkat lunak, *cost*, *effort*, *resource* yang dibutuhkan dan lama pengerjaan perangkat lunak. Untuk memperoleh estimasi ukuran perangkat lunak dalam satuan Kloc, nilai FP dikalikan dengan *Productivity Factor* berdasarkan bahasa pemrograman yang akan digunakan. *Productivity Factor* adalah jumlah kode logis per-*function point*. Untuk mengkalkulasi nilai Kloc maka dapat menggunakan rumus sebagai berikut :

$$Kloc = FP \times 56 \quad (3)$$

## 2.4 Mendistribusikan Nilai *Effort* Ke Aktivitas

Setelah melakukan perhitungan mulai dari menentukan CFP (*Crud Function Point*), RCAF (*Relative Complexity Adjustment Factor*), menghitung nilai FP, menghitung nilai *Effort*, dilanjutkan dengan konfersi FP menjadi biaya, maka langkah selanjutnya adalah dengan mendistribusikan *Effort* pada aktivitas masing-masing menggunakan *Payrate* yang sudah ditentukan. Adapun aktivitas yang digunakan pada penelitian ini berdasarkan pengembangan perangkat lunak yang dimulai dari *requirement*, *specifications*, *design*, *implementation*, *integration testing*, *acceptance & deployment*, *project management*, *configuration management*, *quality assurance*, *documentation*, *training & support*, *evaluation & testing*.

## 3. ANALISA DAN PEMBAHASAN

Pada bagian ini berisi analisa, hasil serta pembahasan dari metode penelitian. Berikut adalah hasil analisa dan pembahasan pada penelitian ini :

### 3.1 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan dengan 2 cara sebagai berikut :

a) Observasi

Dalam metode ini, penulis secara langsung melakukan pengamatan terhadap menu-menu dan fitur-fitur yang terdapat pada software FAS (Financing Analysis System) untuk selanjutnya dilakukan perhitungan serta pembobotan.

b) Wawancara

Disini penulis melakukan wawancara kepada pihak-pihak yang secara langsung menggunakan software FAS (Financing Analysis System). Pihak-pihak yang berkaitan langsung dengan software FAS (Financing Analysis System) antara lain adalah pihak-pihak yang terdapat dalam divisi AO (Accounting Office) seperti Pak Iwan, Pak Hadi serta Pak Musta'in. Penulis melakukan wawancara untuk menanyakan spesifikasi software yang dibutuhkan oleh perusahaan.

### 3.2 Identifikasi 5 Komponen Utama

Pada tahap ini penulis mencari jumlah dari komponen fungsional sistem, lalu diidentifikasi dan dilanjutkan dengan mengevaluasi kuantitasi bobot kerumitan dari tiap komponen tersebut. Berikut adalah bobot kompleksitas dari masing-masing komponen :

Tabel 3. Bobot Kompleksitas

Komponen	Level Kompleksitas			Total
	Mudah	Sedang	Kompleks	
<i>External Inputs</i> (EI)	156	1	0	-
<i>Internal Outputs</i> (EO)	42	0	0	-
<i>External Inquiries</i> (EQ)	1	0	0	-
<i>Internal Logical Files</i> (ILF)	0	0	0	-

<i>External Interface File (EIF)</i>	0	0	0	-
<b>Total</b>				<b>-</b>

### 3.3 Menjumlahkan Sesuai Pembobotan menjadi CFP sampai perhitungan Effort

a) Menghitung CFP (*Crude Function Points*)

Setelah mendefinisikan jumlah komponen pada perangkat lunak FAS kemudian pada tahap ini akan dikalikan dengan pembobotan kompleksitas. Pada Tabel 4 menunjukkan hasil perhitungan CFP adalah 634 pada perangkat lunak FAS yang dapat dilihat dari total penjumlahan masing-masing komponen.

**Tabel 4.** Hasil Perhitungan CFP (*Crude Function Points*)

Komponen	Level Kompleksitas			Total
	Mudah	Sedang	Kompleks	
External Inputs (EI)	468	4	0	<b>472</b>
Internal Outputs (EO)	168	0	0	<b>168</b>
External Inquiries (EQ)	3	0	0	<b>3</b>
Internal Logical Files (ILF)	0	0	0	<b>0</b>
External Interface File (EIF)	0	0	0	<b>0</b>
<b>Total</b>				<b>634</b>

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

Tabel 4 adalah hasil dari pembobotan masing-masing karakteristik. Hasil perhitungan dari RCAF (*Relative Complexity Adjustment Factor*) adalah 52.

**Tabel 5.** Hasil Pembobotan Karakteristik

No	Karakteristik	Bobot
1	Tingkat kompleksitas Komunikasi Data	4
2	Tingkat kompleksitas Pemrosesan Terdistribusi	4
3	Tingkat kompleksitas Performance	3
4	Tingkat kompleksitas Konfigurasi	3
5	Tingkat Frekuensi Penggunaan Software	5
6	Tingkat Frekuensi Input Data	5
7	Tingkat Kemudahan Penggunaan Bagi User	5
8	Tingkat Frekuensi Update Data	2
9	Tingkat Kompleksitas Prosesing Data	4
10	Tingkat Kemungkinan Penggunaan Kembali/Reusable Kode Program	3
11	Tingkat Kemudahan Dalam Instalasi	5
12	Tingkat Kemudahan operasional software (backup, recovery, dsb)	3
13	Tingkat Software dibuat untuk multi organisasi/perusahaan/client	3
14	Tingkat kompleksitas dalam mengikuti perubahan/fleksibel	3
<b>TOTAL</b>		<b>52</b>

c) Menghitung Nilai *Function Point (FP)*

Dalam menghitung FP menggunakan rumus hasil dari penghitungan (Crud Function Point) CFP yang dikalikan dengan hasil perkalian nilai kontanta dan juga (Relative Complexity Adjustment Factor) RCAF. Nilai FP untuk pengembangan perangkat lunak FAS (*Financing Analysis System*) adalah 752,31 sesuai dengan perhitungan sebagai berikut :

$$FP = CFP \times (0.65 + (0.01 \times RCAF)) \quad (1)$$

$$FP = 643 \times (0.65 + (0.01 \times 52))$$

$$FP = 752.31$$

d) Menghitung Nilai *Effort*

Untuk menghitung nilai *Effort* yaitu dengan mengalikan hasil penghitungan FP dan dikalikan dengan PF. Nilai *Effort* untuk pengembangan perangkat lunak FAS (*Financing Analysis System*) adalah 6.168,94 orang/jam sesuai dengan perhitungan sebagai berikut :

$$\begin{aligned} \text{Effort} &= \text{FP} \times 8.2 & (2) \\ \text{Effort} &= 752.31 \times 8.2 \\ \text{Effort} &= 6,168.94 \text{ (orang/jam)} \end{aligned}$$

e) Mengkalkulasi Nilai Kloc

Pada tahap ini mengubah nilai FP menjadi biaya yaitu menggunakan rumus hasil penghitungan FP dikalikan dengan nilai konstanta yaitu 56. Nilai Kloc untuk pengembangan perangkat lunak FAS (*Financing Analysis System*) adalah 7,768.32 Untuk mengkalkulasi nilai Kloc maka dapat menggunakan rumus sebagai berikut :

$$\begin{aligned} \text{Kloc} &= \text{FP} \times 56 & (3) \\ \text{Kloc} &= 752.31 \times 56 \\ \text{Kloc} &= 7,768.32 \end{aligned}$$

### 3.4 Mendistribusikan Nilai *Effort* Ke Aktivitas

Setelah melakukan perhitungan mulai dari menentukan *Crud Function Point* (CFP), (*Relative Complexity Adjustment Factor*) RCAF, menghitung nilai FP, menghitung nilai *Effort*, dilanjutkan dengan konversi FP menjadi biaya, maka langkah selanjutnya adalah dengan mendistribusikan effort pada aktivitas/jobdesk masing-masing menggunakan *Payrate* yang sudah ditentukan. Tabel 3 merupakan hasil dari distribusi effort kedalam aktifitas.

**Tabel 6.** Hasil Distribusi Effort Ke Aktivitas

No.	Activities	%	Effort	Payrate/hr
1	Requirement	1.60	117.69472	17,188
2	Specifications	7.50	551.694	17,188
3	Design	6.00	441.3552	10,313
4	Implementation	52.00	3825.0784	10,313
5	Integration testing	7.00	514.9144	10,313
6	Acceptance & deployment	5.50	404.5756	10,313
7	Project management	3.80	279.52496	34,375
8	Configuration management	4.30	316.30456	34,375
9	Quality assurance	0.90	66.20328	10,313
10	Documentation	8.40	617.89728	10,313
11	Training & support	1.00	73.5592	10,313
12	Evaluation & testingn	2.00	147.1184	10,313
<b>Total</b>		<b>100</b>	<b>7,355.92</b>	

## 4. IMPLEMENTASI

Pada tahap implementasi menjelaskan mengenai biaya keseluruhan yang disesuaikan dengan masing-masing pekerja dalam pengembangan perangkat lunak FAS dengan aktivitas yang dilakukannya. Berikut merupakan hasil dari perhitungan estimasi biaya perangkat lunak menggunakan metode *Function Point* pada pengembangan perangkat lunak FAS (*Financing Analysis System*) setelah dikelompokkan sesuai *jobdesk* masing-masing.

**Tabel 7.** Perhitungan Estimasi Biaya Perangkat Lunak

No	Job	Activities	Cost	%
1	Business Analyst	Reuirement	Rp 2,022,878	12%
		Specifications	Rp 9,482,241	
2	Desainer	Desainer	Rp 4,551,476	5%
		Integration testing	Rp 5,310,055	
3	Business Developer	Acceptance and Deployment	Rp 4,172,186	11%
		Training and Support	Rp 758,579	
4	Programer	Implementation	Rp 39,446,121	42%
5	Project Manager	Project managementt	Rp 9,608,671	22%
		Configuration managemen	Rp 10,872,969	
6	Quality Assurance	Quality Assurance	Rp 682,721	1%

7	Documentation	Documentation	Rp 6,372,066	7%
8	Evaluation And Tseting	Evaluation And Testing	Rp 1,517,159	2%
<b>Total Akhir</b>			<b>Rp 94,797,120</b>	<b>100%</b>

## 5. KESIMPULAN

Berdasarkan penelitian ini, metode *function point* dapat digunakan sebagai acuan perhitungan estimasi biaya pengembangan perangkat lunak dalam suatu perusahaan tentunya dalam bidang perbankan. Terdapat 5 komponen yang dapat digunakan secara efektif dalam perhitungan estimasi biaya yaitu *External Inputs* (EI), *Internal Outputs* (EO), *External Inquiries* (EQ), *Internal Logical Files* (ILF), dan *External Interface File* (EIF). Masing-masing komponen memiliki penilaian yang berbeda yang disesuaikan dengan perangkat lunak yang dibuat. Dari seluruh total komponen yang dinilai pada perangkat lunak FAS (*Financing Analysis System*) pada PT BPRS Lantabur Tebuireng menghasilkan CFP (*Crude Function Points*) sebesar 634 dan *Effort* adalah 6.168,94 orang/jam. Nilai *Effort* tersebut selanjutnya didistribusikan ke dalam aktivitas pengembangan perangkat lunak. Pada tahap akhir diperoleh hasil dalam pengembangan perangkat lunak yang sudah disesuaikan dengan kebutuhan perusahaan, sehingga hasil dari penelitian estimasi biaya untuk pengembangan *software* FAS (*Financing Analysis System*) pada PT BPRS Lantabur Tebuireng membutuhkan biaya sekitar Rp 94.797.120.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pegawai pada PT BPRS Lantabur Tebuireng yang mengizinkan dan membantu dalam menyelesaikan penelitian ini, serta Dosen Sistem Informasi Universitas Internasional Semen Indonesia (UISI) yang telah memberi dukungan moral terhadap penelitian ini.

## REFERENCES

- [1] A. J. Albrecht, "Measuring application development productivity," in *Joint SHARE/GUIDE/IBM Application Development Symposium*, 1979.
- [2] F. Fachrudin and Y. Pratama, "Eksperimen Seleksi Fitur Pada Parameter Proyek Untuk Software Effort Estimation dengan K-Nearest Neighbor," *J. Inform. J. Pengemb. IT*, 2017.
- [3] F. P. Laksamana, A. R. Perdanakusuma, and M. C. Saputra, "Evaluasi Biaya Pengembangan Sistem Informasi Pengelolaan Arsip Surat (SIPAS) Menggunakan Function Point dan Object Point (Studi Kasus : PT Sekawan Media Informatika)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 8, pp. 2692–2701, 2018.
- [4] J. Sevilla, L. I. Jiménez, and A. Plaza, "Sparse Unmixing-Based Content Retrieval of Hyperspectral Images on Graphics Processing Units," *IEEE Geosci. Remote Sens. Lett.*, 2015.
- [5] R. S. Dewi, T. W. Andari, A. P. Subriadi, and Sholih, "Function Points Method in Game Casual Context," in *Proceedings of 2018 International Conference on Electrical Engineering and Computer Science, ICECOS 2018*, 2019.
- [6] Suharjito, S. Nanda, and B. Soewito, "Modeling software effort estimation using hybrid PSO-ANFIS," in *Proceeding - 2016 International Seminar on Intelligent Technology and Its Application, ISITIA 2016: Recent Trends in Intelligent Computational Technologies for Sustainable Energy*, 2017.
- [7] T. N. Sharma *et al.*, "Estimasi Biaya Pembuatan Modul Enterprise Resource Planning (Erp) Untuk Unit Bisnis Pabrik Gula Di Pt. Perkebunan Xyz Dengan Metode Use Case Point," *J. Ilm. Teknol. Inf.*, 2012.