Jurnal Riset Komputer (JURIKOM), Vol. 6, No. 4, Agustus 2019
ISSN 2407-389X (Media Cetak)
Hal: 340-347

Eksperimen Estimasi Biaya Proyek Perangkat Lunak Menggunakan Metode *Function Point Analysis* (FPA)

Hendrawan[1], Pareza Alam Jusia[2], Errissya Rasywir[2], Yovi Pratama[2]

[1]Sistem Informasi, STIKOM Dinamika Bangsa, Jambi, Indonesia
[2]Teknik Informatika, STIKOM Dinamika Bangsa, Jambi, Indonesia
Email: [1]hendrawan.stikom@gmail.com, [2]parezaalam@gmail.com, [3]errissya.rasywir@gmail.com, [4]yovi.pratama@gmail.com

Abstrak

Dalam membangun perangkat lunak sebaiknya dimulai dengan *project planning*. Kegiatan ini diinisialisasi dengan aktivitas *estimation*. Ketika estimasi dilakukan, perlu dilakukan prediksi pada masa depan serta meng-*handle* ketidakpastian yang akan dilalui dalam menjalankan sebuah proyek perangkat lunak. Terdapat beberapa opsi dalam estimasi proyek. Namun, dalam penelitian ini, kami menggunakan sebuah teknik paling umum dan familiar untuk diujicobakan pada data proyek yang telah berjalan di lapangan. Dari hasil eksperimen yang dilakukan pada penelitian tersebut pada 20 (duapuluh) data proyek pembangunan perangkat lunak pada salah satu *sosftware house* di kota Jambi menggunakan metode *Function Point Analysis* (FPA), diperoleh hasil perhitungan nilai RMSE *(root mean squared error*) sebesar Rp 1,073,001,- dan nilai *error* sebesar 0.25. Artinya dapat ditarik kesimpulan bahwa perbedaan RMSE dari komparasi harga nyata yang ditransaksikan di lapangan dengan harga yang diestimasi menggunakan metode *Function Point Analysis* (FPA) adalah sebesar nilai yang disebutkan di atas, yang mana nilai tersebut termasuk nilai dengan gap yang cukup kecil. Serta, nilai error yang dihasilkan juga cukup kecil yakni sebesar 0.25.

Kata Kunci: Estimasi, *Software*, *Engineering*, Biaya, Proyek

Abstract

In building software, it should start with project planning. This activity is initialized with estimation activity. When estimation is done, it is necessary to make predictions in the future and handle the uncertainty that will be passed in running a software project. There are several options in project estimation. However, in this study, we used the most common and familiar technique to be tested on project data that had been running in the field. From the results of experiments conducted in the study on 20 (twenty) data software development projects in one of the software houses in Jambi city using the Function Point Analysis (FPA) method, the results of the calculation of the RMSE (root mean squared error) of Rp 1,073,001, - and an error value of 0.25. This means that it can be concluded that the difference in RMSE from the real price comparison transacted in the field with the price estimated using the Function Point Analysis (FPA) method is equal to the value mentioned above, which value includes a value with a fairly small gap. As well, the error value produced is also quite small at 0.25.

Keywords: Estimates, Software, Engineering, Costs, Projects

1. PENDAHULUAN

Dalam membangun perangkat lunak sebaiknya dimulai dengan *project planning* [1], [2]. Kegiatan ini diinisialisasi dengan aktivitas *estimation*. Ketika estimasi dilakukan, perlu dilakukan prediksi pada masa depan serta meng-*handle* ketidakpastian yang akan dilalui dalam menjalankan sebuah proyek perangkat lunak [3], [4]. Dalam estimasi juga perlu dilakukan perancangan kerangka kerja sehingga memudahakan estimasi terkait tenaga, sumber daya, biaya dan jadwal. Perencanaan dan estimasi harus melalui suatu proses penemuan informasi yang merujuk sebuah nilai yang dapat dipertanggungjawabkan [5]. Beberapa indikator yang perlu diestimasi pada sebuah proyek pembangunan perangkat lunak antara lain ruang lingkup (*scope*), sumber daya usaha atau manusia, data *hardware-software*, biaya operasional, total biaya dan *timeline*. Estimasi yang diperlukan dalam proyek *software* agar sumber daya, biaya, dan jadwal serta parameter terkait tersebut dapat digunakan untuk mengakses informasi historis yang baik dalam usaha dalam pengembangan *software* [6]. Kecerobohan dalam pengukuran kuantitatif dan kualitatif pada estimasi dalam *project planning* mampu menimbulkan ketidakpastian dalam estimasi. *Project complexity* sangat berpengaruh pada perencanaan. Kompleksitas merupakan pengukuran yang dipengaruhi oleh kebiasaan dengan pekerjaan sebelumnya secara relatif [7]. *Project size* juga merupakan faktor yang sangat mempengaruhi estimasi. *Software developer* juga harus melengkapi fungsi *input-ouput*, fitur, proses *inquiry* dan fungsi *interface*. Berbagai pendekatan dalam *software engineering* menggunakan pengembangan interaktif. Pendekatan ini dimungkinkan untuk melihat estimasi dan merevisinya sesuai kebutuhan kostumer[8].

Penentuan ruang lingkup perangkat lunak merupakan aktivitas pertama dalam perencanaan proyek perangkat lunak. Ruang lingkup perangkat lunak menggambarkan fungsi, proses, batasan, *interface* dan kehandalan. Ruang lingkup atau batasan ini mengidentifikasi dari batas yang ditempatkan pada perangkat lunak oleh perangkat keras eksternal, memori atau sistem informasi yang ada [9], [10]. Terdapat beberapa opsi dalam estimasi proyek secara manual antara lain: menunggu estimasi hingga proyek berakhir, berkaca pada proyek

sejenis, mendekomposisi komponen proyek menjadi leih sederhana serta memakai sebuah model empiris dalam estimasi. Dalam penelitian ini, kami mengajukan sebuah teknik paling umum dan familiar untuk diujicobakan pada data proyek yang telah berjalan di lapangan. Data yang digunakan mempunyai parameter yang cukup lengkap untuk diujikan dengan metode *function point analysis* (FPA) [1], [11]. Dengan eksperimen yang dilakukan, diketahui seberapa besar keakuratan estimasi dengan metode *function point analysis* (FPA) ini. Sebagaimana diketahui, metode *function point analysis* (FPA) selalu menjadi *underground* berbagai teknik estimasi proyek *software* pada umumnya. Walaupun banyak pendekatan lain yang dilakukan, seperti pendekatan *machine learning* atau berbasis heuristic. Namun, pada dasarnya metode *function point analysis* (FPA) merupakan indikator pengukuran yang tidak bisa ditinggalkan.

2. METODE PENELITIAN

2.1 Penelitian Awal Analisis Estimasi Biaya *Software* Dengan Metode FPA

Penelitian eksperimen estimasi biaya proyek perangkat lunak menggunakan metode *function point analysis* (FPA) diinisialisasi dengan kegiatan awal berupa penelitian awal yang meliputi sub kegiatan lainnya, antara lain: identifikasi masalah estimasi biaya proyek pembangunan suatu perangkat lunak, kemudian kegiatan studi literatur, dan berikutnya adalah kegiatan analisis solusi masalah estimasi biaya proyek perangkat lunak menggunakan metode *Function Point Analysis* (FPA). Kegiatan identifikasi masalah estimasi biaya proyek pembangunan suatu perangkat lunak dilakukan untuk menemukan masalah pada riset ini. Masalah yang diangkat dalam penelitian ini adalah menemukan bagaimana metode FPA dalam melakukan perkiraan biaya atau estimasi biaya pada data proyek pembangunan perangkat lunak yang kami gunakan dalam penelitian ini. Kegiatan studi literatur dilakukan untuk mempersiapkan kajian-kajian dan referensi terkait permasalahan yang didefenisikan. Adapun studi literatur yang kami lakukan antara lain adalah dengan melakukan kajian mengenaik estimasi biaya pada proyek perangkat lunak serta melakukan review pada penelitian terdahulu mengenai *function point analysis* (FPA). Selanjutnya, kegiatan analisis solusi masalah estimasi pada proyek pembangunan perangkat lunak. Kami mencoba melakukan percobaan untuk menkomparasi biaya *real* dilapangan dengan biaya yang diestimasi oleh metode *function point analysis* (FPA).

2.2 Pengumpulan Data Proyek *Software*

Data yang kami gunakan dalam penelitian ini adalah data 20 proyek perangkat lunak dari sebuah *Software House* di Kota Jambi. Data yang kami gunakan adalah besarnya biaya proyek, jumlah baris kode program, nilai performa kode program, jenis Bahasa pemrograman yang digunakan dan data kompleksitas proyek perangkat lunak.

2.3 Analisis Proyek *Software* Dengan Metode FPA

Pada tahapan ini kami melakukan penerapan metode *function point analysis* (FPA) pada 20 data proyek pembangunan perangkat lunak yang berisi data besarnya biaya proyek, jumlah baris kode program, nilai performa kode program, jenis Bahasa pemrograman yang digunakan dan data kompleksitas proyek perangkat lunak. Berikut ini adalah langkah untuk melakukan estimas biaya dengan menggunakan *function point analysis* (FPA):

a) Menghitung banyak RET/FTR dan DET
Nilai bobot dalam metode *function point analysis* (FPA) didapat dari menghitung banyak RET (*Record Element Type*) atau FTR (*File Type Reference* terdiri dari EIF & IEF) dan DET (ILF & EIF) yang kemudian masuk ke salah satu kelompok (Rendah, Rata-rata atau Tinggi). Kemudian setiap data proyek yang ujikan akan dinilai berdasarkan RET dan DET-nya masing-masing.

Tabel 1. Matrik Penilaian RET dan DET Proyek *Software*

ILF/EIF	**DET (*Data Element Type*)**		
RET	**1--19**	**20-50**	**51+**
1	Rendah	Rendah	Rata-rata
2--5	Rendah	Rata-rata	Tinggi
6+	Rata-rata	Tinggi	Tinggi

Selanjutnya, adalah memberikan nilai bobot dari kategori komponen pada proyek perangkat lunak. Terdapat 3 (tiga kategori penilaian pada setiap komponen perangkat lunak.Tiga kategori penilaian antara lain adalah kategori rendah, rata-rata dan tinggi. Penilaian ini, kami serahkan pada developer proyek untuk menilai komponen input, output, queries dan interface proyek *software*.

Tabel 2. Bobot Komponen Perangkat Lunak Pada Metode FPA

Komponen	Rendah	Rata-rata	Tinggi
Input	3	4	6
Output	4	5	7
Queries	3	4	6
File	7	10	15
Antarmuka	5	7	10

Setelah menghitung penilaian DET dan Bobot komponen pada 20 data proyek *software* dalam penelitian ini, dilakukan penentuan dari output perangkat lunak yang diukur dengan menggunakan tabel 3 berikut:

Tabel 3. Output Perbandingan DET dan Komponen

Rule	DET	Bobot Komponen	Output
1	Rendah	Rendah	Rendah
2	Rendah	Rata-rata	Rendah
3	Rendah	Tinggi	Rata-rata
4	Rata-rata	Rendah	Rendah
5	Rata-rata	Rata-rata	Rata-rata
6	Rata-rata	Tinggi	Tinggi
7	Tinggi	Rendah	Rata-rata
8	Tinggi	Rata-rata	Tinggi
9	Tinggi	Tinggi	Tinggi

b) Menghitung TUFP (*Total Unadjusted Function Point*)

Perhitungan TUFP menggunakan perkalian dari input pada pembobotan pada tabel 2 dikali dengan penilaian yang dikalikan oleh developer pada setiap komponen pada 20 data proyek *software* lalu dijumlahkan. Hasil penilaian TUFP setiap proyek dalam penelitian inni dapat dilihat pada tabel (bagian 3 mengenai analisis dan pembahasan).

c) Menghitung TDI (*Total Degree of Influence*)
Pada tahapan ini dilakukan perhitungan terhadap TDI (*Total Degree of Influence)* pada sebuah proyek perangkat lunak. Karena didalam riset ini digunakan 20 data proyek *software*, maka terdapat 20 kali penilaian TDI yang dilakukan. Hasil penilaian TDI pada 20 proyek yang digunakan dapat dilihat pada tabel 8. Penilaian tersebut dilakukan langsung oeh tim programmer yang membangun proyek *software* tersebut.

Tabel 4. Daftar Faktor Penilaian Terhadap *Software* dan Contoh Penilaian

ID	Faktor	Nilai VAF Sistem
C1	Data Communications	3
C2	Distributed Function	4
C3	Performance Objectives	3
C4	Heavily used configuration	4
C5	Transaction Rate	3
C6	On-line Data entry	3
C7	End user efficiency	4
C8	On-line Update	3
C9	Complex Processing	3
C10	Reusability	5
C11	installation ease	5
C12	operational ease	3
C13	Multiple Sites	3
C14	Facilitate change	4
	Total TDI	**50**

Penilaian yang dilakukan pada Tabel 4 untuk menghitung TDI digunakan skala 1 untuk menyatakan nilai sangat rendah, nilai 2 untuk nilai cukup rendah, nilai 3 untuk nilai sedang, nilai 4 untuk nilai baik dan nilai 5 untuk penilaian sangat baik.

d) Menghitung nilai TCA((*Technical Complexity Adjusment*)
Nilai TCA digunakan untuk mendapatkan nilai ketetapan kompleksitas teknis pada sebuah proyek *software*. Untuk mendapatkan nilai TCA digunakan rumus sebagai berikut:

TCA = 0.65 + 0.01 * TDI……………………………………(1)

e) Menghitung FP (*Function Point)/* TUFP(*Total Unadjusted Function Point)*
Menghitung TUFP(*Total Unadjusted Function Point)* dilakukan dengan menggunakan tabel 5 ini:

Tabel 5. Standar Penilaian Kompleksitas Proyek

Standar *Adjusted Project Complexity* (PCA)			TAFP
1	0.65	Simple systems	
2	1	Normal systems	?
3	1.35	Complex systems	

f) Menghitung Bobot Bobot Pemrograman
Pembobotan Bahasa pemrograman dilakukan untuk menghitung LOC (*Line Of Code)*. Tabel berikut berisi daftar pembobotan Bahasa pemrograman setiap jenis Bahasa pemrograman yang digunakan. Cara menghitung LOC (*Line Of Code)* menggunakan daftar bobot pada tabel 6 dapat dilihat pada langkah berikutnya.

Tabel 6. Daftar Pembobtan Bahasa Pemrograman

Konversi Bobot Bahasa	
C	130
php	53
COBOL	110
JAVA	55
C++	50
Turbo Pascal	50
Visual Basic	30
PowerBuilder	15
HTML	15
Packages (e.g., Access, Excel)	10--40

g) Menghitung LOC (*Line Of Code)*
Menghitung Lines of Codes

LOC = TUFP * LOC Bahasa……………………..…………(2)

h) Menghitung Effort dan Biaya
Pada langkah ini dilakukan perhitungan nilai *effort* lalu dikonervsikan menjadi nilai biaya estimasi pembangunan perangkat lunak.

$$Pm = A(ukuran)^{sf} \times (em_1) \times (em_2) \times \ldots \times (em_n) \quad (3)$$

Dimana nilai pm adalah biaya dalam 'person-month', kemudian nilai A adalah konstanta, nilai ukuran adalah jumlah FP (kdsi) dan sf adalah eksponen faktor skala.

2.4. Evaluasi Eksperimen

Metode evaluasi Eksperimen Estimasi Biaya *Software* menggunakan metode FPA yang digunakan dalam penelitian ini adalah nilai RMSE (*root mean squared error)* serta nilai *error*. Cara menghitung nilai RMSE dan nilai *error* adalah sebagai berikut:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(p_i - a_i)^2}{n}} \quad (4)$$

$$Error\ (\%) = \frac{\sum_{i=1}^{n} \frac{p_i - a_i}{a_i}}{n}\ x\ 100 \quad \text{.....} \quad (5)$$

Di mana nilai p_i adalah nilai estimasi biaya *software* output ke i , a_i adalah nilai asli pada lapangan dan nilai n=jumlah data yang digunakan.

3. ANALISIS DAN PEMBAHASAN

3.1 Analisis Hasil Perhitungan TUFP

Tabel 7 berikut ini adalah tabel hasil kegiatan menghitung penilaian DET dan Bobot komponen pada 20 data proyek *software* dalam penelitian ini, dilakukan penentuan dari output perangkat lunak yang diukur dengan menggunakan tabel 3.Setiap nilainya yang diinputkan dikalikan dengan bobot pada tabel 3. Hasil perkalian keseluruhan dijumlahkan menjadi niai TUFP.

Tabel 7. Hasil Perhitungan *TUFP*

NO	Id-Proyek	Tahap I TUFP															TUFP
		EI			EO			EQ			ILF			EIF			
		L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	
1	KSI-001	5			3	5		3			13			15			227
2	KSI-002	12				12			4			34			23		613
3	KSI-003	20			7		5	5			13	12		11			404
4	KSI-004	*4*				8			15			13			11		319
5	KSI-005	7				6			25						26		333
6	KSI-006		6	3	15	12		16				32			32		754
7	KSI-007	32			14	3			23				20				559
8	KSI-008	4				3	6	5		24		12			12		432
9	KSI-009					4			5		13			13			196
10	KSI-010	9				7					15			15			242
11	KSI-011		7	8	15				12			40			5		619
12	KSI-012				5	23		5	34		44	12			6		756
13	KSI-013					12	8		21	13		34	21		23		1094
14	KSI-014	5				6			14		12			2	4		223
15	KSI-015		17	3		16			32				24				654
16	KSI-016		4	3		4		12			39			6			393
17	KSI-017		3	3		12			21		33			7			440
18	KSI-018					13		15			30			11			375
19	KSI-019	7		3	8		9		25			27			14		602
20	KSI-020	4	5			12				13		32			13		581

Pada tabel 7 terdapat kolom EI, EO, EQ, ILF dan EIF. Kolom EI atau *External Input* digunakan untuk penilaian pada input sistem pada proyek perangkat lunak. Kolom EO atau *External Output* digunakan untuk penilaian pada output sistem pada proyek perangkat lunak.Kolom EQ atau *External Inquiry* digunakan untuk penilaian pada inquiry sistem pada proyek perangkat lunak. Kolom ILF atau *Internal logical File* digunakan untuk penilaian pada logika sistem pada proyek perangkat lunak. Kolom EIF atau *External Interface File* digunakan untuk penilaian pada *interface* sistem pada proyek perangkat lunak. Sedangkan nilai diwakilkan dengan kategori L, M dan H. Nilai L beratti low atau rendah, Nilai M berarti medum atau sedang. Nilai H berarti high atau tinggi. Selanjutnya adalah pehitungan nilai TDI pada 20 proyek *software* yang digunakan.

Tabel 8. Hasil Perhitungan TDI (*Total Degree of Influence)*

No	Id-Proyek	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	TDI
1	KSI-001	3	3	3	4	5	4	2	3	2	4	3	4	3	4	47
2	KSI-002	5	4	4	3	3	3	2	4	2	5	4	5	3	5	52
3	KSI-003	2	5	5	3	4	3	4	5	4	3	3	3	4	3	51
4	KSI-004	3	2	2	2	5	4	5	3	5	4	4	4	3	4	50
5	KSI-005	3	3	3	2	2	5	4	4	5	3	3	3	3	3	46

6	KSI-006	4	4	3	2	2	5	3	3	3	3	2	5	4	3	46
7	KSI-007	5	3	5	3	4	3	3	3	4	2	2	5	2	2	46
8	KSI-008	2	5	3	4	4	3	4	2	3	2	2	3	4	2	43
9	KSI-009	3	5	4	3	5	4	3	2	3	4	4	4	5	4	53
10	KSI-010	4	2	3	4	3	4	3	4	2	5	4	3	5	5	51
11	KSI-011	3	3	3	3	4	5	3	5	2	4	5	3	3	3	49
12	KSI-012	3	4	2	3	3	3	4	4	2	5	3	2	4	4	46
13	KSI-013	4	3	2	2	3	4	3	5	4	3	4	2	4	3	46
14	KSI-014	2	3	4	2	2	5	3	5	5	4	3	4	5	3	50
15	KSI-015	3	3	2	5	4	3	2	2	2	2	4	4	5	2	43
16	KSI-016	5	2	2	3	4	4	3	3	3	3	5	4	4	5	50
17	KSI-017	3	4	2	3	5	5	4	4	4	4	3	5	5	3	54
18	KSI-018	4	5	4	4	4	3	5	3	3	3	4	4	3	5	54
19	KSI-019	3	3	5	3	5	2	2	2	2	4	3	5	4	2	45
20	KSI-020	3	5	2	4	2	5	2	2	2	2	2	3	5	3	42

Tabel 8 di atas merupakan hasil dari perhitungan TDI (*Total Degree of Influence*). Kolom C1 adalah penilaian untuk *Data Communications*. Kolom C2 adalah penilaian untuk *Distributed Function*. Kolom C3 adalah penilaian untuk data *Performance Objectives*. Kolom C4 adalah penilaian untuk *data Heavily used configuration*. Kolom C5 adalah penilaian untuk data *Transaction Rate*. Kolom C6 adalah penilaian untuk data *On-line Data entry*. Kolom C7 adalah penilaian untuk data *End user efficiency*. Kolom C8 adalah penilaian untuk data *On-line Update*. Kolom C9 adalah penilaian untuk data *Complex Processing*. Kolom C10 adalah penilaian untuk data *Reusability*. Kolom C11 adalah penilaian untuk data *installation ease*. Kolom C12 adalah *operational ease*. Kolom C13 adalah penilaian untuk data *Multiple Sites*. Kolom C14 adalah penilaian untuk data *Facilitate change*.

Tabel 9. Analisis Hasil Penilaian PCA, TUFP LOC dan Productivity

No	Id-Proyek	PCA Range*	(PCA)	TAFP	FP LOC	SL OC	Productivity
1	KSI-001	0.65	1.12	254.24	15	3813.6	0.7
2	KSI-002	1	1.52	931.76	15	13976.4	1
3	KSI-003	1	1.51	610.04	15	9150.6	1
4	KSI-004	0.65	1.15	366.85	15	5502.75	0.7
5	KSI-005	0.65	1.11	369.63	15	5544.45	0.7
6	KSI-006	1.35	1.81	1364.74	15	20471.1	1.3
7	KSI-007	1	1.46	816.14	15	12242.1	1
8	KSI-008	1	1.43	617.76	15	9266.4	1
9	KSI-009	1	1.53	299.88	15	4498.2	1
10	KSI-010	1	1.51	365.42	15	5481.3	1
11	KSI-011	1.35	1.84	1138.96	15	17084.4	1.3
12	KSI-012	1.35	1.81	1368.36	15	20525.4	1.3
13	KSI-013	1.35	1.81	1980.14	15	29702.1	1.3
14	KSI-014	1	1.5	334.5	15	5017.5	1
15	KSI-015	1	1.43	935.22	15	14028.3	1
16	KSI-016	0.65	1.15	451.95	15	6779.25	0.7
17	KSI-017	1	1.54	677.6	15	10164	1
18	KSI-018	1	1.54	577.5	15	8662.5	1
19	KSI-019	1.35	1.8	1083.6	15	16254	1.3
20	KSI-020	1	1.42	825.02	15	12375.3	1

Tabel 9 adalah hasil penilaian dari beberapa parameter yang digunakan pada metode FPA untuk mengukur nilai-nilai seperti penilaian PCA (*Adjusted Project Complexity)*, TUFP (*Total Unadjusted Function Point*), LOC (*Line of code*) dan *Productivity*. Penilaian tersebut dilakukan pada 20 data proyek perangkat lunak. Hasil Penilaian ini dilakukan untuk menghitung nilai *effort* dan estimasi biaya pembangunan perangkat lunak.

4. IMPLEMENTASI

Dari hasil penerapan metode *Function Point Analysis (*FPA) yang dilakukan pada 20 data proyek perangkat lunak dari salah satu *software house* di kota Jambi. Diperoleh data yang disajikan pada tabel 10 berikut ini:

Tabel 10. Hasil Pengukuran Nilai RMSE dan Error

No	Id-Proyek	pi	ai	pi-ai	(pi-ai)2	$\frac{\sum_{i=1}^{n} \frac{p_i - a_i}{a_i}}{n}$
1	KSI-001	6,201,340	5,250,000	951,340	905,047,795,600	0.18
2	KSI-002	10,020,303	9,250,000	770,303	593,366,711,809	0.08
3	KSI-003	4,505,093	3,250,000	1,255,093	1,575,258,438,649	0.39
4	KSI-004	12,459,459	11,250,000	1,209,459	1,462,791,072,681	0.11
5	KSI-005	5,639,003	4,250,000	1,389,003	1,929,329,334,009	0.33
6	KSI-006	6,309,409	5,250,000	1,059,409	1,122,347,429,281	0.20
7	KSI-007	7,239,302	6,250,000	989,302	978,718,447,204	0.16
8	KSI-008	9,480,502	8,250,000	1,230,502	1,514,135,172,004	0.15
9	KSI-009	10,203,120	9,250,000	953,120	908,437,734,400	0.10
10	KSI-010	3,249,490	2,250,000	999,490	998,980,260,100	0.44
11	KSI-011	4,209,232	3,250,000	959,232	920,126,029,824	0.30
12	KSI-012	1,109,487	1,250,000	(140,513)	19,743,903,169	0.11
13	KSI-013	3,323,429	2,250,000	1,073,429	1,152,249,818,041	0.48
14	KSI-014	1,193,209	1,250,000	(56,791)	3,225,217,681	0.05
15	KSI-015	5,957,928	4,250,000	1,707,928	2,917,018,053,184	0.40
16	KSI-016	6,948,029	5,250,000	1,698,029	2,883,302,484,841	0.32
17	KSI-017	9,029,321	8,250,000	779,321	607,341,221,041	0.09
18	KSI-018	2,019,324	1,250,000	769,324	591,859,416,976	0.62
19	KSI-019	3,232,242	2,250,000	982,242	964,799,346,564	0.44
20	KSI-020	4,239,245	3,250,000	989,245	978,605,670,025	0.30
		RMSE			**1,073,002.41**	
			Error			**0.25**

Dari tabel 10 di atas terdapat beberapa kolom hasil perhitungan yang dilakukan dengan metode *Function Point Analysis* (FPA). Kolom Id-proyek adalah kolom yang berisi kode 20 (duapuluh) data daftar proyek pembangunan perangkat lunak yang digunakan pada penerapan metode *Function Point Analysis* (FPA) pada penelitian ini. Kemudian kolom p_i adalah kolom yang berisi biaya hasil estimasi yang dihitung menggunakan metode *Function Point Analysis* (FPA) dari 20 pdata proyek yang digunakan. Kolom a_i adalah kolom yang berisi biaya asli dari proyek pembangunan 20 data perangkat lunak yang diperoleh dari *sosftware house* yang merupakan vendor yang melakukan pembangunan perangkat lunak terrsebut. Selanjutnya, kolom pi-ai adalah kolom yang berisi selisih biaya real dari pembangunan perangkat lunak dan biaya estimasi hasil perhitungan dengan metode *Function Point Analysis* (FPA). Selanjutnya adalah kolom $(pi\text{-}ai)^2$ yang merupakan nilai selisih yang dikuadratkan, nilai kuadrat dari selisih ini digunakan untuk menghitung nilai RMSE *(root mean squared error*)dan nilai *error* dari komparasi biaya real dan biaya hasil estimasi.

5. KESIMPULAN

Dari hasil eksperimen yang dilakukan pada penelitian tersebut pada 20 (duapuluh) data proyek pembangunan perangkat lunak pada salah satu *sosftware house* di kota Jambi menggunakan metode *Function Point Analysis* (FPA), diperoleh hasil perhitungan nilai RMSE *(root mean squared error*) sebesar Rp 1,073,001,- dan nilai *error* sebesar 0.25. Artinya dapat ditarik kesimpulan bahwa perbedaan RMSE dari komparasi harga nyata yang ditransaksikan di lapangan dengan harga yang diestimasi menggunakan metode *Function Point Analysis* (FPA) adalah sebesar nilai yang disebutkan di atas, yang mana nilai tersebut termasuk nilai dengan gap yang cukup kecil. Serta, nilai error yang dihasilkan juga cukup kecil yakni sebesar 0.25. Namun, untuk catatan eksperimen ini hanya diujicobakan pada proyek dengan harga rendah yakni dibawah harga Rp 13.000.000,-. Belum diketahui hasilnya jika metode ini diujicobakan pada nilai proyek yang besar.

REFERENCES

[1] Y. Pratama and E. Rasywir, "Automatic Cost Estimation Analysis on Datawarehouse Project with Modified Analogy Based Method," in *Proceedings of 2018 International Conference on Electrical Engineering and Computer Science, ICECOS 2018*, 2019, pp. 171–176.

[2] 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.*, vol. 12, no. 12, pp. 2443–2447, 2015.

[3] S. Kumari and S. Pushkar, "Comparison and Analysis of Different Software Cost Estimation Methods," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 1, pp. 153–157, 2013.

[4] F. Fachruddin and Y. Pratama, "Eksperimen Seleksi Fitur Pada Parameter Proyek Untuk Software Effort Estimation dengan K-Nearest Neighbor," *J. Inform. J. Pengemb. IT*, vol. 2, no. 2, pp. 53–62, 2017.
[5] S. Kang, "Model-based Dynamic Cost Estimation and Tracking Method for Agile Software Development," in *9th IEEE/ACIS International Conference on Computer and Information Science Model-based*, 2010.
[6] K. Sangeetha and P. P. Dalal, "Analysis of Software Estimation Method : Function point and Use case point," *IJISET*, vol. 2, no. 11, pp. 880–884, 2015.
[7] R. Sarno and J. Sidabutar, "Comparison of Different Neural Network Architectures for Software Cost Estimation," in *International Conference on Computer, Control, Informatics and Its Applications Comparison*, 2015, pp. 68–73.
[8] S. Shekhar, "Review of Various Software Cost Estimation Techniques," vol. 141, no. 11, pp. 31–34, 2016.
[9] M. Azzeh, A. B. Nassif, and L. L. Minku, "An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation," *J. Syst. Softw.*, vol. 103, pp. 36–52, 2015.
[10] E. Rasywir, Y. Pratama, Hendrawan, and M. Istoningtyas, "Removal of Modulo as Hashing Modification Process in Essay Scoring System Using Rabin-Karp," *2018 Int. Conf. Electr. Eng. Comput. Sci.*, pp. 159–164, 2018.
[11] E. Kocaguneli, E. Kocaguneli, T. Menzies, A. Bener, and J. W. Keung, "Exploiting the Essential Assumptions of Analogy-based Effort Estimation," *IEEE Trans. Softw. Eng.*, vol. 38, no. 2, pp. 425–438, 2012.