Complexity of Web-based Application for Research and Community Service in Academic

Fitriana Fitriana[1], Sukarni Sukarni[2], Zulkifli Zulkifli[3]
Midwifey, Faculty of Biomedical Science, Universitas Aisyah Pringsewu, Lampung, Indonesia[1, 2]
Informatics Engineering, Faculty of Informatics & Technology, Universitas Aisyah Pringsewu, Lampung, Indonesia[3]

***Abstract*—Research data and community service in academic environment is a very important asset that must be managed properly. They have to be applied synergically in order to obtain as quality standards of higher education. A centralized web-based application designed for research data management and community service have been applied in terms of supporting the managerial of activities. To make the application suitable for users, it is necessary to estimate the size of the software built. This study aimed at measuring the consistency of the apps based on feature point analysis method which is owned by research and community service in Indonesia. Fourteen Modification Complexity Adjustment Factor (MCAF) were used for calculating a program scale with adequate precision. The main cost is determining the quality of application sequentially, which includes measuring the weighted value of feature point components, namely, Crude Function Points (CFPs), calculating the Relative Complexity Adjustment Factor (RCAF), and estimating the Function Point (FP) by using the formula itself. The results depict that the size of application was estimated about 18381 lines using FPA methods and achieved successful estimation with 2.2 percent of deviation.**

Keywords*—Application complexity; program scale; software size; function point analysis

I. Introduction

Research data and community service in an academic environment is a very important asset that must be managed properly. Research and community service is an essential factor of higher education in Indonesia. Both have to be applied synergically in order to obtain higher education quality standards. Some institutions have been focused on learning activities only and they neglect research and community service. The quality of their institution is not only seen from their learning quality, but also from the side of their research and community service. 16 of 24 performance indicators of higher education quality standards in Indonesia are determined by research and campus service to the community. These two factors must be considered properly [1]. Higher-education institutions have to pay more attention to the quality of their research and community service. They must be able to present data obtained from the result of their research along with provided contents, processes, assessments, facilities, infrastructure, and funds. If they able to manage all those data, then it can be said that quality of their institution is excellent [2], [3].

Research and community service are generally carried out by lecturers. A source of lecturers' work is not just research, but is a long-term work starting from the implementation of the research, responsibility and ranging from the baseline to achievement. If management of research data and community service is not running well, so then it will become problematic for lecturers as a researcher.

In order to support this matter, we have developed a centralized web-based application for management of research data and community service which can be used by the academic community such as lecturers and students. The key features of an application consist of a machine login module, a master consumer, a textbook, and journal publishing application. Therefore, the consistency of an application needs to be evaluated in order to determine the status of the product during and after the build. Otherwise, to ensure that the application can run properly, it is necessary to measure the compatibility of the application. Software or application measurement is appraisal method consisting of size, review and adaptation to enhance software development [4]. The tool that is widely used to measure functional size of the software work product is Function Point Analysis (FPA). FPA is the software which is relocated to the production application at project implementation. FPA technique is used to analyze the functionality of the software by using Unadjusted Function Point (UFP) [5]. FPA approach calculates the size of the program in Function Point (FP) units derived from five parameters, namely: Internal Logical File (ILF), External Interface File (EIF), External Input (EI), External Output (EO) and External Inquiries (EQ) which are divided into low, medium or large classes depending on the amount of Record Entity Types (RET), Data Element Form (DET) and File Type Reference (FTR) [6].

The quality of software product is a convenience from the user's point of view that can fulfil many characteristics such as: performance, usability, robustness, and security/safety. Meanwhile a high value of maintainability, availability and reusability were tested for specific quality attribute of software project [7]. Quality software is inseparable from both program code, the suitability of data entered and generated by the software process without errors. Hence, this study aims to estimate the size of web-based application project using FPA method. Subsequently, about 72 features were used to estimate the project size. The estimated size is compared with the actual project size.

II. Related Work

In 2018 [8], the author proposed on designing and implementing a system that makes it convenient for users in analyzing software functionality size based on FP method

referring to IFPUG CPM 4.3.1 standards. The system helps users to perform FP analysis in a faster and easier way without sacrificing accuracy. Whereas, in [9], the author builds the application chat messenger fellow android user through internal operation office. Their result shows that the application can translate automatically into a different language. It also shows the application can achieve the good performance in CPU, RAM, GPU and bandwidth usage.

Furthermore, [10] performs a good estimation of size of a mobile software project using FPA method about 4235 lines. When the estimated size is compared to the actual size, the deviation gained a highly successful estimation by 1.2%. Another study by [11], aims to identify improvements proposed for FPA to make the results generated more accurate. Their paper has presented eleven additional factors for FP analysis and suggested to address FPA as considered as an outdated method given by the great diversity and complexity of currently existing technology scenarios. Bani et. al [12] confirm that Random Forests has a better Confusion Matrix and scored better in both classification accuracy, and precision measures. The results of this work verify the validity of data mining in general and the applied technique in particular for software estimation.

Recently, in [13] it was investigated that the PSO feature selection can increase the accuracy or reduce the RMSE average value to 1552,999. Their result indicates that, compared with the original regression linear model, the accuracy or error rate of software effort estimation has increased by 3.12% by applying PSO feature selection. Also, [14] reviewed the machine learning models have been introduced to handle the flaws of parametric estimation models. These models also complement the modern project development and management.

Commonly, several earlier studies have presented only one type of feature that was used. Therefore, this study uses more features which differ from prior study. We will use the function point analysis method to perform software measurements complexity for research and community service application. Another objective of this research is to measure the consistency of the apps which we hope can provide more understanding about the level of complexity application from user's point of view.

III. Research Method

This research applies a Function Point Analysis (FPA) to measure the complexity of web-based application of research and services community in academic environment. It has been done after making several observations in [14]-[19], in which they also use FPA for measure software complexity.

The proposed stages include: measurement of crude function points (CFP), calculating the relative complexity adjustment factor (RCAF), and measurement function point (FP). The stages of software quality measurement are based on feature point analysis of research and community service application called E-LPPM application which is illustrated in Fig. 1.

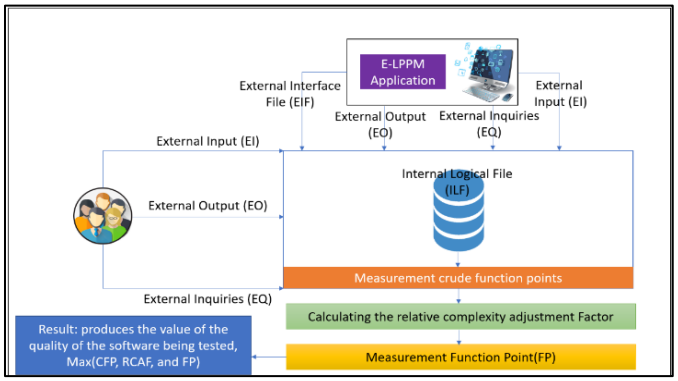

Fig. 1. Stages of Measurement Software.

Calculating the software's quality begins with the determining how to measure the success of software. It has been done with the decipherment of an application into its data and transactional functions. The data functions characterize the functionality afforded to the user by attending to their internal and external requirements in relation to the data, whereas the transactional functions explains the functionality delivered to the user in relation to processing this data by the application.

Second stage is calculating the CFP (Crude Function Points). The number of functional components of the system were first recognized and followed to calculate the complexity of quantization weight of every component including input, output, online queries, logic files, and the external interface.

The next stage is calculating the complexity of transcription factors of RCAF (Relative Complexity Adjustment Factor) for the project. RCAF is to calculate the complexity assessment of software system from several characteristics of subject. Rating scale from 0 to 5 is given to each subject that most affect the development effort required. Lastly, calculating Function Points by the formula:

$$FP = UFP \times VAF \quad (1)$$

The UFP is calculated as the complexities from parameters provided in the question where VAF is Value added Factor 0.65 + (0.01 * TDI), where TDI is Total Degree of Influence of the 14 General System Characteristics.

IV. Implementation and Results

Actual total size of the application which is specified by PHP source code files is 17988 lines in 300 files. Comment line inside the code is also taken in this counting because important comments are the key as the actual code line.

A. Calculating Crude Function Points (CFP)

The number of functional components of the system were first recognized and followed to calculate the complexity of quantization weight of every component.

FP calculation includes five types of software system components following: 1) the number of input applications; 2) the number of output applications; 3) the number of online query applications; 4) the application related to query against the data stored; 5) the number of logical files or tables which are involved; 6) the number of external interfaces. Then given

a weighting factor to every of the above components based on its complexity.

Table I shows the User Function (UF) results for compatibility feature of External Input (EI). There are two modules with low complexity, nine modules with medium complexity, and nine modules with high complexity. The result represents that processes data or control information comes from outside by external applications. As per Table I, this EI is considered as an "Average" complexity EI.

Table II describes Internal Logical File (ILF). There are 11 rows for overall with a low complexity ILF worth 4 points, an average ILF worth 2 points, and a high worth 5 points. ILFs represent data that is stored and maintained within the boundary of the application being counted.

Meanwhile, External Interface File (EIF) listed with their complexity levels is shown in Table III. Low complexity EIF is worth 4 points, an average EIF worth 2 points and high complexity worth 5 points. EIFs represent the data that application used or referenced. This means an EIF is counted for an application in an ILF in another application. As per Table III, EIF is considered as “High”.

Using the UF result and complexity of data, UFP value was calculated afterwards and depicted of 276 as shown in Table IV.

TABLE I. THE PERFORMANCE OF MODULES AND COMPLEXITY OF EXTERNAL INPUT (EI)

Module	Complexity
Login System Module	Low
Master User	Low
Upload Textbook Data	Medium
Upload Lecturer Journal Publish Data	Medium
Upload Lecturer Presenter Data	Medium
Upload Main Speaker Data (Keynote / Invited Speech)	Medium
Upload Data Intellectual Property Rights (IPR)	Medium
Upload Data on Grants or Research Funding / External Service Resources Ristekbrin Funds	Medium
Upload Lecturer Output Data	Medium
Upload Lecturer Research Data	Medium
Upload Lecturer Community Service Data	Medium
Input, Update, Delete Textbook Data	High
Upload Lecturer Community Service Data	High
Input, Update, Delete Textbook Data	High
Input Data Publish Lecturer Journal	High
Lecturer Speaker Data Input	High
Input Data Keynote Speaker (Keynote / Invited Speech)	High
Input Data Intellectual Property Rights (IPR)	High
Input Data Grant or Funding Research / Pengabmas External Source Ristekbrin	High
INPUT Data Output Lecturer	High
INPUT Data Advanced Research & Lecturer	High
INPUT Data Pengabmas Lecturer	High

TABLE II. INTERNAL LOGICAL FILE (ILF)

Module	Complexity
Table lppm_ user	Low
Table lppm_periode	Low
Table lppm_book	Low
The lppm_patent table	Medium
The lppm_item_pen research table	Low
Table lppm_journal	Medium
Table lppm_luaran	High
Table lppm_author	High
Table lppm_speaker	High
Table lppm_research	High
Table lppm_ext_monev_external	High

TABLE III. EXTERNAL INTERFACE FILE (EIF)

Module	Complexity
Excel Textbook Files	Low
Excel File Publish Lecturer Journal	Low
Excel File Speakers Lecturers	Low
Excel Main Speaker File (Keynote/Invited Speech)	Medium
Excel Intellectual Property (IPR) File	Low
Excel File Grants or Research Funding/External Service Resources Research Funds	Medium
Excel Lecturer Output File	High
Excel Lecturer Research File	High
Excel Lecturer Community Service File	High
Excel Textbook Files	High
Excel File Publish Lecturer Journal	High

After calculating the UFP, then the Relative Complexity Adjustment Factor (RCAF) value was computed and portrayed of 46 as total degree of influence which can be seen in Table V.

Finally, the Function Point (FP) of application was calculated by multiplying the UFP to RCAF as in (2).

$$FP = 276 \times 1.11 = 306.36 \quad (2)$$

To estimate the size of application within source code line, the FP value was multiplied by 60 as reference in the [20] and [21].

$$Size\_in\_LOC = 306.36 \times 60 \cong 18381 \quad (3)$$

Size of the project was estimated about 18381 lines using the FPA method. Accomplishment of the project, the actual size was 17988 lines. The estimated size then equaled to the actual size, depicted deviation of around 2.2%. This was considered as successful estimation. Nevertheless, there might be a larger dissimilarity between the estimate and the actual size since acceptable identification of the function type was a tough process.

TABLE IV. CALCULATION OF THE UFP VALUE

Software Parameters	Level of Complexity									Total CF
	Low			Medium			High			
	Count	Weight	Point	Count	Weight	Point	Count	Weight	Point	
	A	B	C=A*B	D	E	F=D*E	G	H	I=G*H	J=C+F+I
External Input (EI)	2	5	10	9	6	54	9	2	18	82
External Output (EO)	4	5	20	4	3	12	0	4	0	32
External in Query (EQ)	0	5	0	7	4	28	2	2	4	32
Internal Logic File (ILF)	4	6	24	2	7	14	5	10	50	88
External Interface File (EIF)	3	2	6	3	5	15	3	7	21	42
Unadjusted Function Point (UFP)										276

TABLE V. CALCULATION OF THE RELATIVE COMPLEXITY ADJUSTMENT FACTOR (RCAF)

System Characteristic	Value
Data Communication	3
Distributed Data Processing	3
Performance	3
Heavily used configuration	3
Transaction rate	4
Online Data Entry	3
End-user efficiency	3
Online Update	3
Complex Process	4
Reusability	3
Installation Ease	3
Operational Ease	3
Multiple Sites	4
Facilitate Change	4
Total Degree of Influence	46

As discussion matters, complexity of the identified function types was a particular matter although International Function Point Users Group (IFPUG) have been provided a broad guideline. Hence, it would be adequate to anticipate a disparity among the estimate and the actual size to a certain point.

V. CONCLUSIONS

In this report, Function Point Analysis software size estimation method was described at some point. The method was used on a web-based application developed by the author, the results were investigated. Applications developed in a web-based are commonly small in size and running properly when compared to regular software tasks like information systems. It is realized that the FPA method yields a perfect estimate for a small-scale web-based application. In extension to function point-based approach, the full data set requires to be prepared for future studies. The observation by end user can be used to make increasingly accurate estimates and classifications of web-based application. Thus, it will be of much support to the researchers' goal in the future for emerging languages and tools of programming tasks.

REFERENCES

[1] N. Sayidah et al., "Quality and University Governance in Indonesia," International Journal of Higher Education," vol. 8, no. 4, 2019, pp. 10–17.

[2] S. Eliot. "Factors Determining the Quality Management of Higher Education: A Case Study at A Business School in Indonesia. Jurnal Cakrawala Pendidikan. 38. 2019, pp. 215-227.

[3] S. Hidayat, "Peningkatan Mutu Penelitian di Indonesia Dalam Mengatasi Masalah Pendidikan", Bioilmi, vol. 4, no. 2, 2018, pp. 34-44.

[4] John Knight, Fundamentals of Dependable Computing for Software Engineers, 1st Edition. CRC Press, 2012.

[5] D. Longstreet, Function Points Analysis Training Course, vol. 2. Longstreet Consult. Inc, 2005.

[6] C. Angelica et al., "Agile Counting Process of Software Product Maintenance Size: A Statistical Analysis Processo de Contagem de Tamanho Agil de Manutenção de Produto de Software: Uma Análise Estatística". Espacios. Vol. 41(32), art.6, 2020.

[7] I. Charles, and N. I. Caesar. "Quantifying Software Quality in Agile Development Environment". Software Engineering. vol. 9, no. 2, 2021, pp. 36-44.

[8] A. & M. K. Irawati, "Measuring Software Functionality Using Function Point Method Based on Design Documentation," International Journal of Computer Science Issues, vol. 9, 2012.

[9] R. Sanjaya and A. S. Girsang, "Implementation application internal chat messenger using android system," 2017 International Conference on Applied Computer and Communication Technologies (ComCom), 2017, pp. 1-4.

[10] V. Tunalı, "Software Size Estimation Using Function Point Analysis-A Case Study for a Mobile Application," Mühendislik ve Teknoloji Sempozyumu, 2014. pp. 73-76.

[11] S. and M. A. K. Patwa, "A Study of Function Point Analysis: Some Suggestive Modifications," in Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) 2019, 2019, pp. 634–637.

[12] A. Banimustafa, "Predicting Software Effort Estimation Using Machine Learning Techniques," in 2018 8th International Conference on Computer Science and Information Technology, CSIT 2018, Oct. 2018, pp. 249–256.

[13] A. Latif, et al., "Comparative analysis of software effort estimation using data mining technique and feature selection", JITK, vol. 6, no. 2, 2021, pp. 167–174.

[14] R. K. G. Ripu Ranjan Sinha, "Software Effort Estimation Using Machine Learning Techniques," in Advances in Information Communication Technology and Computing, 2021.pp. 1-4.

[15] R. B. H. Eti Kapita, et al., "Analisis kualitas perangkat lunak menggunakan metode function point analysis (study kasus: Transaksi Pembelian di eBay)," Jurnal Informatika dan Rekayasa Perangkat Lunak, vol. 1, no. 1, 2019.

[16] B. N. Balaji, et al., “Software cost estimation using function point with non-algorithmic approach,” Global Journal of Computer Science and Technology, vol. 13, no. 8, 2013, pp. 1-5.

[17] D. Pratiwi, “Implementation of function point analysis in measuring the volume estimation of software system in object oriented and structural model of academic system,” International Journal of Computer Applications, vol. 70, no. 10, 2013, pp. 1-4.

[18] N. Rachmat, “Estimasi ukuran perangkat lunak menggunakan function point analysis-studi kasus aplikasi pengujian dan pembelajaran berbasis web,” Annual Research Seminar, vol.3, no.1, 2017, pp. 1-5.

[19] A. Yhurinda, P. Putri, and A. P. Subriadi, “Software Cost Estimation Using Function Point Analysis,” The 4th International Seminar on Science and Technology, no.1, 2018, pp. 79-83.

[20] H. Madhav C. and V. Kumar K.S., "A Method for Predicting Software Reliability Using Object-oriented Design Metrics," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 679-682.

[21] N. Marcheta, I. Hermadi, and Y. Nurhadryani. “Effort Estimation Modeling of E-Government Application Development Using Function Points Based on TOR and SRS Document. Journal of Information Technology and Its Utilization, vol.3, issue 1, 2020, pp. 5-8.