

SISTEM INFORMASI MENGHITUNG BIAYA PEMBANGUNAN PERANGKAT LUNAK DENGAN MENGGUNAKAN METODE MATRIK FUNCTION POINT

Susi Widyastuti, Wahyu Ariandi

Program Komputerisasi Akuntansi, Program Teknik Informatika, STIKOM Poltek Cirebon, Indonesia

Email: miss_siwy@yahoo.com, wahyuariandi@mail.ugm.ac.id

Abstrak

Semakin berkembangnya kebutuhan perangkat lunak di instansi pemerintahan sebagai sistem informasi untuk mewujudkan transparansi, serta membantu mempercepat kegiatan melayani kebutuhan masyarakat, maka semakin luas kebutuhan perangkat lunak, sehingga mengakibatkan melakukan kerjasama dengan pengembang/konsultan (pihak ke tiga). Pengadaan perangkat lunak membutuhkan biaya yang tidak sedikit, namun kesulitan untuk membuktikan biaya pengadaan perangkat lunak, sesuai dengan standar harga yang telah disepakati. Selain itu, dari pihak pengembang juga mengalami kesulitan menentukan perkiraan biaya perangkat lunak, memperkirakan target waktu penyelesaian pembangunan/pembuatan perangkat lunak, dan menghitung gaji pegawai/programer berdasarkan kontribusi yang telah dilakukan dalam satu proyek. Untuk itu, maka diperlukan penelitian yang dapat mengukur kelayakan harga dari perangkat lunak. Salah satu pemecahan masalah adalah dengan menerapkan Metode *Matrik Function Point*, yaitu dengan menghitung parameter pengukuran perangkat lunak, menghitung *factor kompleksitas*, menghitung titik fungsi (*Function Point*), kebutuhan *programer* dalam proyek, serta kemampuan produktifitas programer.

Kata Kunci: pengukuran perangkat lunak; matrik function point; perhitungan biaya pembangunan perangkat lunak

Abstract:

The growing need for software in government agencies as an information system to realize transparency, and help accelerate activities to serve the needs of the community, the wider the need for software, resulting in cooperation with developers / consultants (third parties). Procurement of software requires a lot of costs, but it is difficult to prove the cost of procuring software, in accordance with agreed price standards. In addition, developers also have difficulty determining the estimated cost of software, estimating the target time of completion of software development / creation, and calculating the salaries of employees / programmers based on contributions that have been made in one project. For that, it requires research that can measure the feasibility of the price of the software. One of the problem solving is to apply the Function Point Matrix Method, which is to calculate software measurement parameters, calculate complexity factors,

calculate function points, programmer needs in projects, and programmer productivity capabilities.

Keywords: *software measurements; matrix function point; calculation of software development costs*

Received: 2021-09-20; Accepted: 2021-10-05; Published: 2021-10-20

Pendahuluan

Seiring perkembangan zaman, kemajuan teknologi komputer dan telekomunikasi telah menjadi kebutuhan yang sangat membantu dalam menyelesaikan banyak pekerjaan dengan cepat, tepat dan akurat (Widyastuti, Ariandi, & Sulistiono, 2019). Penggunaan teknologi komputer yang semakin meluas dan berkembang, maka berbagai macam aplikasi tumbuh dalam jumlah, ukuran, dan kegunaannya. Pengadaan untuk mengembangkan perangkat lunak juga tumbuh dengan baik. Sebagai hasil dari pertumbuhan pengembangan perangkat lunak terjadi konsekuensi yaitu kesalahan dalam estimasi biaya perangkat lunak menjadi banyak juga. Di instansi pemerintahan, ketika kerjasama dengan pengembang dalam pengadaan perangkat lunak, seringkali mengalami kesulitan membuktikan kelayakan harga perangkat lunak yang dipesan, antara biaya yang dikeluarkan dengan harga perangkat lunak yang diadakan.

Dalam pengembang/konsultan, seringkali mengalami kesulitan ketika dihadapkan pada suatu keadaan bahwa mereka harus menentukan harga jual dari sebuah perangkat lunak yang telah dibuatnya dan rincian biaya pembangunan perangkat lunak. Kesulitan tersebut dapat diatasi dengan menggunakan suatu teknik untuk memperkirakan ukuran suatu perangkat lunak, yang dapat dijadikan patokan untuk menentukan harga dasar dari perangkat lunak tersebut. Hasil pengukuran tersebut berupa satuan sesuai dengan metoda estimasi biaya perancangan perangkat lunak yang digunakan. Misalnya dengan metoda *Function point* yang mempunyai satuan untuk ukuran yaitu *Function Point* atau untuk ukuran yaitu *size/fp*.

Metoda *Function Point* memiliki keunggulan yaitu kemampuannya untuk menyediakan perkiraan volume proyek dalam bentuk sumber daya pengembangan yang dibutuhkan, perkiraan ini memberikan dasar penting untuk perkiraan sumber daya yang dibutuhkan perusahaan perangkat lunak dalam mempersiapkan proposal tender dan project plan atau pengembang/konsultan dapat memperkirakan harga jual perangkat lunak yang dibuatnya (Permatasari, 2016). Penggunaan metoda *Function Point* juga dapat mencegah atau setidaknya mengurangi secara substansial risiko dari kesalahan manajerial karena *underestimate* pada perencanaan biaya. Dengan adanya masalah tersebut, perusahaan atau orang yang bergerak dibidang pengembang/konsultan membutuhkan sebuah sistem perhitungan biaya pembuatan perangkat lunak untuk memudahkan dalam membuat rencana anggaran biaya dan juga menetapkan harga dari suatu perangkat lunak itu. Maka penulis mengusulkan sebuah penelitian dengan judul “Sistem Informasi Menghitung Biaya Pembangunan Perangkat Lunak dengan menggunakan Metode Matrik *Function Point*”.

State Of The Art bidang penelitian: penelitian ini memiliki aspek analisis utama yaitu menghitung rincian biaya pembangunan perangkat lunak dengan menggunakan *matrik function point*. Penelitian yang telah dilakukan oleh peneliti terdahulu adalah tentang estimasi biaya perangkat lunak menggunakan metode *function point* dengan studi kasus di CV. Aptikma Indonesia, yaitu meneliti tentang Analisis perbandingan antara metode *Function Point* dan *Guesstimate* pada perangkat lunak A dan B sehingga dapat menjadi bahan pertimbangan dalam menyelesaikan masalah. Pembagian lingkup kerja perangkat lunak A dan perangkat lunak B pada CV Aptikma Indonesia menggunakan pendekatan *Work Breakdown Structure* serta penjadwalan menggunakan *Gantt Chart* dan hasil estimasi biaya menggunakan metode *Function Point*. Perhitungan matematika pada perangkat lunak A menggunakan metode *Function Point* menghasilkan estimasi *effort* sejumlah 56 orang selama 11,5 bulan dengan total biaya sebesar Rp.610.000.000,00. Sedangkan pada perangkat lunak B menghasilkan estimasi *effort* sejumlah 9 orang selama 6 bulan dengan total biaya sebesar Rp 59.350.000,00.

Penelitian lainnya tentang estimasi ukuran perangkat lunak, adalah proses pengembangan untuk menentukan arah dan tujuan serta menjaga proses pengembangan tetap dalam kendali. Untuk mendapatkan hasil estimasi yang akurat, diperlukan metode yang memiliki akurasi tinggi. *Function Point Analysis* (FPA) merupakan metode estimasi ukuran perangkat lunak yang memiliki akurasi cukup baik (Saptono & Hutama, 2015). FPA memperkirakan ukuran perangkat lunak dalam *Function Point* (FP) yang dapat digunakan untuk menghitung *Line of Code* (LOC) (Parlika, Wijaya, Khariono, & Fernanda, 2020).

Dalam penelitian ini diusulkan untuk dibangun sebuah aplikasi yang dapat digunakan untuk memperkirakan ukuran *software* dengan metode FPA yang dapat digunakan oleh pengembang perangkat lunak. Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah *Rapid Application Development* (RAD). Dibutuhkan empat iterasi untuk membangun aplikasi dalam penelitian ini (Saputra & Siahaan, 2014).

Dilakukan pengujian dengan metode *Black Box* menggunakan 13 data aplikasi pada iterasi keempat, didapatkan hasil nilai rata-rata error relatif hasil perhitungan yang dilakukan oleh aplikasi adalah sebesar 3,52% (Hamzah, Saptono, & Anggrainingsih, 2017). Biaya perangkat lunak diperkirakan melalui upaya dan jumlah komponen yang berfungsi yang diukur masing-masing dalam satuan orang bulan (p-m) dan titik fungsi (FP). Dalam makalah ini kami telah mempertimbangkan biaya perangkat lunak berdasarkan FP karena FP tidak bergantung pada teknologi. Pada awalnya *function point analysis* (FPA) dirancang tanpa mengacu pada landasan teoritis yang didasarkan pada pengukuran yang dilakukan oleh tim ahli. Data titik fungsi dijelaskan untuk lebih dari seratus proyek pengembangan perangkat lunak dalam literatur. Dibahas pula keterbatasan model yang dihasilkan dalam mengestimasi upaya pembangunan. Makalah ini mencoba untuk mempelajari dan mengukur biaya perangkat lunak dalam kasus beberapa proyek atau kumpulan perangkat lunak. Dalam kasus proyek atau perangkat lunak tunggal, kami mencoba mempelajari dan mengukur jumlah titik fungsi (FPC)

untuk berbagai komponen perangkat lunak atau jenis fungsi (FT). Prinsip *Entropi Maksimum* (MEP) adalah teknik yang sangat populer untuk memperkirakan informasi atau *entropi maksimum* yang tunduk pada batasan yang diberikan. Makalah ini menyajikan aplikasi Prinsip *Entropi Maksimum* (MEP) untuk mendistribusikan Jumlah Titik Fungsi yang Tidak Disesuaikan (UFPC) dengan tunduk pada biaya perangkat lunak tertentu. Setelah itu, aplikasi ini diaplikasikan pada perangkat lunak untuk mengalokasikan biaya perangkat lunak individu Ketika total biaya untuk perangkat lunak diberikan. Dalam penelitian ini kami juga telah menganalisis proporsi *Unadjusted Function Point Counts* (UFPCs), *Number of FPs* (# FPs), dan bobot berbagai komponen fungsional atau FT untuk biaya perangkat lunak tertentu (Patel, 2013). Dalam pengembangan perangkat lunak sangat diperlukan kesesuaian dan ketepatan dalam menentukan ukuran atau nilai perangkat lunak agar sesuai dengan operasi yang akan dilakukan. Berbagai macam metode perhitungan telah banyak digunakan untuk mengestimasi ukuran *software*, salah satunya dengan menggunakan (FPA). *Software* penghitungan volume berdasarkan skala kompleksitas. Karena titik ukurnya sangat subyektif, maka untuk menjaga konsistensi dan *validitas* hasil, metode ini harus dijalankan oleh seorang profesional yang berpengalaman.

Metode inilah yang kemudian penulis terapkan untuk mengukur kompleksitas sistem informasi akademik STIKOM Dinamika Bangsa Jambi dengan menggunakan pendekatan pemodelan terstruktur. Pengukuran yang dilakukan dalam penelitian ini terdiri dari gambaran sistem informasi yang dibangun ke dalam struktur. Yang kemudian dianalisis dengan menghitung model *Crude Function Points* (CRP), kompleksitas relative *Adjustment Factor* (RCAF), dan kemudian menghitung fungsi titik. Dari hasil perhitungan menggunakan FPA untuk pengukuran kualitas *software* sistem akademik STIKOM Dinamika Bangsa Jambi diperoleh nilai FP 166,32 termasuk baik. Nilai *function point* yang dihasilkan akan digunakan pengembang dalam menentukan harga dan biaya sistem perangkat lunak yang akan dibangun atau dikembangkan (Rohayani, Gaol, Soewito, & Warnars, 2017).

Berdasarkan beberapa penelitian terdahulu maka dalam penelitian ini, adalah penelitian yang dapat mengukur kelayakan harga dari perangkat lunak, dengan adanya rincian biaya pembangunan perangkat lunak, sehingga dapat dibuktikan kecocokan harga perangkat lunaknya. Pemecahan masalah adalah dengan menerapkan Metode *Matrik Function Point*, yaitu dengan dilakukan analisa pengukuran perangkat lunak dengan menghitung parameter pengukuran perangkat lunak, menghitung faktor kompleksitas, menghitung titik fungsi (*Function Point*) (Khairani, 2015). Kemudian dengan melakukan analisa data perusahaan, seperti kebutuhan programmer dalam proyek, kemampuan *produktifitas programmer* dan penghasilan *programmer*.

Metode Penelitian

Sesuai Gambar 2, Tahapan-tahapan Metode *Metrik Function Point* (FP) pada penelntian ini mengacu pada metode perhitungan FP dalam bukunya *Software-Engineering 7th ED* by Roger S. Pressman (Pressman, 2012) yaitu sebagai berikut :

A. Menghitung Pengukuran Perangkat Lunak

1. Menghitung metrik *function point Crude Funtion Point*, (CFP)

Rumus: $\sum(\text{input, output, penyelidikan, file, interface}) * \text{Faktor Bobor}$

Parameter pengukuran	jumlah;	Faktor Pembobotan			=	[]
		sederhana	rata-rata	kompleks		
jumlah input pemakai	[] x	3	4	6	=	[]
jumlah output pemakai	[] x	4	5	7	=	[]
jumlah penyelidikan pemakai	[] x	3	4	6	=	[]
jumlah file	[] x	7	10	15	=	[]
jumlah interface internal	[] x	6	7	10	=	[]
total	—————→				=	[]

Gambar 1
Perhitungan Metrik Function Point (Crude Funtion Point, CFP)

Domain informasi pada gambar 1, yang biasa digunakan ada 5 karakteristik, yaitu:

- 1) Jumlah input pemakai: setiap input pemakai yang memberikan data yang berorientasi pada aplikasi yang jelas pada perangkat lunak (harus dibedakan dari penelitian yang dihitung secara terpisah).
- 2) Jumlah output pemakai: setiap output pemakai yang memberikan informasi yang berorientasi pada aplikasi kepada pemakai. Pada konteks ini output mengacu pada laporan, layar, tampilan kesalahan, dsb. Jenis data individual pada laporan tidak dihitung terpisah
- 3) Jumlah penyelidikan pemakai: input online yang mengakibatkan munculnya beberapa respon perangkat lunak yang cepat dalam bentuk output online.
- 4) Jumlah file: setiap master logika (pengelompokan data logis yang menjadi suatu bagian dari sebuah database yang besar atau sebuah file terpisah).
- 5) Jumlah *interface* eksternal: semua *interface* yang dapat dibaca oleh mesin yang digunakan untuk memindahkan informasi ke sistem yang lain

2. Menghitung Harga Penyesuaian Kompleksitas (*Function ke ...i, Fi*)

Harga Penyesuaian Kompleksitas berdasarkan respon (Quisioner) pada tabel Harga Penyesuaian Kompleksitas (*Function ke ...i, Fi*), yaitu menghitung bobot kompleksitas berdasarkan 14 (empat belas) karakteristik atau faktor dan memiliki penilaian kompleksitas sebagai berikut:

Penilaian Komplexitas memilik skala 0 s/d 5:

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

$$\text{Rumus: } Fi = \sum_{i=1}^{14} \text{Karakteristik}$$

Tabel 1
Harga Penyesuaian Kompleksitas (Function ke-i ,Fi)

No	Respon dari pertanyaan	Tidak	Insidental	Moderat	Rata-rata	Signifikan	Esensial
		0	1	2	3	4	5
1.	Tingkat kompleksitas komunikasi data						
2.	Tingkat kompleksitas proses distribusi						
3.	Tingkat kompleksitas kinerja						
4.	Tingkat kompleksitas frekuensi penggunaan software						
5.	Tingkat kompleksitas input data online						
6.	Tingkat kompleksitas kemudahan penggunaan online bagi user						
7.	Tingkat kompleksitas frekuensi update data online						
8.	Tingkat kompleksitas prosesing data (in-out, file, inquiry)						
9.	Tingkat kompleksitas pemrosesan internal						
10	Tingkat kompleksitas kemungkinan penggunaan kembali kode program						
11	Tingkat kompleksitas konversi & instalasi						
12	Tingkat software dibuat untuk multi organisasi/perusahaan/client						
13	Tingkat kompleksitas desain mengikuti perubahan/fleksibel						
SUB TOTAL							
TOTAL (Fi)							

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

$$\text{RCAF} = 0,65 + (0.01 \times Fi)$$

4. Menghitung titik-titik fungsi (*Function Point*, FP)

$$\text{FP} = \text{CFP} \times \text{RCAF}$$

Keterangan:

FP = *Function Point*

CFP = *Crude Function Point*

RCAF = *Relative Complexity Adjustment Factor*

B. Menghitung Harga Per FP

$$\text{Harga per FP} = \frac{\text{Gaji}}{\text{Produktifitas}}$$

Keterangan:

Harga per FP = Harga per *Function Point*

Gaji = Gaji Produktifitas Programmer per FP (*Function Point*)

Produktifitas = Produktifitas yang dihasilkan Programmer per FP (*Function Point*) per bulan

C. Menghitung Biaya Pembangunan Perangkat Lunak

Biaya Pembangunan Perangkat Lunak = FP * Harga per FP

D. Menghitung waktu yang dibutuhkan untuk menyelesaikan pembangunan perangkat lunak (W)

$$W = \frac{FP / \sum \text{Programmer}}{\text{Produktifitas}}$$

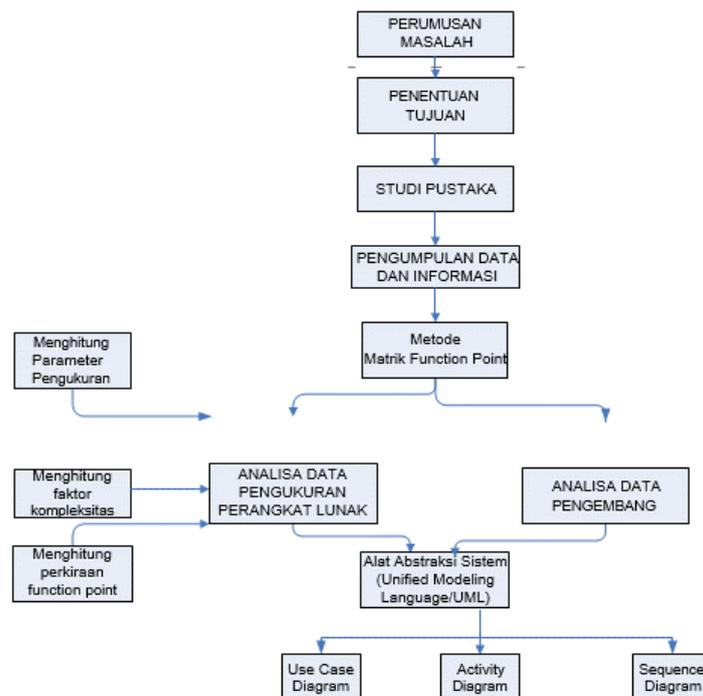
Keterangan:

W = waktu yang dibutuhkan untuk menyelesaikan pembangunan perangkat lunak

FP = titik-titik fungsi (FP)

Programmer = Jumlah Programmer yang terlibat dalam pembangunan perangkat lunak

Produktifitas = Produktifitas yang dihasilkan Programmer per FP (*Function Point*) per bulan



Gambar 2
Tahapan Matrik Function Point

Tahapan matrik *function point* seperti pada gambar 2, dimulai dari analisa pengukuran perangkat lunak dengan menghitung parameter pengukuran perangkat lunak, menghitung faktor kompleksitas, menghitung titik fungsi (*Function Point*). Kemudian dengan melakukan analisa data perusahaan, seperti kebutuhan programmer dalam proyek, kemampuan produktifitas programmer dan penghasilan programmer. Selanjutnya, untuk desain fungsionalitas yang diharapkan dari sebuah system tentang “apa” yang diperbuat system dan menggambarkan interaksi aktor dengan system digambarkan dalam bentuk usecase. Langkah berikutnya memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses metrik *function point*, yang digambarkan dengan menggunakan *Activity Diagrams*. Setelah semua kebutuhan input didapatkan maka langkah selanjutnya yaitu dilakukan perancangan secara *squance diagram*.

Hasil dan Pembahasan

A. Menghitung Pengukuran Perangkat Lunak

1. Menghitung metrik *function point* (*Crude Funtion Point*, CFP)

Crude Function Points (CFP) adalah untuk menghitung bobot nilai dari komponen- komponen *Function Point* yang dikaitkan dengan bobot *software* yang akan dibuat seperti pada table 2 Perhitungan CFP adalah Komponen-komponen *Function Point* terdiri dari 5 karakteristik (Sandhea, Maulidiyah, & Sari, 2020).

Tabel 2
Perhitungan CFP

Tipe komponen	Level Komplexitas									Total	
	Sederhana			Rata-rata			Kompleks				
	Jml	Bobot	Point	Jml	Bobot	Point	Jml	Bobot	Point		
Jumlah input pemakai	100	3	300	4			6			300	
Jumlah output pemakai		4		40	5	200	7			200	
Jumlah penyelidikan pemakai		3		22	4	88	6			88	
Jumlah file		7			10		9	15	135	135	
Tipe Interface External		6			7		20	10	200	200	
										Total CFP	923

2. Menghitung Harga Penyesuaian Kompleksitas (*Function ke-i* , *Fi*)

Tabel 3 Harga Penyesuaian Kompleksitas (*Function ke-i* , *Fi*) digunakan untuk menghitung bobot kompleksitas dari *software* berdasarkan 14 karakteristik. Penilaian Kompleksitas memiliki skala 0 s/d 5.

Tabel 3
Harga Penyesuaian Kompleksitas (Function ke-i, Fi)

No	Respon dari pertanyaan	Tidak	Insident	Moder	Rata-	Signifika	Esensia	
		pengar uh	al	at	rata	n	l	
		0	1	2	3	4	5	
1	Tingkat kompleksitas backup & recovery				3			
2	Tingkat kompleksitas komunikasi data				3			
3	Tingkat kompleksitas proses distribusi						5	
4	Tingkat kompleksitas kinerja			2				
5	Tingkat kompleksitas frekuensi penggunaan <i>software</i>						5	
6	Tingkat kompleksitas input data <i>online</i>				3			
7	Tingkat kompleksitas kemudahan penggunaan <i>online</i> bagi user						5	
8	Tingkat kompleksitas frekuensi update data <i>online</i>		1					
9	Tingkat kompleksitas prosesing data (<i>in-out, file, inquiry</i>)				3			
10	Tingkat kompleksitas pemrosesan <i>internal</i>				3			
11	Tingkat kompleksitas kemungkinan penggunaan kembali kode program				3			
12	Tingkat kompleksitas konversi & instalasi			2				
13	Tingkat software dibuat untuk multi organisasi/perusahaan/client				3			
14	Tingkat kompleksitas desain mengikuti perubahan/fleksibel					4		
SUB TOTAL		0	1	4	21	4	15	
TOTAL (Fi)		45						

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

Diketahui pada tabel $Fi = 45$, maka:

$$RCAF = 0,65 + (0,01 \times Fi) \quad RCAF = 0,65 + (0,01 \times 45)$$

$$RCAF = 0,65 + (0,45)$$

$$RCAF = 1,1$$

4. Menghitung titik-titik fungsi (*Function Point*, FP)

Diketahui Tabel 1 Perhitungan CFP, CFP = 923 dan berdasarkan tahap ke-4, *Relative Complexity Adjustment Factor* (RCAF) = 1,1 maka:

$$\begin{aligned} \text{FP} &= \text{CFP} \times \text{RCAF} \\ \text{FP} &= 923 \times 1,1 \\ \text{FP} &= 1.015,3 \end{aligned}$$

Total FP untuk proyek pembangunan perangkat lunak adalah 1.015,3 FP

B. Menghitung Harga Per FP

Jika di perusahaan pengembang/konsultan dalam pengerjaan pembangunan perangkat lunak melibatkan 10(sepuluh) programer, dan produktifitas programer 20 FP per orang perbulan dengan gaji programmer Rp. 200.000,- per orang per bulan. Maka:

$$\text{Harga per FP} = \frac{\text{Gaji}}{\text{Produktifitas}}$$

$$\text{Harga per FP} = \frac{\text{Rp.200.000,-}}{20} = \text{Rp.10.000,-}$$

C. Menghitung Biaya Pembangunan Perangkat Lunak Diketahui total FP yang dibutuhkan untuk proyek pembangunan perangkat lunak, sesuai perhitungan titik-titik fungsi (*Function Point*, FP) pada tahapan no. 4 (empat) metode metrik *function point* adalah 1015,3 FP, maka Biaya Pembangunan Perangkat Lunak = FP * Harga per FP
Biaya Pembangunan Perangkat Lunak = 1.015,3 * Rp. 10.000,-
Biaya Pembangunan Perangkat Lunak = Rp. 101.530.000,-

D. Menghitung waktu yang dibutuhkan untuk menyelesaikan pembangunan perangkat lunak (W).

Diketahui total FP yang dibutuhkan untuk proyek pembangunan perangkat lunak, sesuai perhitungan titik-titik fungsi (*Function Point*, FP) pada tahapan no. 4 (empat) metode metrik *function point* adalah 1015,3 FP, sedangkan perusahaan pengembang/konsultan dalam pengerjaan pembangunan perangkat lunak melibatkan 10(sepuluh) programer dan produktifitas programer 20 FP per orang perbulan, maka:

$$W = \frac{\text{FP} / \sum \text{Programer}}{\text{Produktifitas}}$$

$$W = \frac{1.015,3 / 10}{20} = \frac{101,53}{20} = 5,0765 \text{ bulan}$$

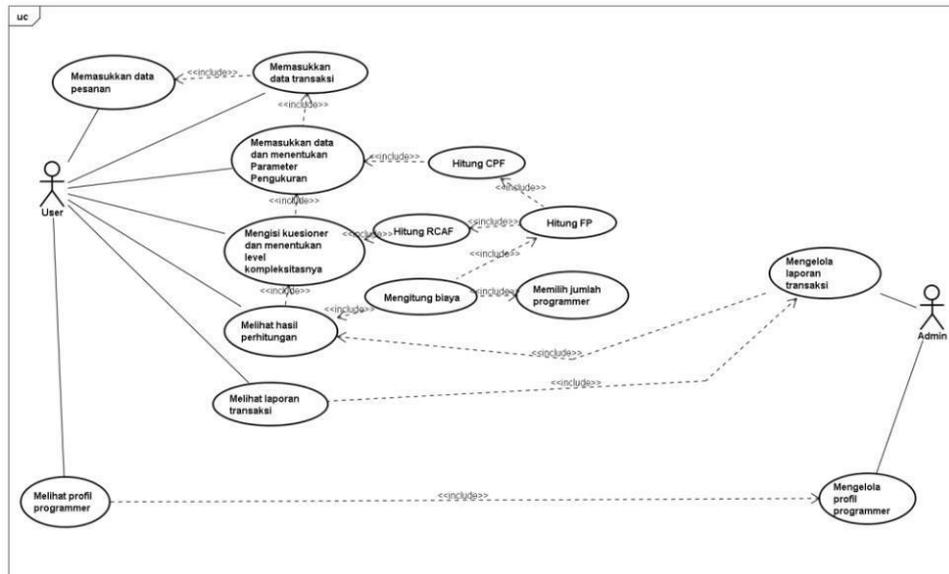
Jadi waktu yang dibutuhkan untuk menyelesaikan pembangunan perangkat lunak adalah sekitar 5 – 6 bulan.

1) **Desain Perancangan**

Desain perancangan Sistem Informasi Menghitung Biaya Pembangunan Perangkat Lunak dengan menggunakan Metode *Metrik Function Point* adalah sebagai berikut (Widyaningtyas, Arwan, & Rusdianto, 2016):

2) Use Case Diagrams

Diagram ini digunakan untuk menggambarkan fungsionalitas dari sebuah sistem dan menggambarkan interaksi aktor dengan sistem tersebut dapat dilihat pada gambar 3 *Use case diagrams* sistem informasi biaya perhitungan.



Gambar 3

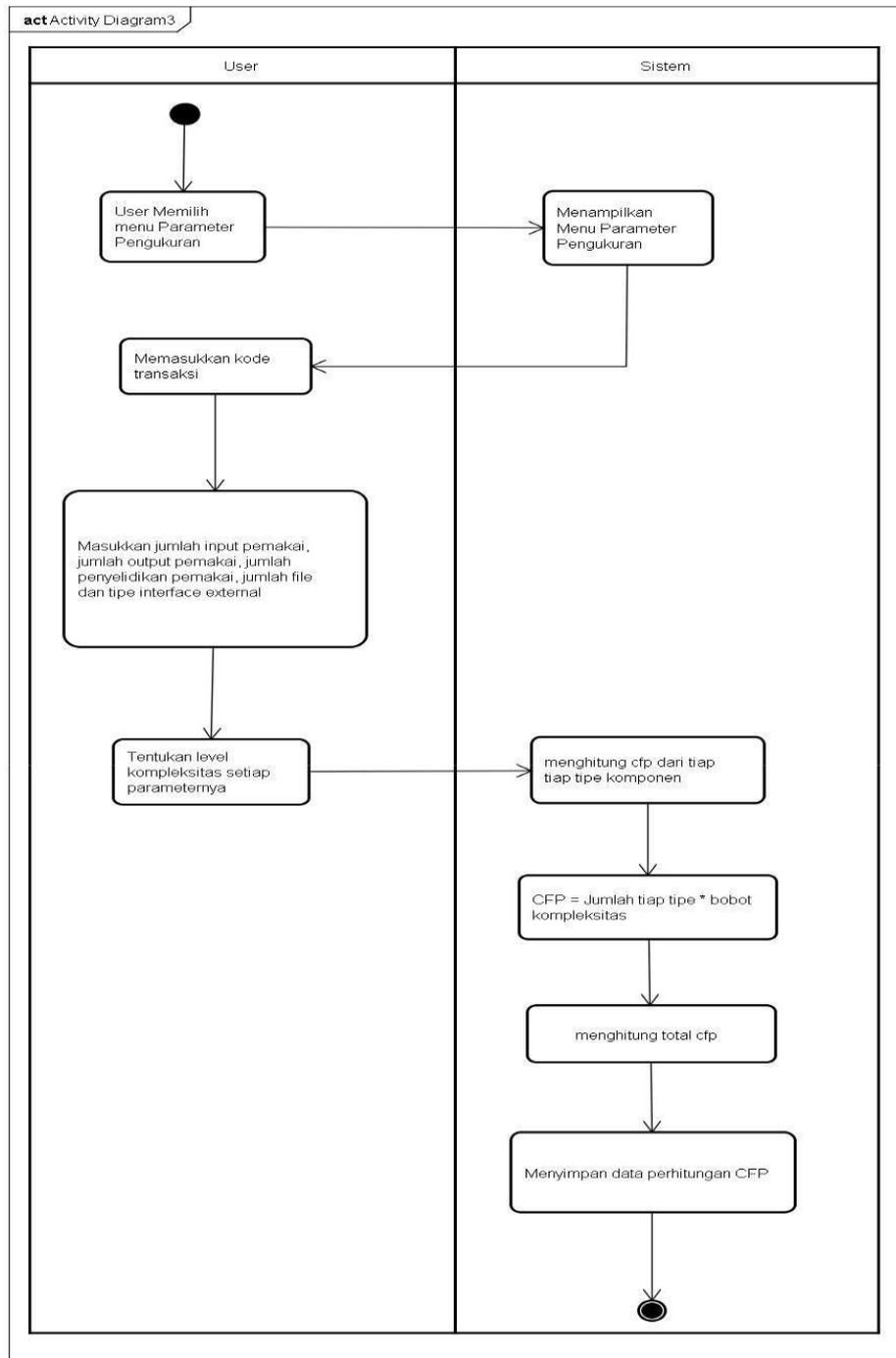
***Use Case Diagrams* Sistem Informasi Perkiraan Biaya Pembuatan Perangkat Lunak**

3) Activity Diagrams

Diagram ini menjabarkan proses dari masing-masing menu yang ada pada sistem baru.

a. *Activity Diagrams* Menu Parameter Pengukuran

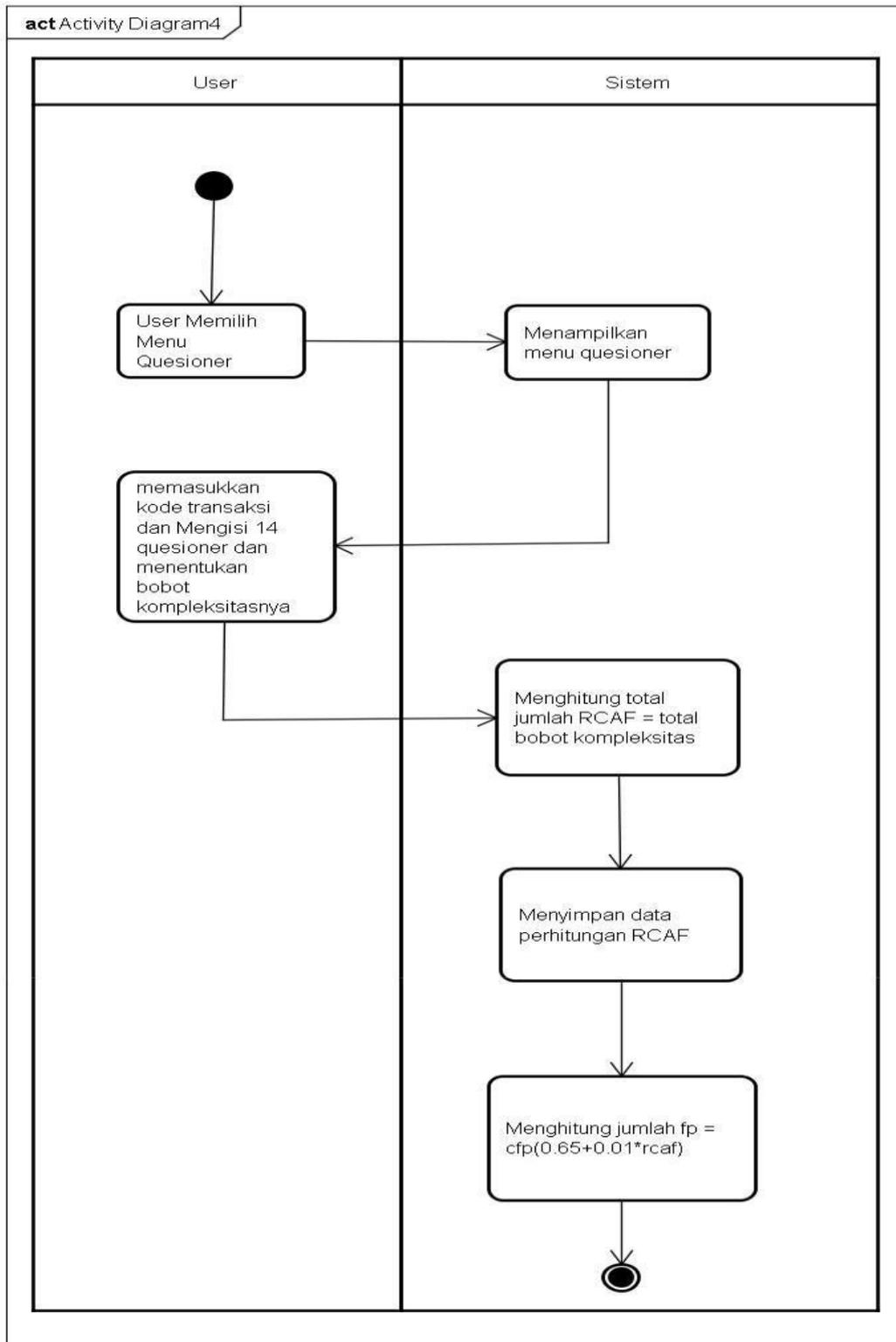
Berikut ini adalah Gambar 4 *Activity Diagrams* Menu parameter Pengukuran, pada menu ini terdapat proses perhitungan CFP.



Gambar 4
Activity Diagrams Menu Parameter pengukuran

b. Activity Diagrams Menu Quesioner

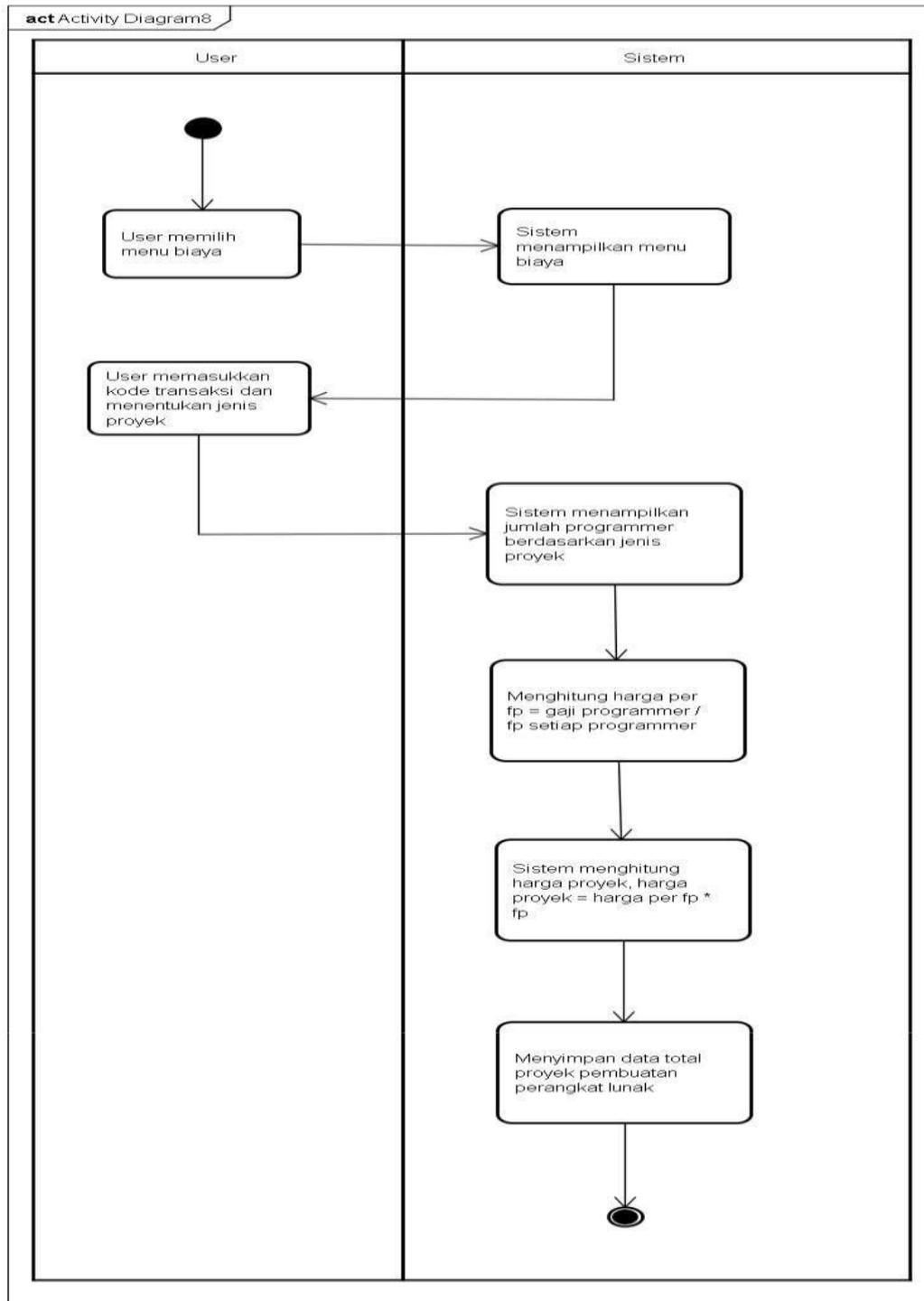
Berikut ini adalah Gambar 5 *Activity Diagrams Menu Quesioner*, pada menu ini terdapat proses perhitungan RCAF dan FP.



Gambar 5
Activity Diagrams Menu Kuesioner

c. *Activity Diagrams* Menu Perhitungan Biaya

Berikut ini adalah Gambar 6 *Activity Diagrams* Menu Perhitungan Biaya

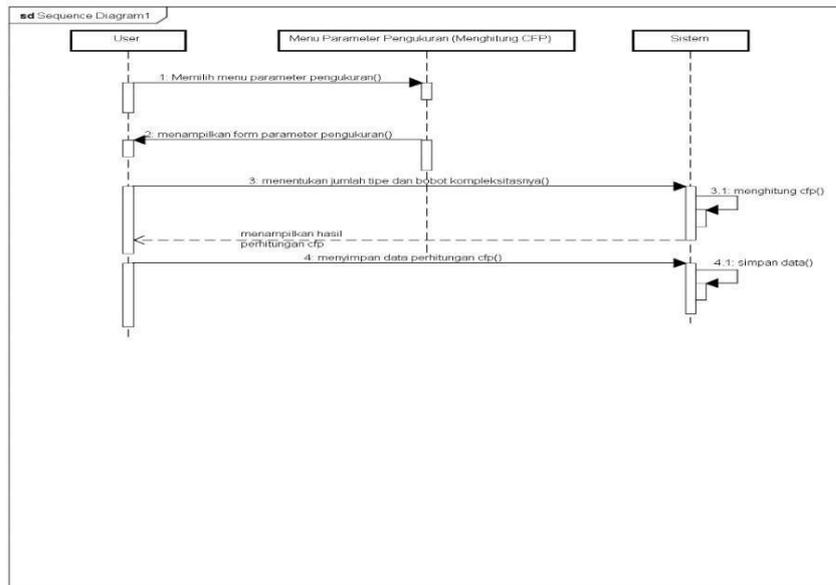


Gambar 6
Activity Diagrams Menu Perhitungan Biaya

d. Sequence Diagrams

1) *Sequence Diagrams* Menu Parameter Pengukuran

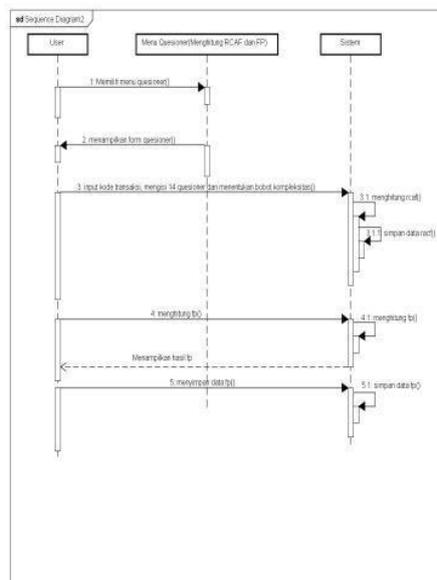
Berikut ini adalah Gambar 7 *Sequence Diagrams* Menu Parameter Pengukuran:



Gambar 7
Sequence Diagram Parameter Pengukuran

e. Sequence Diagrams Menu Quesioner

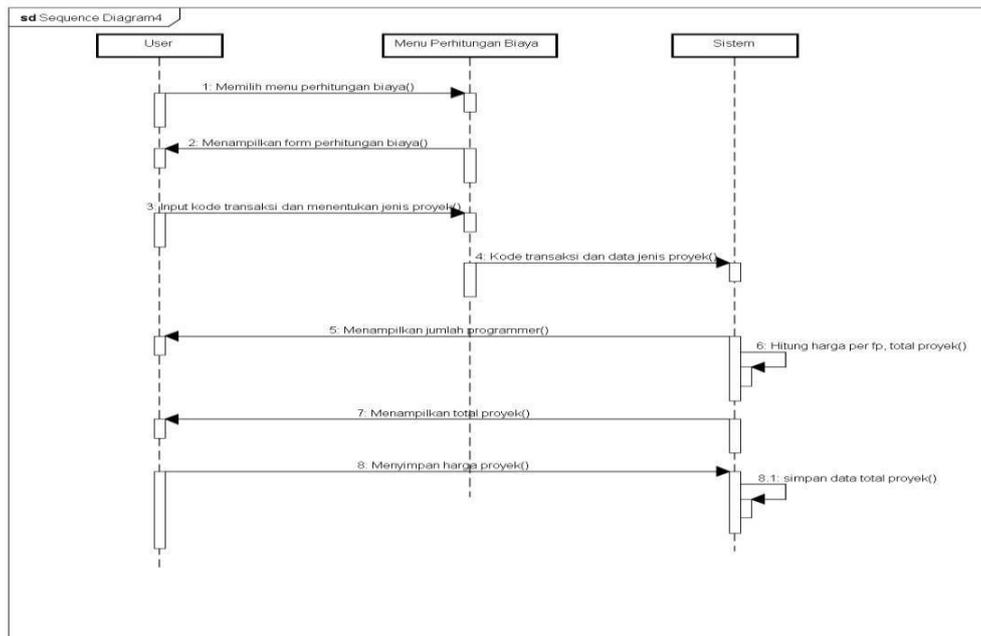
Berikut ini adalah Gambar 8 *Sequence Diagrams* Menu Quesioner:



Gambar 8
Sequence Diagram Quesioner

f. Sequence Diagrams Menu Perhitungan Biaya

Berikut ini adalah Gambar 9 *Sequence Diagrams* Menu Perhitungan Biaya:



Gambar 9
Sequence Diagram Perhitungan biaya

Kesimpulan

Berdasarkan hasil penelitian dapat disimpulkan bahwa dengan adanya penelitian yang dapat mengukur kelayakan harga dari perangkat lunak, dengan pemecahan masalah menerapkan Metode *Matrik Function Point*, maka: 1). Memberikan kemudahan bagi instansi pemerintahan untuk membuktikan kelayakan harga perangkat lunak yang diadakan. 2). Memberikan kemudahan pada perusahaan (pengembang/konsultan) dalam menyusun rencana anggaran biaya untuk membangun perangkat lunak dan membantu dalam menentukan harga jual perangkat lunak yang dibuatnya. 3). Memberikan perkiraan waktu yang dibutuhkan untuk menyelesaikan pembangunan perangkat lunak berdasarkan jumlah programmer yang dilibatkan.

BIBLIOGRAFI

- Hamzah, Hamzah, Saptono, Ristu, & Anggrainingsih, Rini. (2017). Development of Software Size Estimation Application using Function Point Analysis (FPA) Approach with Rapid Application Development (RAD). *ITSmart: Jurnal Teknologi Dan Informasi*, 5(2), 96–103. [Google Scholar](#)
- Khairani, Dewi. (2015). Studi kasus pengukuran sistem informasi menggunakan function point (FP). *Jurnal Teknik Informatika*, 8(2). [Google Scholar](#)
- Parlika, Rizky, Wijaya, Devan Cakra Mudra, Khariono, Heri, & Fernanda, Rifky Akhmad. (2020). Studi literatur perbandingan antara metode LOC, COCOMO, FPA dalam ranah software metric. *Jurnal Pendidikan Informatika Dan Sains*, 9(1), 66–74. [Google Scholar](#)
- Patel, S. (2013). Function point distribution using maximum entropy principle. *IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, Shimla, 20. [Google Scholar](#)
- Permatasari, Rochma. (2016). *Analisis Perbandingan Estimasi Effort Proyek Perangkat Lunak Menggunakan Metode Function Point Dengan Masukan Dokumen Use Case, Data Flow Diagram, Dan Natural Language*. Institut Teknologi Sepuluh Nopember. [Google Scholar](#)
- Pressman, Roger S. (2012). Software-Engineering 7th ED by Roger S. Pressman. In *Software Engineering A Practitioner's Approach*. [Google Scholar](#)
- Rohayani, Hetty, Gaol, Ford Lumban, Soewito, Benfano, & Warnars, Harco Leslie Hendric Spits. (2017). Estimated measurement quality software on structural model academic system with function point analysis. *2017 International Conference on Applied Computer and Communication Technologies (ComCom)*, 1–5. IEEE. [Google Scholar](#)
- Sandhea, Annisa Azzar, Maulidiyah, Sri, & Sari, Renny Dewi. (2020). Estimasi Biaya Perangkat Lunak Pada Aplikasi SIBIMA Universitas XYZ dengan Menggunakan Metode Function Point. *JURIKOM (Jurnal Riset Komputer)*, 7(1), 41–49. [Google Scholar](#)
- Saptono, Ristu, & Hutama, Galih Dian. (2015). Peningkatan Akurasi Estimasi Ukuran Perangkat Lunak dengan Menerapkan Logika Samar Metode Mamdani. *Scientific Journal of Informatics*, 2(1), 41–51. [Google Scholar](#)
- Saputra, Pramana Yoga, & Siahaan, Daniel Oranova. (2014). Analisis dan Desain Sistem Informasi Akademik Politeknik Negeri Malang Menggunakan Metode Rapid Application Development (RAD). *Prosiding Seminar Nasional Manajemen Teknologi XX Program Studi MMT-ITS, Surabaya 1 Februari 2014*. [Google Scholar](#)

Widyaningtyas, Yuni, Arwan, Achmad, & Rusdianto, Denny S. (2016). Perhitungan Ukuran Kompleksitas Fungsional Perangkat Lunak Dengan Metrik Function Point. *Teknologi: Jurnal Ilmiah Sistem Informasi*, 6(1), 38–48. [Google Scholar](#)

Widyastuti, Susi, Ariandi, Wahyu, & Sulistiono, Vergamana. (2019). Implementasi Kriptografi AES Dalam Pengamanan Data Seleksi Peserta JAMKESMAS. *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*. <https://doi.org/10.46772/intech.v1i02.66> [Google Scholar](#)

Copyright holder:

Susi Widyastuti, Wahyu Ariandi (2021)

First publication right:

Syntax Literate: Jurnal Ilmiah Indonesia

This article is licensed under:

