JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 4, Nomor 2, April 2020, Page 445-452
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI 10.30865/mib.v4i2.2083

Pengukuran Perangkat Lunak Untuk *Effort Estimation* dengan Teknik Pembelajaran Mesin

Maria Rosario Borroek[1], Errissya Rasywir[2], Yovi Pratama[2]

[1] Sistem Informasi, Program Studi Imu Komputer, Universitas Dinamika Bangsa, Jambi, Indonesia
[2] Teknik Informatika , Program Studi Imu Komputer, Universitas Dinamika Bangsa, Jambi, Indonesia
Email: [1] diamar_ros@yahoo.com, [2]errissya.rasywir@gmail.com, [3]yovi.pratama@gmail.com

Abstrak−Software effort estimation adalah mengestimasi sejumlah sumber daya yang diperlukan dalam mengembangkan perangkat lunak. Untuk itu software effort estimation merupakan hal yang penting sehingga perlu melihat pengaruh pengukuran perangkat lunak terhadap software effort estimation yang dilakukan dengan teknik pembelajaran mesin.Berdasarkan hal tersebut peneliti mencoba membangun sebuah sistem yang mampumelakukan pengukuran perangkat lunak.Pada penelitianinidilakukan eksperimen terhadap teknik pengukuran perangkat lunak (FPA, FPA dengan Sugeno fuzzy dan FPA dengan mamdani fuzzy). Tiga Jenis teknik tersebut dikomparasi dengan tiga data proyek untuk selanjutnya dilakukan software effort estimation.Untuk evaluasi, penelitianini melakukan evaluasi menggunakan penilaian Developer sebagai Sistem Analis Proyek.Hasil penelitian bahwa nilai LOC dan effort pada sebuah sistem yang sama dapat berbeda jika dihitung dengan penggunaan metode FPA, FPA Mamdany fuzzy dan FPA Sugeno Fuzzy. Nilai LOC dan Effort tertinggi dihasilkan oleh FPA Mamdany Fuzzy pada Proyek DUMAS POLDA SUMSEL. Sedangkan nilai effort terendah dan LOC terendah dihasilkan oleh FPA Sugeno Fuzzy. Hal ini dapat ditelusuri dari mekanisme perhitungan yang dilakukan oleh FPA Sugeno Fuzzy dimana metode ini sama sekali tidak menghitung nilai input , output , file, query dan interface. Perhitungan FPA Sugeno fuzzy dilakukan dengan cara kasar hanya menilai dari kesulitan pembuatan sistem. Untuk menaikkan harga suatu proyek agar dihargai lebih tinggi Metode FPA Mamdani Fuzzy lebih direkomendasikan.

Kata Kunci: FPA, Mamdani, Sugeno, Fuzzy, Effort

Abstract−Software effort estimation is to estimate the amount of resources needed in developing the software. For that software effort estimation is important so need to see the effect of software measurement to software effort estimation which is done by machine learning technique. Based on this the researcher tries to build a system capable of measuring software. In this study experiments on software measurement techniques (FPA, FPA with Sugeno fuzzy and FPA with mamdani fuzzy). The three types of techniques are compared with the three project data for further software effort estimation. For evaluation, this study evaluates using the assessment of the Developeras Analyst of the Project. The results of the study that the LOC and effort values on a similar system can be different if calculated by the use of FPA, Fam Mamdany fuzzy and FPA Sugeno Fuzzy. The highest LOC and Effort values are generated by FPA Mamdany Fuzzy on Project DUMAS POLDA SUMSEL. While the lowest effort value and lowest LOC produced by FPA Sugeno Fuzzy. This can be traced from the calculation mechanisms performed by FPA Sugeno Fuzzy where this method does not count the input, output, file, query and interface values at all. The calculation of FPA Sugeno fuzzy is done by roughly judging only from the difficulty of making the system. To raise the price of a project in order to be rewarded higher FAT methods Mamdani Fuzzy is recommended.

Keywords: FPA, Mamdani, Sugeno, Fuzzy, Effort.

1. PENDAHULUAN

Software effort estimation atau estimasi usaha perangkat lunak adalah bagian penting dari sebuah manajemen proyek.Estimasi yang akurat membantu kita menyelesaikan proyek dalam waktu dan anggaran yang telah ditentukan.Terdapat berbagai teknik, model estimasi dan *tools* untuk estimasi perangkat lunak [1], [2].

Berbagai penelitian terdahulu telah melakukan *software effort estimation* dengan berbagai metode, baik metode non pembelajaran mesin maupun pembelajaran mesin. Beberapa metode estimasi usaha perangkat lunak dengan metode non pembelajaran mesinantara lain SLIM, COCOMO, *use case point*, dan metode analogi *sherpped*. Penelitian *software effort estimation* yang dikembangkan dengan pembelajaran mesin antara lain penggunaan jaringan syaraf tiruan [3], *supportvector machine*[4], [5], *naïve bayes*; *k-nearest neighbor* [2], *fuzzy;* serta *liner regression* [6]–[8]. Kemudian penelitian eksperimen Seleksi Fitur Pada Parameter Proyek Untuk *Software Effort Estimation* dengan *K-Nearest Neighbors* berhasil menurunkan nilai error estimasi [1], [3], [9]. Selain itu penulis juga pernah melakukan perhitungan besaran fungsional dengan menggunakan COSMIC dataset yang dikemas dengan *Case Based*[10], [11] .Hasil penelitian tersebut menemukan bahwa*software effort estimation* yang dilakukan dengan teknik pembelajaran mesin menghasilkan nilai estimasi yang lebih baik dengan nilai *error* yang lebih kecil dibanding dengan teknik non pembelajaran mesin.

Software effort estimation sangat tergantung pada variabel perhitungan dari ukuran perangkat lunak[12].Semakin akurat mengestimasi pengukuran perangkat lunak maka semakin akurat dalam *software effort estimation* [3]. Dalam melakukan pengukuran perangkat lunak terdapat berbagai metode yang dapat digunakanmisalnya *line of code* (LOC),*function point analysis* metoda FPAdan *COSMIC*, FPA dengan *Mamdani Fuzzy* [4].

Tantangan yang dihadapi oleh pengembang adalah bagaimana melakukan pengukuran perangkat lunaksecara akurat.Pengukuran perangkat lunak dapat menentukan biaya perangkat lunak yang terlampir pada tahap awal proyek pengembangan *software*.Namun, pengukuran perangkat lunaksulit dilakukan karena nilainya

Submitted: **13/03/2020**; Accepted: **02/04/2020**; Published: **25/04/2020**

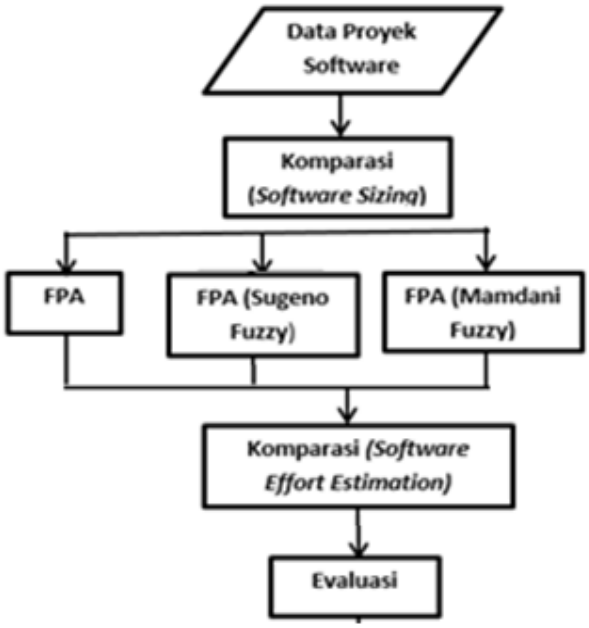
JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 4, Nomor 2, April 2020, Page 445-452
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI 10.30865/mib.v4i2.2083

tidak menentu dan tidak akurat. Hal tersebut karena hanya sedikit detail yang diketahui tentang proyek di awal pengembangan *software*. Terdapat beberapa kelemahan dari metode pengukuran perangkat lunak formal antara lain nilai pengukuran yang bergantung pada desain atau impementasi sistem, proses pengukuran *software* yang belum terdapatnya standar pengukurnya. Berdasarkan hal tersebut maka penelitian ini dilakukan untuk menemukan metode pengukuranterbaik dalam melakukan *software effort estimation* dengan teknik pembelajaran mesin.

2. METODE PENELITIAN

Berikut merupakan tahapan penulis dalam melakukan penelitian, yaitu:

1. Penelitian Awal
   Pada tahapan ini dikumpulkan bahan penelitian dari berbagai sumber pustaka, seperti buku, jurnal (baik cetak maupun online), prosiding, majalah, artikel dan sumber lain yang relevan. Masalah yang dibahas adalah bagaimana pengaruh metode pengukuran perangkat lunak terhadap *software effort estimation* yang dilakukan dengan teknik pembelajaran mesin, metode pengukuran perangkat lunakapa yang paling akurat dalam mengukur *software* serta bagaimana melakukan pengukuran perangkat lunak secara otomatis. Teori yang diambil tentang *software estimation effort*, pengukuran perangkat lunak, *FPA*, *FPA dengan Sugeno fuzzy* dan*FPA dengan mamdani fuzzy*, *k-nearest neighbor dan SVM* yang akan diujikan guna melakukan eksperimen yang dibahas dalam penelitian ini.
2. Pengumpulan Data
   Data diambil dari beberapa proyek perangkat lunak. Proyek tesebut antara lain: Sistem Pengaduan Masyarakat POLDA (DUMAS Sumsel), SIMPEG BKD OKI Sumsel dan RSFD *Activity System* (RAS) BTN Pusat Jakarta. Proyek tersebut merupakan proyek yang sudah final dan berupa *source code* dengan Bahasa pemrograman PHP yang dibangun dengan *framework* codeigniter.
3. Melakukan *Software Sizing* untuk *Software Effort Estimation*
   Kegiatan ini berupa pengukuran perangkat lunakyang mana hasil pengukuran tersebut digunakan dalam melakukan *software effort estimation*. Kemudian hasil *effort estimation*akan dievaluasi. Berikut gambar 2 adalah gambaran secara umum mengenai skema sistem penelitian yang dilakukan:



Gambar 1. Gambar Alur Riset

4. Melakukan Perbandingan Metoda
   Kegiatan ini melakukan eksperimen dengan cara melakukan komparasi terhadap beberapa metode pengukuran perangkat lunak(*FPA*, *FPA dengan Sugeno fuzzy* dan *FPA dengan mamdani fuzzy*). Kemudian hasil pengukuran dengan 3 metode tersebut dilakukan *software effort estimation.* Alur Eksperimen telah dijelaskan dalam sistem eksperimen pada gambar gambar 2 di atas.
5. Evaluasi Hasil
   Pada kegiatan ini dilakukan evaluasi dan analisis terhadap hasil estimasi proyek perangkat lunak dengan menggunakan sistem eksperimen penelitian yang telah dibangun. Hasil evaluasi dianalisis yakni dengan mengukur nilai *Effort.*

Submitted: **13/03/2020**; Accepted: **02/04/2020**; Published: **25/04/2020**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 4, Nomor 2, April 2020, Page 445-452
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI 10.30865/mib.v4i2.2083

3. HASIL DAN PEMBAHASAN

3.1 Analisis Komparasi Proyek dan Metode *Software Sizing*

Proyek yang akan dianalisis pada penelitian ini, antara lain:
1. Sistem Pengaduan Masyarakat POLDA Sumsel (Sumatera Selatan)
2. Sistem Kepegawaian kabupaten Ogan Komering Ilir (Sumatera Selatan)
3. RSFD *Activity System* pada Kantor BTN Pusat Jakarta

Dengan metode *software sizing* yang digunakan adalah 3 jenis metode yakni:
1. *Function Point Analysis* metoda FPA
2. FPA dengan Sugeno Fuzzy
3. FPA dengan Mamdani Fuzzy

Data-data proyek yang digunakan antara lain disajikan dalam:
1. Tabel *Line of Code* dari Setiap Proyek tersebut
2. Diagram Use Case setiap Proyek
3. Tampilan Sistem setiap Proyek

Untuk perhitungan *Line of Code*(LOC) setiap proyek yang dikomparasi dilakukan secara otomatis dengan menggunakan sebuah function penghitung LOC. Berikut adalah isi dari function LOC yang digunakan:

```
$fileCounter = array('php');
$totalLines = countLines('.', $fileCounter);
echo $totalLines." lines in the current folder<br>";
echo $totalLines - $fileCounter['gen']['commentedLines'] - $fileCounter['gen']['blankLines'] ." actual lines of code (not a comment or
blank line)<br><br>";

foreach($fileCounter['gen'] as $key=>$val) {
   echo ucfirst($key).":".$val."<br>";
}
echo "<br>";
foreach($fileCounter as $key=>$val) {
   if(!is_array($val)) echo strtoupper($key).":".$val." file(s)<br>";
}
function countLines($dir, &$fileCounter) {
   $_allowedFileTypes = "(html|htm|phtml|php|js|css|ini)";
   $lineCounter = 0;
   $dirHandle = opendir($dir);
   $path = realpath($dir);
   $nextLineIsComment = false;
   if($dirHandle) {
      while(false !== ($file = readdir($dirHandle))) {
         if(is_dir($path."/".$file) && ($file !== '.' && $file !== '..')) {
            $lineCounter += countLines($path."/".$file, $fileCounter);
         } elseif($file !== '.' && $file !== '..') {
            //Check if we have a valid file
            $ext = _findExtension($file);
            if(preg_match("/".$_allowedFileTypes."$/i", $ext)) {
               $realFile = realpath($path)."/".$file;
               $fileHandle = fopen($realFile, 'r');
               $fileArray = file($realFile);
               //Check content of file:
               for($i=0; $i<count($fileArray); $i++) {
                  if($nextLineIsComment) {
                     $fileCounter['gen']['commentedLines']++;
                     //Look for the end of the comment block
                     if(strpos($fileArray[$i], '*/')) {
                        $nextLineIsComment = false;
                     }
                  } else {
                     //Look for a function
                     if(strpos($fileArray[$i], 'function')) {
                        $fileCounter['gen']['functions']++;
                     }
                     //Look for a commented line
                     if(strpos($fileArray[$i], '//')) {
                        $fileCounter['gen']['commentedLines']++;
                     }
                     //Look for a class
                     if(substr(trim($fileArray[$i]), 0, 5) == 'class') {
                        $fileCounter['gen']['classes']++;
                     }
                     //Look for a comment block
                     if(strpos($fileArray[$i], '/*')) {
                        $nextLineIsComment = true;
                        $fileCounter['gen']['commentedLines']++;
```


Submitted: **13/03/2020**; Accepted: **02/04/2020**; Published: **25/04/2020**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 4, Nomor 2, April 2020, Page 445-452
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI 10.30865/mib.v4i2.2083

```
                    $fileCounter['gen']['commentBlocks']++;
                }
                //Look for a blank line
                if(trim($fileArray[$i]) == '') {
                    $fileCounter['gen']['blankLines']++;
                }
            }

        }
        $lineCounter += count($fileArray);
    }
    //Add to the files counter
    $fileCounter['gen']['totalFiles']++;
    $fileCounter[strtolower($ext)]++;
  }
 }
 } else echo 'Could not enter folder';
 return $lineCounter;
}
function _findExtension($filename) {
 $filename = strtolower($filename) ;
 $exts = split("[/\\.]", $filename) ;
 $n = count($exts)-1;
 $exts = $exts[$n];
 return $exts;
}
```

Fungsi tersebut menghitung file-file berikut yang tedapat dalam baris program setiap proyek, antara lain:

Tabel 1. File Yang Dihitung Otomatis Oleh Function LOC Counter

FILE YANG DIHITUNG OTOMATIS OLEH FUNCTION LOC COUNTER	
1. lines in the current folder	2. PHP
3. actual lines of code (not a comment or blank line)	4. JPG
5. TotalFiles	6. CGIECHO
7. BlankLine	8. CGIEMAIL
9. CommentedLines	10. CGI
11. Functions	12. CSS
13. Classes	14. PNG
15. CommentBlocks	16. JS:114 file(s)
17. php file	18. TTF:2 file(s)
19. HTACCESS	20. JPEG:2 file(s)
21. GIF	22. DB
23. SQL	24. ERROR_LOG
25. HTML	

Parameter yang dinilai dalam perhitungan FPA diperoleh dari perhitungan file di atas dan nilai berikut:

Tabel 2. Parameter Perhitungan TUFP

Parameter	Keterangan
External Input (EI)	*Interface* untuk memasukan data pada aplikasi.
External Output (EO)	*output* yang dihasilkan.
External Inquiry (EQ)	*query* terhadap data yang tersimpan
InternalLogical File (ILF)	logik penyimpan data.
ExternalInterface File (EIF)	komunikasi data pada parangkat/mesin yang lain.

3.2 Analisis Nilai Line Of Code *Software*

Tabel 3. Analisis LOC Sistem Pengaduan Masyarakat POLDA (DUMAS POLDA Sumsel)

Baris Koding Interface Sistem	
139563 lines in the current folder	PHP:231 file(s)
93415 actual lines of code (not a comment	JPG:49 file(s)
or blank line)	CGIECHO:3 file(s)
TotalFiles:596	CGIEMAIL:3 file(s)
BlankLines:15854	CGI:6 file(s)
CommentedLines:30294	CSS:14 file(s)
Functions:8393	PNG:70 file(s)
Classes:60	JS:114 file(s)
CommentBlocks:565	TTF:2 file(s)
0:php file(s)	JPEG:2 file(s)

Submitted: **13/03/2020**; Accepted: **02/04/2020**; Published: **25/04/2020**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 4, Nomor 2, April 2020, Page 445-452
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI 10.30865/mib.v4i2.2083

HTACCESS:6 file(s)	DB:2 file(s)
GIF:46 file(s)	ERROR_LOG:1 file(s)
SQL:4 file(s)	
HTML:43 file(s)	

Tabel 4. Perhitungan *Total Udjasted Function Point* (TUFP) Sistem Pengaduan Masyarakat POLDA (DUMAS POLDA Sumsel)

TUFP	**Complexity**					
	Low		**Medium**		**High**	
Decription	**Point**	**Bobot**	**Point**	**Bobot**	**Point**	**Bobot**
External Input (EI)	5	3	6	4	3	6
External Output (EO)	0	4	3	5	0	7
External Inquiry (EQ)	0	3	4	4	0	6
InternalLogical File (ILF)	0	7	596	10	0	15
ExternalInterface File (EIF)	0	6	0	7	0	10

Tabel 5. Analisis LOC Sistem Kepegawaian BKD Ogan Komering Ilir (SIMPEG BKD OKI Sumsel)

Baris Koding Interface Sistem
155442 lines in the current folder
108107 actual lines of code (not a comment or blank line)
TotalFiles:993
CommentedLines:27244
BlankLines:20091
Functions:6592
Classes:131
CommentBlocks:1569
0:php file(s)
HTACCESS:10 file(s)
SQL:1 file(s)
PHP:265 file(s)
HTML:45 file(s)
TXT:6 file(s)

Tabel 6. Perhitungan *Total Udjasted Function Point* (TUFP) Sistem Kepegawaian BKD Ogan Komering Ilir (SIMPEG BKD OKI Sumsel)

TUFP	**Complexity**					
	Low		**Medium**		**High**	
Decription	**Point**	**Bobot**	**Point**	**Bobot**	**Point**	**Bobot**
External Input (EI)	5	3	5	4	3	6
External Output (EO)	0	4	3	5	0	7
External Inquiry (EQ)	0	3	1	4	0	6
InternalLogical File (ILF)	0	7	993	10	0	15
ExternalInterface File (EIF)	0	6	0	7	0	10

Tabel 7. Nilai *Line Of Code* ProyekRSFD Activity System (RAS) BTN Pusat Jakarta

233695 lines in the current folder	XLSX:12 file(s)	MD:1 file(s)
152335 actual lines of code (not a comment or blank line)	PNG:215 file(s)	DOCX:23 file(s)
	CSS:92 file(s)	MMAP:4 file(s)
	GIF:106 file(s)	PPT:1 file(s)
TotalFiles:1671	JS:554 file(s)	PDF:7 file(s)
BlankLines:34512	SH:1 file(s)	RAR:3 file(s)
Classes:205	TXT:17 file(s)	SQL:2 file(s)
Functions:9150	HTM:34 file(s)	XLS:13 file(s)
CommentedLines:46848	SWF:5 file(s)	JPEG:1 file(s)
CommentBlocks:2848	JPG:76 file(s)	TTF:3 file(s)
	SWC:1 file(s)	EOT:1 file(s)
0:php file(s)	AS:2 file(s)	SVG:1 file(s)
HTACCESS:9 file(s)	PSD:4 file(s)	WOFF:1 file(s)
PHP:246 file(s)	GZ:1 file(s)	OTF:1 file(s)

Submitted: **13/03/2020**; Accepted: **02/04/2020**; Published: **25/04/2020**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 4, Nomor 2, April 2020, Page 445-452
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI 10.30865/mib.v4i2.2083

HTML:223 file(s)	GITIGNORE:1 file(s)	LESS:9 file(s) TIF:1 file(s)

Tabel 8. Nilai *Line Of Code* Proyek RSFD Activity System (RAS) BTN Pusat Jakarta

TUFP	**Complexity**					
	Low		**Medium**		**High**	
Decription	**Point**	**Bobot**	**Point**	**Bobot**	**Point**	**Bobot**
External Input (EI)	12	3	5	4	6	6
External Output (EO)	0	4	23	5	0	7
External Inquiry (EQ)	0	3	0	4	0	6
InternalLogical File (ILF)	0	7	1671	10	0	15
ExternalInterface File (EIF)	0	6	0	7	0	10

Keterangan:

1. Data *External Input* (EI) : Jumlah file *Interface* untuk memasukan data pada aplikasi.
   a. Low : Terdapat12 Modul untuk task input pada sistem yang mudah dibangun.
   b. Medium : Terdapat 5 Modul untuk task input pada sistem yang medium dibangun.
   c. High : Terdapat 6 Modul untuk task input pada sistem yang sulit dibangun.
2. Data *External Output* (EO): Jumlah *output* yang dihasilkan.
   a. Low : Terdapat 0 Modul untuk task input pada sistem yang mudah dibangun.
   b. Medium : Terdapat 23 Modul task input pada sistem yang medium dibangun.
   c. High : Terdapat 0 Modul untuk task input pada sistem yang sulit dibangun.
3. *External Inquiry* (EQ) : Jumlah query terhadap data yang tersimpan
   a. Low : Terdapat 0 Modul untuk task input pada sistem yang mudah dibangun.
   b. Medium : Terdapat 4 Modul untuk task input pada sistem yang medium dibangun.
   c. High : Terdapat 0 Modul untuk task input pada sistem yang sulit dibangun.
4. *Internal Logical File*(ILF) : Jumlah file yang terdapat dalam sistem.
   a. Low : Terdapat 0 Modul untuk task input pada sistem yang mudah dibangun.
   b. Medium : Terdapat 1671 Modul untuk task input pada sistem yang medium dibangun.
   c. High : Terdapat 0 Modul untuk task input pada sistem yang sulit dibangun.
5. *External Interface File* (EIF): Keterhubungan sistem dengan device lain (0 jika *standalone*)
   a. Low : Terdapat 0 Modul EIF untuk task input pada sistem yang mudah dibangun.
   b. Medium : Terdapat 0 Modul EIF untuk task input pada sistem yang medium dibangun.
   c. High : Terdapat 0 Modul EIF untuk task input pada sistem yang sulit dibangun.

3.2 Analisis Perhitungan Keseluruhan Sistem dan Metode

Pada sub bab ini dijelaskan analisis perhitungan sistem DUMAS POLDA SUMSEL, SIMPEG BKD OKI SUMSEL dan RAS BTN PUSAT Jakarta. Analisis yang dilakukan meliputi nilai *Line Of Codes* (LOC) dan *Effort* pada ketiga sistem yang diujicobakan. Berikut tabel 103 merupakan analisis keseluruhan sistem dan metode yang diujicobakan.

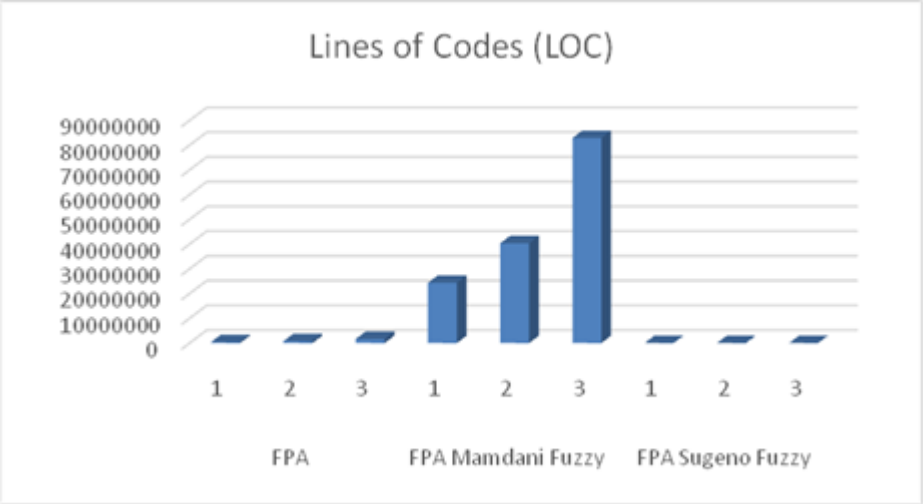
Tabel 9. Analisis Perhitungan Keseluruhan Sistem dan Metode

Metode	**No**	**Proyek**	**Lines of Codes (LOC)**	**Effort**
FPA	1	Sistem Pengaduan Masyarakat POLDA (DUMAS Sumsel)	467.994,24	35.145.647,83
	2	SIMPEG BKD OKI Sumsel	779.255,82	63.819.856,39
	3	RSFD *Activity System* (RAS) BTN Pusat Jakarta	1.666.732,87	1.553364.512
FPA Mamdani Fuzzy	1	Sistem Pengaduan Masyarakat POLDA (DUMAS Sumsel)	24.384.583,44	3.586.039.820
	2	SIMPEG BKD OKI Sumsel	40.350.873,56	6.464.547.040
	3	RSFD *Activity System* (RAS) BTN Pusat Jakarta	82,926,477,06	15.016.240.703
FPA Sugeno Fuzzy	1	Sistem Pengaduan Masyarakat POLDA (DUMAS Sumsel)	7.334,1294	271,736,5039
	2	SIMPEG BKD OKI Sumsel	7.281,087	269.438,5524
	3	RSFD *Activity System* (RAS) BTN Pusat Jakarta	11.625,603	465.828,7406

Submitted: **13/03/2020**; Accepted: **02/04/2020**; Published: **25/04/2020**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 4, Nomor 2, April 2020, Page 445-452
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI 10.30865/mib.v4i2.2083

Dari tabel di atas dapat dilihat bahwa nilai LOC dan *effort* pada sebuah sistem yang sama dapat berbeda jika dihitung dengan penggunaan metode FPA, FPA Mamdany fuzzy dan FPA Sugeno Fuzzy. Nilai LOC tertinggi dihasilkan oleh FPA Mamdany Fuzzy pada Sistem Pengaduan Masyarakat POLDA (DUMAS Sumsel). Begitu juga nilai *Effort* tertinggi, juga dihasilkan FPA Mamdany Fuzzy pada Sistem Pengaduan Masyarakat POLDA (DUMAS Sumsel). Sedangkan Nilai effort terendah sekaligus LOC terendah dihasilkan oleh FPA Sugeno Fuzzy. Hal ini dapat ditelusuri dari mekanisme perhitungan yang dilakukan oleh FPA Sugeno Fuzzy dimana metode ini sama sekali tidak menghitung nilai input ,output, file, query dan interface. Perhitungan FPA Sugeno fuzzy dilakukan dengan cara kasar hanya menilai dari kesulitan pembuatan sistem.



Gambar 2. Kurva Perbandingan Nilai LOC yang dihasilkan Tiga Proyek dengan Tiga Metode (FPA, FPA Mamadani Fuzzy dan FPA Sugeno Fuzzy)

Dari gambar 2 yang dihasilkan kurva di atas dapat dilihat bahwa Nilai LOC yang dihasilkan sangat berbeda dan sangat signifikan. Nilai tertinggi dihasilkan oleh FPA Mamdani Fuzzy. Terendah dihasilkan FPA Sugeno Fuzzy

4. KESIMPULAN

Adapun kesimpulan dari penelitian ini adalah bahwa nilai LOC dan effort pada sebuah sistem yang sama dapat berbeda jika dihitung dengan penggunaan metode FPA, FPA Mamdany fuzzy dan FPA Sugeno Fuzzy. Nilai LOC tertinggi dihasilkan oleh FPA Mamdany Fuzzy pada Sistem Pengaduan Masyarakat POLDA (DUMAS POLDA Sumsel). Nilai Effort tertinggi, juga dihasilkan FPA Mamdany Fuzzy pada Sistem Pengaduan Masyarakat POLDA (DUMAS POLDA Sumsel). Sedangkan nilai effort terendah dan LOC terendah dihasilkan oleh FPA Sugeno Fuzzy. Hal ini dapat ditelusuri dari mekanisme perhitungan yang dilakukan oleh FPA Sugeno Fuzzy dimana metode ini sama sekali tidak menghitung nilai input , output , file, query dan interface. Perhitungan FPA Sugeno fuzzy dilakukan dengan cara kasar hanya menilai dari kesulitan pembuatan sistem. Walaupun FPA Sugeno menghasilkan nilai effort yang rendah, tapi nilai yang dihasilkan lebih merata. Berdasarkan hasil penelitian ini, untuk menaikkan harga suatu proyek agar dihargai lebih tinggi Metode FPA Mamdani Fuzzy lebih direkomendasikan. Selanjutnya, peneitian ini perlu dilakukan eksperimen dengan jumlah proyek yang lebih banyak serta dapat dikembangkan kembali dengan menggunakan metode perhitungan LOC dan Effort lainnya.

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] 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.

[3] 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.

[4] B. Demir and L. Bruzzone, "A Novel Active Learning Method in Relevance Feedback for Content-Based Remote Sensing Image Retrieval," *Geosci. Remote Sensing, IEEE Trans.*, vol. 53, no. 5, pp. 2323–2334, 2015.

[5] 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.

[6] J. W. S. L. L. Tang, "Software, Improve Analogy-based Estimation, Effort Principal, Using Analysis, Components Weighting," in *Software Engineering Conference*, 2011, vol. 16, no. 2, pp. 11–12.

[7] Y. F. Li, M. Xie, and T. N. Goh, "A study of project selection and feature weighting for analogy based software cost

Submitted: **13/03/2020**; Accepted: **02/04/2020**; Published: **25/04/2020**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 4, Nomor 2, April 2020, Page 445-452
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI 10.30865/mib.v4i2.2083

estimation," *J. Syst. Softw.*, vol. 82, no. 2, pp. 241–252, 2009.
[8] M. Azzeh, "A replicated assessment and comparison of adaptation techniques for analogy-based effort estimation," *Empir. Softw. Eng.*, vol. 17, no. 1–2, pp. 90–127, 2012.
[9] N. H. Chiu and S. J. Huang, "The adjusted analogy-based software effort estimation based on similarity distances," *J. Syst. Softw.*, vol. 80, no. 4, pp. 628–640, 2007.
[10] A. Idri, F. A. Amazal, and A. Abran, "Analogy-based software development effort estimation: A systematic mapping and review," *Inf. Softw. Technol.*, vol. 58, pp. 206–230, 2015.
[11] A. K. Bardsiri and S. M. Hashemi, "Software Effort Estimation : A Survey of Well-known Approaches," *Ijcse*, vol. 3, no. 01, pp. 46–50, 2014.
[12] S. Kaur, S. Assistant, J. Kaur, S. Faculty, N. Chandigarh, and S. Singh, "Effect of Data Preprocessing on Software Effort Estimation," *Int. J. Comput. Appl.*, vol. 69, no. 25, pp. 975–8887, 2013.

Submitted: **13/03/2020**; Accepted: **02/04/2020**; Published: **25/04/2020**