Economic Analysis Based on Software Cost Estimation Model on The Development of Telemetry Equipment to Support the Irrigation Modernization

Andri Prima Nugroho¹, Siwi Yuwanita Muliana¹, Murtiningrum¹, and Sigit Supadmo Arif¹

¹Smart Agricultural Research, Department of Agricultural and Biosystems Engineering, Faculty of Agricultural Technology, Universitas Gadjah Mada Jln. Flora No.1 Bulaksumur, Yogyakarta 55281.

ABSTRACT

Irrigation is one of the main supporters in the application of modern agriculture to ensure the availability of water as an input to agricultural production. Nowadays, irrigation modernization in Indonesia is intended to realize a participatory irrigation management system that is oriented towards fulfilling the level of irrigation services in an effective, efficient and sustainable manner. Information management with the support of equipment based on sensor networks and online databases is absolutely necessary to support the implementation of irrigation modernization. Sensor-based supporting equipment includes the Automatic Water Level Monitoring System (AWLMS) for estimating the flow of open irrigation canals based on the water level in conventional measuring buildings. However, the other important factor is the development of the supporting software and hardware as well as the performance testing of the equipment needed to prepare for the overall implementation in the next five to ten years. The purpose of this research was to conduct a study for the development of telemetry equipment (AWLMS) to support the Implementation of Irrigation Modernization. The method used is Software Cost Model Estimation for the Research and Development (R&D), providing solutions for solving budget estimation problems, especially when developing AWLMS so that budget estimates for equipment are not only physical aspects (hardware and software) of the tool but also cover maintenance and operational costs within a certain period of time. The estimation tools used to estimate is the Function Point method, a function-oriented approach to measuring software functionality to estimate the size of the software and then used for cost estimation and effort estimation needed to develop the system. Retrieval of data used is by measuring the weight of the complexity of the software. Data analysis carried out includes Crude Function Point (CFP) value, Relative Complexity Adjustment Factor (RCAF), Function Point (FP) value, effort value, Kloc (Kilo Lines of Code) value, final effort value, effort value to activity, and distribution to the workforce. The results of the study using the Function Point method obtained that the factor are the complexity of the software, the value of the measured software size, the level of difficulty at the software implementation stage, the estimated time required for software development, and the activities carried out by the workforce for software development.

Keywords: Irrigation modernization, AWLMS (Automatic Water Level Monitoring System), Software Cost Estimation Model (SCEM), Function Point.

1. INTRODUCTION

Agricultural productivity in Indonesia is still considered low in line with the development of population growth. One way to increase productivity is with the precision agriculture approach. Precision agricultural is one of the approaches that can be adapted to increase the agricultural productivity as well as reduce the cost and environmental impact by optimizing the resource usages through the utilization of appropriate technology [3]. One form of technology utilization is by monitoring the water discharge from the water level in the discharge measuring building. Irrigation is one of the main supports in modern agriculture because irrigation
will ensure the availability of water as an input for agricultural production.

Irrigation modernization is one of the policies and solutions for the future of irrigation in Indonesia. One way to implement irrigation modernization is by changing the manual method on paper-based to an automatic-digital one that is recorded in soft computing so that it will be more efficient and effective [5]. Irrigation modernization is an effort to realize an active irrigation management system oriented to the level of irrigation services in an effective, efficient and sustainable manner so that it can be fulfilled [7].

Previous study have developed a real time water level monitoring system using the internet for data transmission as well as monitoring water level and discharge integrated with a cloud server to support the irrigation modernization process in Indonesia, the tool is in the form of AWLMS (Automatic Water Level Monitoring System)[8]. However, during the one year testing and implementation in actual irrigation field, the other important factor is the development of the supporting software and hardware as well as the performance testing of the equipment needed to prepare for the overall implementation in the next five to ten years.

SCEM (Software Cost Estimation Model) is an important process in software development that is used to estimate the amount of cost, time, and the number of members required. There are several methods for estimating software costs, namely the algorithmic method in the form of Function Point Analysis (FPA) which is better [1].

In this study, this method is used to estimate the software cost of the AWLMS tool using the Function Point method. Function Point Analysis is an approach that focuses on the function of measuring software functionality to estimate the size of the software and is used to estimate costs and estimate the effort required for system development [4].

The purpose of this research was to conduct an analysis for the development of telemetry equipment (AWLMS) to support the Implementation of Irrigation Modernization. This research was conducted at the Faculty of Agricultural Technology, Universitas Gadjah Mada and Wadaslintang Irrigation System as an example for the case study.

2. MATERIALS AND METHODS

2.1. Software Cost Estimation Model

Economic analysis based on Software Cost Estimation Model is based on the development of supporting equipment for the application of irrigation modernization consisting of hardware and software. In its implementation, it requires a method for estimating budgets and costs which is expected not only to assess the equipment but also to R&D budgeting and field maintenance.

Figure 1. The framework for observing Irrigation Flow Discharge

The SIPASI software is used to monitor discharge on AWLMS in Primary Channel, AWLMS in Secondary Channel, AWLMS in Tertiary Channel, AWLMS in inlet/outlet tertiary area measurement. AWLMS contains power management module components equipped with solar panels and its controller, microcontroller unit equipped with data logging function, waterproof ultrasonic distance sensor, network communication module using GSM modem as gateway.

AWLMS will try to connect to an available WiFi network if successful then the water level data will be sent directly to the cloud server. If AWLMS is unable to successfully connect to the WiFi network, the water level data will be stored in local storage in AWLMS, to minimize data loss. When AWLMS has run all these programs, AWLMS will immediately go into sleep mode to save AWLMS power consumption. AWLMS works with power from the battery, this battery is also connected to the battery power sensor to monitor the power consumption of the AWLMS. The data will also be sent to the cloud server or stored on local storage.

2.2. Crude Function Point (CFP) Calculation

Crude Function Point (CFP) is a formula used to calculate the weight value of function point components associated with the software to be developed. The value of each complexity weight has been determined in Table 1.
There are 14 RCAF characteristics that are used to calculate the weight of software complexity. Characteristics that are made affect the level of difficulty associated with the implementation of the system. The table of RCAF characteristics can be seen in Table 2.

| No | Characteristics                        | Score                             |
|----|----------------------------------------|-----------------------------------|
| 1  | Data Communication Complexity Level     | (0, 1, 2, 3, 4, 5)                |
| 2  | Distributed Processing Complexity Level | (0, 1, 2, 3, 4, 5)                |
| 3  | Performance Complexity Level            | (0, 1, 2, 3, 4, 5)                |
| 4  | Configuration Complexity Level          | (0, 1, 2, 3, 4, 5)                |
| 5  | Frequency of Data Input Frequency       | (0, 1, 2, 3, 4, 5)                |
| 6  | Data Input Frequency Rate               | (0, 1, 2, 3, 4, 5)                |
| 7  | Level of Ease of Use for Users          | (0, 1, 2, 3, 4, 5)                |
| 8  | Data Update Frequency Rate              | (0, 1, 2, 3, 4, 5)                |
| 9  | Data Processing Complexity Level        | (0, 1, 2, 3, 4, 5)                |
| 10 | Degree of Reusable Program Code         | (0, 1, 2, 3, 4, 5)                |
| 11 | Level of Ease of Installation          | (0, 1, 2, 3, 4, 5)                |
| 12 | Ease of Operational Software (backup, recovery, etc.) | (0, 1, 2, 3, 4, 5) |
| 13 | Software Level Made For Multi Organizations / Companie s/Client | (0, 1, 2, 3, 4, 5) |
| 14 | Level of Complexity in Following Changes/Flexibility | (0, 1, 2, 3, 4, 5) |

Score Description:
0 = No effect
1 = Incidental influence
2 = Slightly influential
3 = Average effect
4 = Significant influence
5 = Very influential/essential

2.4. Function Point (FP) Calculation

In calculating the FP value, you can use the formula from the CFP (Crude Function Point) calculation which is then multiplied by the constant value and the RCAF (Relative Complexity Adjustment Factor) value which is shown in the following equation: Function Point which is then multiplied by the constant value and the RCAF (Relative Complexity Adjustment Factor) value which is shown in the following equation:
FP = CFP x (0.65+(0.01 x RCAF))  \hspace{1cm} (1)

Values 0.65 and 0.01 are constant values set by IFPUG (International Function Point Users Group).

**2.5. Effort Value Calculation**

In calculating the effort value, there are FP values that have been obtained and development based on time (Man/Hours) [5]. The calculation of the effort value can be shown in the following equation:

\[ \text{Effort}=\text{FP} \times 8.2 \]  \hspace{1cm} (2)

The value of 8.2 is the value of the productivity factor. Productivity Factor is the number of logical codes per function point.

**2.6. Kloc (Kilo Line Codes) Value Calculation**

Software size estimation in Kloc units can be obtained by multiplying the FP value by the Productivity Factor based on the programming language to be used. So to calculate the value of Kloc can be shown in the following equation:

\[ \text{Kloc} = \text{FP} \times 56 \]  \hspace{1cm} (3)

The value 56 is the Productivity Factor value based on the programming language that will be used.

**2.7. Final Effort Value Calculation**

The calculation of the final effort value is used for development based on Man/Hour [5]. The final effort value can be calculated using the following equation:

\[ \text{Final effort} = \frac{\text{FP}}{10} \times 22 \times 8 \]  \hspace{1cm} (4)

The value of 22 is the number of working days, the value of 8 is the number of hours worked, and the value of 18 is the constant value set by IFPUG.

**2.8. Distribution of Effort Value to Activities**

Each known effort value is distributed into each activity that has the aim of finding the estimated cost needed to build software [2].

The calculation of the effort value uses the following equation formula:

\[ \text{Effort} = \text{Final effort} \times \text{persentase} \]  \hspace{1cm} (5)

Cost calculation into each activity using the following equation:

\[ \text{Cost} = \text{effort} \times \text{payrate/hr} \]  \hspace{1cm} (6)

The distribution table for the effort value to activities can be shown in Table 3.

Table 3. The distribution table for the effort value to activities

| No | Activity                        | Persentase (%) | Payrate/hour |
|----|--------------------------------|----------------|--------------|
| 1  | Requirement                     | 1.6            | 17188        |
| 2  | Specifications                  | 7.5            | 17188        |
| 3  | Design                          | 6              | 10313        |
| 4  | Training And Support            | 1              | 10313        |
| 5  | Acceptance And Deployment       | 5.5            | 10313        |
| 6  | Integration Testing             | 7              | 10313        |
| 7  | Implementation                  | 52             | 34375        |
| 8  | Project Management              | 3.8            | 34375        |
| 9  | Configuration Management        | 4.3            | 10313        |
| 10 | Quality Assurance               | 0.9            | 10313        |
| 11 | Documentation                   | 8.4            | 10313        |
| 12 | Evaluation and Testing          | 2              | 10313        |

**2.9. Distribution to Labor**

Distribution to the workforce describes the overall costs that are adjusted to each worker in software development. So, each activity is distributed into the workforce as shown in Table 4.

Table 4. Distributed into the workforce

| No | Labor          | Activity                        | Persentase (%) |
|----|----------------|--------------------------------|----------------|
| 1  | Business Analyst | Requirement                   | 12             |
| 2  | Designer        | Specifications                  | 5              |
| 3  | Business Developer | Training and Support       | 11             |
|    |                 | Acceptance And Deployment     |                |
|    |                 | Integration Testing           |                |
| 4  | Programmer      | Implementation                 | 42             |
| 5  | Project Manager | Project Management            | 22             |
|    |                 | Configuration Management      |                |
| 6  | Quality Assurance | Quality Assurance            | 1              |
| 7  | Testing         | Documentation                  | 8              |
|    |                 | Evaluation and Testing        |                |
3. CURRENT RESULT AND DISCUSSION

The software used in the Automatic Water Level Monitoring System (AWLMS) tool is by using a server called SIPASI. SIPASI is an irrigation management information system with a basic concept from humans to machines and then to humans again, because it cannot be fully automated. SIPASI Server means a service/provider that provides information about irrigation information. On this SIPASI server there are several displays that can load the required data.

On the dashboard of SIPASI, a decision support system for Irrigation management where there are data such as irrigation water needs, river discharge, intake discharge, K factor, irrigation area, service area, number of tertiary plots, date, and five-point data. Then the widget in SIPASI for monitoring of discharge in a real-time manner shows data such as hourly intake discharge, hourly weir runoff, and hourly river discharge. These displays can be seen in Figure 2, Figure 3, and Figure 4.

Figure 2. SIPASI server display on AWLMS

Figure 3. Dashboard SIPASI

Figure 4. Widget in SIPASI for monitoring of discharge in a real-time manner

From the server it can be used to calculate the Crude Function Point (CFP) value. The value of Crude Function Point (CFP) can be obtained by means of the function value of each complexity weight obtained multiplied by each factor times its complexity. In the CFP calculation there are several components needed, namely External Inputs, External Outputs, Internal Logical Files, External Logical Files, and External Inquiry. The calculation of the CFP value can be seen in Table 5.

Table 5. The calculation of the CFP value

| Component                  | Complexity Weight | Low | Currently | High | Total CFP |
|----------------------------|-------------------|-----|-----------|------|-----------|
|                            | Count | Factor | Value | Count | Factor | Value | Count | Factor | Value |
| External Inputs            | 2     | 3      | 6     | 3     | 4      | 12    | 1     | 6      | 6     | 24       |
| External Outputs           | 0     | 4      | 0     | 2     | 5      | 10    | 1     | 7      | 7     | 17       |
| External Inquiry           | 1     | 3      | 3     | 1     | 4      | 4     | 1     | 6      | 6     | 13       |
| Internal Logical Files     | 1     | 7      | 7     | 2     | 10     | 20    | 1     | 15     | 15    | 42       |
| External Logical Files     | 0     | 5      | 0     | 0     | 7      | 0     | 0     | 10     | 0     | 0        |
| TOTAL                      |        |        |       |        |        |       |        |        |       | 96       |
Based on Table 5 of each component has been classified into three levels of complexity, namely easy, medium, and high. This CFP value will greatly affect the size or small value of the estimated cost of a software development. The larger the CFP value, the larger the software size. The calculation of the amount of complexity requires not only some of the features provided to the user, but to the operation of the system environment as well. At this stage, the calculation of the Relative Complexity Adjustment Factor (RCAF) in a software is carried out. The RCAF calculation uses 14 characteristics that are designed to influence the level of difficulty associated with system implementation. RCAF’s assessment based on 14 factors can be seen in Table 6.

### Table 6. The RCAF calculation

| No | Characteristics                                      | Score |
|----|------------------------------------------------------|-------|
| 1  | Data Communication Complexity Level                  | 1     |
| 2  | Distributed Processing Complexity Level              | 4     |
| 3  | Performance Complexity Level                         | 2     |
| 4  | Configuration Complexity Level                        | 1     |
| 5  | Frequency of Data Input Frequency                    | 2     |
| 6  | Data Input Frequency Rate                            | 5     |
| 7  | Level of Ease of Use for Users                       | 5     |
| 8  | Data Update Frequency Rate                           | 5     |
| 9  | Data Processing Complexity Level                      | 4     |
| 10 | Degree of Reusable Program Code                      | 3     |
| 11 | Level of Ease of Installation                        | 4     |
| 12 | Ease of Operational Software (backup, recovery, etc.)| 2     |
| 13 | Software Level Made For Multi Organizations/Companies/Perusahaan/Client | 5 |
| 14 | Level of Complexity in Following Changes/Flexibility | 4 |

**TOTAL** 47

In the total results obtained based on table 6, it can be seen that the AWLMS software has a fairly high complexity weight. This is because of the 14 factors that have a significant and very influential effect on the software being measured.

The calculation of the function point value uses the results of the CFP and RCAF calculations which can be calculated by equation. The value of the function point will affect the size of the estimated cost to be obtained in a software development from AWLMS.

The calculation of the effort value can be calculated with the result of the value of the existing FP then with equation. The effort value shows how long it takes to work on software development on the AWLMS tool that is carried out by the workforce.

The estimated size of the system in Kloc can be obtained by Equation. The Kloc value can be obtained by multiplying the FP value obtained by the constant value of the programming language productivity factor. The Kloc value indicates the code used by the programmer to create a program on the device. The more lines of code that are generated, the more complicated the programming will be.

This final effort value is used to calculate the effort value when distributing the effort value to activities. The calculation of the effort value can be done with equation. This final effort value is used as a parameter to calculate the estimated cost of software development.

The final effort value obtained in the previous stage is divided into 12 activities carried out in software development. Then the effort value is distributed in each activity using the payrate/hour that has been determined based on the salary standard by Kelly Service. The results of the distribution of the effort value into activities can be seen in Table 7.

### Table 7. Distribution of The Effort Value Into Activities

| No | Activity                        | Persentase (%) | Effort (final effort*persentase) | Payrate/hour | Cost (Effort*Payrate/hour) |
|----|---------------------------------|----------------|----------------------------------|--------------|---------------------------|
| 1  | Requirement                     | 1,6            | 16,821                           | 17188        | 289117,835               |
| 2  | Specifications                  | 7,5            | 78,848                           | 17188        | 1355239,854              |
| 3  | Design                          | 6              | 63,078                           | 10313        | 650527,745               |
| 4  | Training And Support            | 1              | 10,514                           | 10313        | 108421,291               |
| 5  | Acceptance And Deployment       | 5,5            | 57,822                           | 10313        | 596317,100               |
| 6  | Integration Testing             | 7              | 73,591                           | 10313        | 758949,036               |
| 7  | Implementation                  | 52             | 8,747                            | 34375        | 300673,802               |
| 8  | Project Management              | 3,8            | 2,996                            | 34375        | 102995,233               |
| 9  | Configuration Management        | 4,3            | 2,712                            | 10313        | 27972,693                |
Table 7: Cost Estimation of Developing the Automatic Water Level Monitoring System (AWLMS)

| No | Labor | Activity                        | Percent (%) | Cost (Rs)        | Total Cost (Rs) |
|----|-------|---------------------------------|-------------|------------------|-----------------|
| 1  | Business Analyst | Requirement Specifications       | 12          | 289,117,855     | 1,644,357,689   |
|    | Designer        | Design                          | 5           | 650,527,745     | 650,527,745     |
| 3  | Business Developer | Training and Support            | 11          | 108,421,291     | 1,463,687,427   |
|    |                | Acceptance and Deployment       |             |                  |                 |
|    |                | Integration Testing             |             |                  | 758,949,036     |
| 4  | Programmer      | Implementation                  | 42          | 300,673,802     | 3,006,73,802    |
| 5  | Project Manager | Project Management              | 22          | 1,029,95,233    | 1,309,67,926    |
|    |                | Configuration Management        |             |                  | 279,72,693      |
| 6  | Quality Assurance | Quality Assurance               | 1           | 975,791,62      | 975,792         |
| 7  | Testing         | Documentation                   | 8           | 500,90,636      | 266,933,218     |
|    |                | Evaluation and Testing          |             |                  | 2,168,42,582    |
|    |                | Testing                         |             |                  |                 |
| *Total* | |                                   | 100         | 4,458,123,599   | 4,458,123,599   |

4. CONCLUSION AND FUTURE WORKS

Based on the results and discussion of the research that has been carried out, it can be concluded that the estimated cost of developing software from the Automatic Water Level Monitoring System (AWLMS) with the Function Point method to support irrigation modernization is obtained quite simple results because the factor of the complexity weight measured is quite low.

Farther, to obtain the maximum estimation results of the development costs of the AWLMS software, it is expected to carry out an assessment from the tool to the software in more detail and analyze it using the Software Cost Estimation Model (SCEM) method which is more complex so that it can support the modernization of irrigation.

AUTHORS’ CONTRIBUTIONS

Andri Prima Nugroho provides advice on economic analysis, recommends system development and reviews scripts. Siwi Yuwanita Muliana prepares literature, creates economic analysis, collects research data, prepares and edits manuscripts. Murtiningrum designed the research environment settings. Sigit Supadmo Arif designed the research system.
ACKNOWLEDGMENTS

This study was supported by Department of Agricultural and Biosystems Engineering, Faculty of Agricultural Technology at the Universitas Gadjah Mada. Also, the author would like to thank the Smart Agriculture Research group of Agricultural and Biosystems Engineering UGM for the support.

REFERENCES

[1] Hendrawan., Pareza Alam Jusia., Errissya Rasywir., Yovi Pratama. 2019. Aplikasi Estimasi Biaya Pembangunan Software Menggunakan Metode Function Point Analysis (FPA) Dengan Platform Android. Seminar Nasional Sains dan Teknologi Informasi (SENSASI), 412-417.

[2] Kristi, Juliana., Siti Nur Azizah., Renny Sari Dewi. 2020. Estimasi Biaya Software FAS (Financing Analysis System) Menggunakan Metode Function Point (Studi Kasus Pada PT BPRS Lantabur Tebuireng). Jurnal Riset Komputer (JURIKOM), 7(1), 97-103.

[3] Nugroho, A P., L Sutiarso., T Okayasu. 2019. Appropriate adaptation of precision agriculture technology in open field cultivation in tropics. IOP Conf. Series: Earth and Environmental Science 355.

[4] Pratiwi, D. 2013. Implementation of Function Point Analysis in Measuring the Volume Estimation of Software System in Object Oriented and Structural Model of Academic System, 70(10), pp. 1-4.

[5] Priyonugroho, Anton. 2014. Analisis Kebutuhan Air Irigasi (Studi Kasus Pada Daerah Irigasi Sungai Air Kebar Daerah Kabupaten Empat Lawang). Jurnal Teknik Sipil dan Lingkungan, 2(3).

[6] Puspitasari, Reinindhi, Dian Nugraini, Renny Sari Dewi. 2019. Estimasi Biaya Perangkat Lunak Menggunakan Metode Function Point Analysis (Studi Kasus: Sistem BKD Universitas XYZ). Jurnal Riset Komputer (JURIKOM), 7(1),600-605.

[7] Soekrano, S. 2018. Penyempurnaan Sistem Pengelolaan Air Irigasi Menghadapi Irigasi Modern Di Indonesia. Indonesian Journal on Construction Engineering and Sustainable Development, 1(2).

[8] Wiyantanu, D. 2020. Perancangan Automatic Water Level Monitoring System (AWLMS) Berbasis IOT Untuk Saluran Irigasi Sekunder dan Tersier. Yogyakarta: Universitas Gadjah Mada