# Development of Analogy-Based Estimation Method for Software Development Cost Estimation in Government Agencies

Imam Kurniawan[1], Arry Akhmad Arman[2], Sukrisno Mardiyanto[3]
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
imam.kurniawan@students.itb.ac.id[1], arman@stei.itb.ac.id[2], sukrisno@stei.itb.ac.id[3]

***Abstract*—Cost estimation for software development in Government Agencies is still considered as a challenge. Owner Estimate Cost (OEC) should be estimated based on the specifications at the early stage of the procurement. There is no standard used by the government in formulating technical specifications. This affects the OEC value that tends to be highly subjective. Therefore, it is important to develop an estimation method that is able to represent software complexity at the early stage of procurement. The approach that can be considered in the early stages is the Use Case approach. UCP is a method to calculating effort algorithmically using use case complexity. However, UCP has a limitation, which only provides a level of fixed complexity and cannot handle uncertain conditions. In the other hand, Some previous research result shows that the analogy method has better performance than the algorithmic method. In this study, we developed the effort and cost estimation methods based on analogy by building a new dataset using the use case complexity parameters. Datasets are collected from 100 historical software projects data that have been built using the UCP method approach. Cost components for the project follow the current procurement regulations in Indonesia. Dataset evaluation for effort estimation using the proposed analogy method shows the best results of MMRE of 0.36 and PRED(0.25) of 0.57. MMRE shows the average difference of actual effort and estimated effort, whereas PRED (0.25) shows prediction level with error value smaller or equal to 25%. Furthermore, the estimated cost of the three software projects resulted in an average percentage deviation of 7.37%.**

***Keywords—Analogy-Based Estimation; Use Case Points; Software Cost Estimation; Effort Estimation***

## I. Introduction

IT implementation in government agencies has increasingly become a necessity, in public services, as well as other business services. This brings the emergence of many IT projects, particularly in software development projects [1]. However, software development projects often have high failure rate. According to a survey conducted by [2], only 29% information technology projects were successful in 2011 to 2015. According to [3], the reason of the software projects failure are insufficient requirements engineering, poor planning the project, suddenly decisions at the early stages of the project and inaccurate estimations were the most important reasons. The implementations of IT procurement projects in Indonesia present various problems. Several cases of corruption in the IT procurement occurred due to unsuitable software prices. According to a survey conducted in 30 Indonesian companies in 2013, most of the problems arose from the software procurement cost that reached 67% [4].

One of the important aspects to consider in software development projects is how to estimate cost in software procurement. Cost estimation calculated based on the requirement and specifications of the system to be built. In many cases of software procurement, early estimation must be done by the organization to get the approximate price of the software. Estimated costs at the early stages are usually performed based on requirements specifications from the perspective of users and stakeholders. Accurate estimation results are needed although the estimation is performed at an early stage. For example in government agencies, owner estimated cost (OEC) is the base to establish the highest bid limit and to assess the fairness of the offers, including cost details. OEC values is considered important in procurement, because if the price set too high, then it can potentially be a high cost for the government, whereas if it is lower from the normal price, then there is a potential failure in the procurement processes because no one would be interested. Therefore, it is essential to create an estimation method that can be performed at the early stage of procurement that is able to describe the detailed complexity of the system. The method requires the software functional parameters to measure the system complexity, which is obtained earlier.

Several approaches may be used to measure software complexities. The commonly used approaches are Line of Code (LOC) and Function Point (FP) [5]. The project size of LOC depends on the used programming language and cannot be determined before the project completes the implementation phase, whereas FP is based on data processing in measuring the system functions. Hence, FP might be subjective if used at the early stage of development and requires experts judgment to assess the level of functional complexities of the software that will be built [6]. Amongst the approaches to be considered at the early phase of a software procurement are Use Case approach [7]. Use Case is considered to be useful in preparing the system needs and also in communicating the design with the client [8]. The UCP

978-1-5386-0475-5/17/$31.00 ©2017 IEEE

method that was proposed by [9] are used to obtain software size based on the Use Case complexity of the system and perform the calculation of effort algorithmically. UCP has the potential due to its ability to accommodate the needs of the system from the business and technical perspectives and is able to be used at the early stage of the project, where the specification documents do not detail to describe the system that will be built [10]. However, the UCP method has a limitation in calculating the effort algorithmically, which only provides a level of fixed complexity and cannot handle uncertain conditions [11]. This affects the accuracy of the result of the estimated effort.

Analogy based estimation has been successfully used to estimate efforts and costs of software [12]. Previous studies [13], [14], and [15] found that the performance generated by analogy based estimation method outperformed the algorithmic method. In previous studies, the development of analogy method used LOC and FP parameters as the dataset, therefore it is not suitable to perform estimation at the early stages. In this study, we developed the analogy method using the UCP approach as a parameter to calculate software development effort. UCP is selected because the Use Case information can be obtained at the early stage so that the effort and cost estimation can be performed earlier.

## II. RELATED WORKS

### A. Technical Descriptions of Analogy-Based Estimation

Analogy based estimation is a process of problem-solving based on the events that resemble similar problems occurred in the past [12]. This method is considered as a systematic form of expert judgment because they often search for situations that are analogous to describe the experts' opinions. This method includes the introduction of a characteristic, that is the parameter values of the new project to be estimated, that would be used further as the basis to search similar projects. The effort value from a similar project (analog) is then used as the basis to estimate the effort of new projects. There are several advantages of analogy method, particularly for software estimation, namely: (1) Ability to be applied at the early phase of the software project, where there is little or no detailed information about the project [15], (2) Possess similar form of human reasoning in problem solving [16], and due to its dependency to historical data, thus it is (3) Able to handle uncertain domain and conditions that are difficult to model, and can be improved further when more detailed information is available [17]. An analogy is a general form of Case-Based Reasoning (CBR), which uses specific knowledge from previous experiences in implementing the situation of the problem so that it is able to produce the required solution with equations from previous issues [18].

### B. Software Complexity with Use Case Point Approach

The "Use Case Point" (UCP) method performs software measurements based on several parameters, including the weight of actor complexity and the use case of the system, technical factors, and the environment of system development [9]. Furthermore, Karner also proposed a method of calculating effort algorithmically based on the result of software complexity measurement. He calculated the software size based on the following parameters:

1. Use case complexity (UUCW)

UUCW is calculated based on the number of transactions in each of the use cases according to its weight.

TABLE I. WEIGHTED USE CASE

| Type | No of Transaction | Weight |
|---|---|---|
| Simple | <= 3 | 5 |
| Average | 4 to 7 | 10 |
| Complex | >7 | 15 |

2. Actor complexity (UAW)

UAW was calculated based on the interaction of the actors with the system, as described in Table II.

TABLE II. WEIGHTED ACTOR

| Type | Description | Weight |
|---|---|---|
| Simple | Represents another system with a defined application programming interface. | 1 |
| Average | An interaction with another system through a protocol and human interaction with a line terminal | 2 |
| Complex | Interacts through a graphical user interface | 3 |

3. Technical complexity factor (TCF)

TCF consists of thirteen non-functional technical factors that were calculated on the system that will be built. The total weight of TCF was calculated using Equation 1.

TABLE III. WEIGHTED TECHNICAL COMPLEXITY FACTOR

| Factor | Description | Weighting Factor |
|---|---|---|
| TCF1 | Distributed System | 2 |
| TCF2 | Response Adjectives | 2 |
| TCF3 | End-user Efficiency | 1 |
| TCF4 | Complex Processing | 1 |
| TCF5 | Reusable Code | 1 |
| TCF6 | Easy to Install | 0.5 |
| TCF7 | Easy to Use | 0.5 |
| TCF8 | Portable | 2 |
| TCF9 | Easy to Change | 1 |
| TCF10 | Concurrent | 1 |
| TCF11 | Security Features | 1 |
| TCF12 | Access for Third Parties | 1 |
| TCF13 | Special Training Requires | 1 |

$$TCF = 0.6 + (0.01 \sum_{i=1}^{13} w_i * impact_i) \qquad (1)$$

4. Environmental factor (EF)

EF consists of eight factors that assess the capabilities and experience of the team in the environment of the software that will be developed. EF is calculated using Equation 2.

TABLE IV. WEIGHTED ENVIRONMENTAL FACTOR

| Factor | Description | Weighting Factor |
|---|---|---|
| EF1 | Familiar with RUP | 1.5 |
| EF2 | Application Experience | 0.5 |
| EF3 | Object Oriented Experience | 1 |
| EF4 | Lead Analyst Capability | 0.5 |
| EF5 | Motivation | 1 |
| EF6 | Stable Requirement | 2 |
| EF7 | Part-time working | -1 |

| Factor | Description | Weighting Factor |
|---|---|---|
| EF8 | Difficult Programming Language | -1 |

$$EF = 1.4 + (-0.03 \sum_{i=1}^{8} w_i * impact_i) \quad (2)$$

The size of the software on the UCP method was calculated using Equation 3, whereas the calculation of the effort estimation was using Equation 4.

$$Size = (UAW + UUCW) * TCF * EF \quad (3)$$

$$Effort = Size * ER \quad (4)$$

Some previous studies obtained various ER values, that are: (1) 20 man-hours, using the data from three development software projects [9], and (2) 8.2 man-hours, using 10 software data projects in Indonesia [19].

## III. RESEARCH METHODOLOGY

Several steps that were conducted in this study are as follow:

1. Problem identification was performed using interviews with a number of government institutions in Indonesia to clarify the problems on software procurements
2. Literature study on the theories of procurements of goods/services in government institutions, analogy based estimation method and software complexity measurement.
3. Analysis and designing using analogy based estimation method, that are (1) Building a new dataset using data collection of previous software projects, (2) Proposing the KNN with mean and IDW interpolation technique for the effort calculation method, and (3) Designing the cost allocation model based on the regulations of goods and service procurement in government institutions in Indonesia.
4. Evaluation of the proposed effort estimation method using the built dataset, and the cost estimation using the real data of software development in government institutions.

## IV. PROPOSED WORK

### A. *The proposed estimation method*

Based on the studies from existing theories, we propose a solution to obtain the OEC value for software procurement in government agencies by developing analogy based estimation models using the use case complexity parameters. The software development was performed using the use case complexity as historical data and analogy parameter and also to calculate the effort. The proposed method is described in Figure 1.

### B. *Feature identification*

Software features are attributes or properties that describe the characteristics of the project. There are five parameters used as software features in the built analogy model, namely: UAW, UUCW, TCF, EF, and Size. These parameters are used in UCP method to assess the complexity of a software system. Features are derived from the results of the technical specifications compiled by the procurement team based on the user requirement identification.

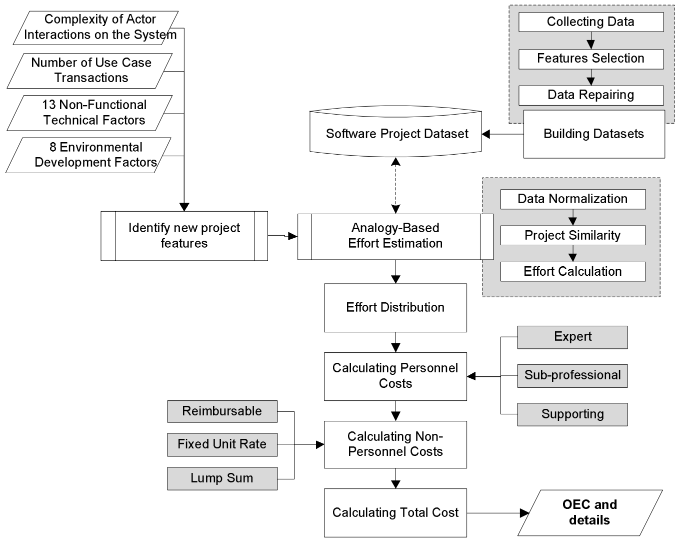


Fig. 1. Proposed Estimation Method

### C. *Development of Historical Project Dataset*

The proposed analogy model using the use case complexity approach seeks project similarities, therefore it is essential to collect the history of the software development projects that used the use case complexity metrics to build the dataset. The steps in the dataset building are (1) Collecting data, (2) Feature selection, and (3) Data repairing. The project data collection was performed from works of literature study of similar research. Historical data was obtained from several software development datasets using the use case point method.

TABLE V. SOFTWARE PROJECT DATASET

| No | Dataset | Features | Cases |
|---|---|---|---|
| 1 | Ochodek [20] | 15 | 14 |
| 2 | Popopic [21] | 11 | 32 |
| 3 | Silhavy [22] | 5 | 28 |
| 4 | Saroha [23] | 8 | 5 |
| 5 | Hariyanto [24] | 6 | 6 |
| 6 | Kenestie [25] | 7 | 7 |
| 7 | Kurniawan [26] | 14 | 8 |
| | | **Total Cases** | **100** |

Based on the collected dataset, we discovered that each of the datasets has a number of features that are not uniform to each other; therefore, not all features are used for datasets. Testing and improvements were necessary for the data that were collected. The test of outlier or the extreme data existence was performed. Data outliers will disrupt the overall data and may result in bias conclusions. In this study, we found 3 project data detected as data outliers. We eliminate the outlier data so that the remaining 97 projects case data will be used on the dataset.

TABLE VI. USED FEATURES FOR DATASET

| No | Feature | Type | Description |
|---|---|---|---|
| 1 | UAW | Continuous | Total weight of the complexity of the actors in the system |
| 2 | UUCW | Continuous | Total weight of the use case complexity |
| 3 | TCF | Continuous | Total value of the non functional technical factors |
| 4 | EF | Continuous | Total value of the environmental factor in system development |
| 5 | Size | Continuous | Total size of the software complexity |
| 6 | Effort | Continuous | Actual effort of the software development in man-hours unit |

### D. *Method of Data Normalization*

Data normalization was used to avoid large differences in data units that will be compared; a striking difference will produce invalid distance calculation. Data normalization is performed on the data that exist on each of the parameters in order to have a uniform weight value, that is from 0 to 1. The data normalization was performed using Equation 5.

### E. *Method to Determine Project Similarity*

The method used to determine projects similarity was distance measurements between the features on the estimated project and the analogy project. In this study, the Euclidean distance method is used to measure the distance of the software features, as in Equation 6.

$$M'_{ij} = \frac{m_{ij} - \min(m_j)}{\max(m_j) - \min(m_j)} \quad (5) \qquad D(s,p) = \sqrt{\sum_{i=1}^{n} (X_{si} - X_{pi})^2} \quad (6)$$

### F. *Effort Calculation Method*

At this stage, the K-Nearest Neighbor (KNN) algorithm was used to select projects which features are closest to the target project. Similar historical projects were selected as many as K projects and used as an analogy project to calculate the estimated effort. In this study, we analyzed and tested several methods to find the best method.

1. Nearest Project Adaptation Method

The method only selects one historical project that is most similar to the target project (K=1). The effort value of the historical project becomes the value of the estimated effort. This simple and easy to use method is very sensitive to outlier data [15].

2. KNN With Mean Adaptation Method

As the most widely used method, the estimated effort are calculated based on the average effort value as many as K similar projects. Studies using this method are [14] and [27]. The calculation of effort using this method is described in Equation 7.

3. KNN With Inverse Distance Weight (IDW)

IDW is a simple deterministic method by considering the points around it. The assumption of this method is that the interpolation value will be more similar to the closer sample data compared to the further ones. The weights will change linearly according to their distance to the sample data. In this study, the IDW method is used to calculate the effort on the built analogy model. The calculation of effort was performed by weighting the analogy project by considering the total distance of the features of each of the projects. The calculation of effort using KNN and IDW method are using Equation 8.

$$Yt = \frac{1}{k}\sum_{i=1}^{k} y_{ni} \quad (7) \qquad Yt = \frac{\sum_{i=1}^{n}\left(\frac{y_i}{d_i^p}\right)}{\sum_{i=1}^{n}\left(\frac{1}{d_i^p}\right)} \quad (8)$$

### G. *Effort distribution*

The allocation of effort distribution in this study follows the effort distribution of software development of small- and medium-scale projects in Indonesia, the study was conducted by [28].

### H. *Calculating cost*

The cost component used in software procurement is based on current regulations in Indonesia, which consists of personnel and non-personnel direct costs. In order to calculate personnel costs, firstly we determined the personnel involved in each of the activities, then converted the unit of salary personnel, from person per month to person per hour. Direct personnel costs were derived from the current wage standard in Indonesia, which consists of the cost of experts, sub-professionals, and supporting personnel. Non-personnel direct costs are the costs required to support the implementation of activities, that consists of three components, namely: reimbursable, fixed unit rate, and lump sum.

## V. EVALUATION

### A. *Evaluation of Dataset and Effort Estimation Method*

In order to monitor the proposed model performance, the estimation model was tested using the built-in dataset. The test was performed to monitor the accuracy of the estimation effort model using the proposed dataset. Aspects that were tested are Mean Magnitude Relative Error (MMRE) value and Prediction level (PRED 0.25) value of several effort estimation methods, i.e. : UCP [9], UCP2 [19], nearest project adaptation method [15] (1NN), analogy with mean adaptation [14] and [27] (3NN and 5NN), KNN with IDW method (3NNIDW and 5NNIDW). The test was performed using the k-fold cross validation method, where the dataset is randomly divided into four groups: K1, K2, K3, and K4. Testing was conducted in four times iteration, where each of the experiments is using the K-partition data as the data testing, whereas the remaining partitions are used as the training data. The MMRE calculation is shown in Equation 9. The results of the effort estimation model (MMRE and PRED(2.5)) are shown in the Table VII and Table VIII, respectively.

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|ActEffort_i - EstEffort_i|}{ActEffort_i} \quad (9)$$

TABLE VII. RESULTS OF THE MMRE VALUE TEST

| Test | UCP | UCP2 | 1NN | 3NN | 5NN | 3NNIDW | 5NNIDW |
|---|---|---|---|---|---|---|---|
| 1 | 0.92 | 0.50 | 0.71 | 0.50 | 0.57 | 0.49 | 0.52 |
| 2 | 0.64 | 0.44 | 0.32 | 0.58 | 0.46 | 0.43 | 0.37 |
| 3 | 0.75 | 0.31 | 0.30 | 0.40 | 0.43 | 0.26 | 0.29 |
| 4 | 0.69 | 0.41 | 0.32 | 0.31 | 0.32 | 0.28 | 0.26 |
| **Avg** | **0.75** | **0.41** | **0.41** | **0.44** | **0.45** | **0.36** | **0.36** |

TABLE VIII. RESULTS OF THE PRED(0.25) VALUE TEST

| Test | UCP | UCP2 | 1NN | 3NN | 5NN | 3NNIDW | 5NNIDW |
|---|---|---|---|---|---|---|---|
| 1 | 0.32 | 0.24 | 0.48 | 0.48 | 0.44 | 0.60 | 0.44 |
| 2 | 0.38 | 0.08 | 0.63 | 0.50 | 0.54 | 0.46 | 0.50 |
| 3 | 0.21 | 0.42 | 0.54 | 0.63 | 0.50 | 0.67 | 0.54 |
| 4 | 0.33 | 0.29 | 0.50 | 0.46 | 0.63 | 0.54 | 0.63 |
| **Avg** | **0.31** | **0.26** | **0.54** | **0.52** | **0.53** | **0.57** | **0.53** |

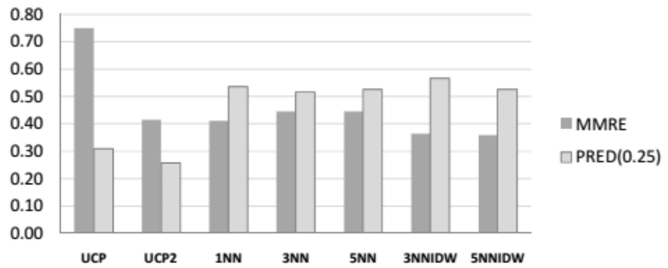


Fig. 2. Graph of MMRE and PRED(0.25) evaluation results

We found that 3NNIDW model displayed the best results with an MMRE value of 0.36 and PRED (0.25) of 0.57. Therefore, the 3NNIDW model will be used to evaluate cost estimation. Before to apply to testing this model with software project data, the proposed model needs to be validated. Validation is done by asking an expert opinion about the advantages of the proposed model. From the results of expert judgment, this model has good scored on some aspects i.e. easy to understand, the applicability levels to be implemented by the organization, and the performance of the proposed model. The ability of the model to estimate at the early stages of the project got scores an average. In general, this model can be implemented using software project data to be tested for evaluating cost estimation.

### *B. Cost Estimation Test*

At this stage, the analogy based estimation model was tested using the real software development projects data that were developed in the government institutions in Indonesia. The results of the evaluation are used to observe the cost deviation values between the cost estimation model and the actual cost.

1. Description of the tested data

The cost estimation test was performed on three government software projects. The characteristics of the test projects are shown in Table IX.

TABLE IX. CHARACTERISTICS OF THE TESTED PROJECTS

| *ID* | *Project* | *Tech* | *Tim* | *Mth* | *Actual cost (IDR)* |
|---|---|---|---|---|---|
| A | Climate Early Warning System | ASP, Microsoft SQL Server | 4 | 3 | 237.657.000 |
| B | Procurement Service Unit Information System | PHP, MySQL Server | 6 | 3 | 330.385.000 |
| C | Management of State Owned Property Information System | Delphi, MySQL Server | 5 | 2 | 190.248.000 |

2. Identification of the feature complexity and effort

The identification was performed using interviews with questionnaires and observations on the studied software to obtain the system complexity. After the property values were obtained, we calculated the effort estimation (effort value unit in man-hours) using the 3NNIDW method.

TABLE X. FEATURES COMPLEXITY AND ESTIMATED EFFORT

| ID | UAW | UUCW | TCF | EF | Size | Effort (3NNIDW) |
|---|---|---|---|---|---|---|
| A | 10 | 210 | 1.190 | 0.920 | 240.86 | 2275 |
| B | 18 | 290 | 1.165 | 0.845 | 303.20 | 3153 |
| C | 9 | 170 | 1.090 | 1.025 | 199.99 | 1718 |

3. Calculation of effort distribution

At this stage, the effort is distributed into software development activities. In this study, the distribution effort was using the results of research [28].

TABLE XI. EFFORT DISDRIBUTION

| Activities | % | A | B | C |
|---|---|---|---|---|
| **Development Phase** | | | | |
| Requirement | 1.6 | 36 | 50 | 27 |
| Specification | 7.5 | 171 | 236 | 129 |
| Design | 6.5 | 137 | 189 | 103 |
| Implementation | 52 | 1183 | 1640 | 893 |
| Testing and Integration | 7 | 159 | 221 | 120 |
| Deployment | 5 | 125 | 173 | 94 |
| **Ongoing Activities Phase** | | | | |
| Project Management | 3.8 | 86 | 120 | 65 |
| Configuration Management | 4.3 | 98 | 136 | 74 |
| Quality Assurance | 0.9 | 20 | 28 | 15 |
| Documentation | 8.4 | 191 | 265 | 144 |
| Training & Tech. Support | 1.0 | 23 | 32 | 17 |
| Evaluating & Testing | 2.0 | 46 | 63 | 34 |
| **Total Effort (Hours)** | **100** | **2275** | **3153** | **1718** |

4. Personnel costs

After the effort distribution for all software development activities was obtained, direct personnel costs can then be allocated based on the effort distribution of development as in Table XII (cost units in IDR).

TABLE XII. CALCULATION OF PERSONNEL COSTS

| Activities | A | B | C |
|---|---|---|---|
| **Development Phase** | | | |
| Requirement | 4.037.200 | 5.595.336 | 3.048.839 |
| Specification | 18.924.377 | 26.228.136 | 14.291.435 |
| Design | 15.139.502 | 20.982.509 | 11.433.148 |
| Implementation | 87.472.676 | 121.232.273 | 66.058.188 |
| Testing and Integration | 11.775.168 | 16.319.729 | 8.892.448 |
| Deployment | 7.726.876 | 10.709.022 | 5.835.233 |
| **Ongoing Activities** | | | |
| Project Management | 17.385.472 | 24.095.298 | 13.129.274 |
| Conf. Management | 7.233.317 | 10.024.976 | 5.462.504 |
| Quality Assurance | 1.513.950 | 2.098.251 | 1.143.315 |
| Documentation | 6.676.908 | 9.253.824 | 5.042.311 |
| Train & Tech. Support | 996.784 | 1.339.909 | 730.102 |
| Evaluating & Testing | 3.364.334 | 4.662.780 | 2.540.700 |
| **Total (IDR)** | **182.216.565** | **252.542.042** | **137.607.497** |

5. Non-personnel costs

Non-personnel costs are calculated based on the length of work performed. The results of the interviews concluded that there are seven types of expenditures which usually exist in the procurement activities of software consulting services.

6. Calculation of total OEC

Non-personnel direct costs generally do not exceed 40% from the total cost, except for a particular type of work. Furthermore, there are Value Added Tax of 10% from the total personnel and non-personnel direct costs. Calculation of the total cost of OEC is as shown in Table XIV.

TABLE XIII. NON PERSONNEL COST CALCULATION

| | Cost Type | A | B | C |
|---|---|---|---|---|
| 1 | Transportation | 1.360.000 | 1.360.000 | 1.360.000 |
| 2 | Office equipment rent | 12.926.818 | 17.915.852 | 9.762.159 |
| 3 | Operational cost | 18.000.000 | 18.000.000 | 12.000.000 |
| 4 | Office stationary cost | 10.800.000 | 10.800.000 | 7.200.000 |
| 5 | Computer & printer cost | 7.800.000 | 7.800.000 | 5.200.000 |
| 6 | Communication cost | 8.400.000 | 8.400.000 | 5.600.000 |
| 7 | Reporting cost | | | |
| | - Initial Report | 180.000 | 180.000 | 180.000 |
| | - Final Report | 360.000 | 360.000 | 360.000 |
| | - Manual O/M | 330.000 | 330.000 | 330.000 |
| | - CD | 25.000 | 25.000 | 25.000 |
| | - Flash disk | 200.000 | 200.000 | 200.000 |
| | **Total (IDR)** | **60.381.818** | **65.370.852** | **42.217.159** |

TABLE XIV. THE OEC CALCULATION

| Cost | A | B | C |
|---|---|---|---|
| Personnel (P) | **182.216.565** | **252.542.042** | **137.607.497** |
| Non Personnel (NP) | **60.381.818** | **65.370.852** | **42.217.159** |
| *NP percentage* | 25% | 21% | 23% |
| Total (P + NP) | 242.598.383 | 317.912.894 | 179.824.656 |
| VAT 10% | 24.259.838 | 31.791.289 | 17.982.465 |
| **Total OEC** | **266.858.221** | **349.704.183** | **197.807.122** |

7. Evaluation of cost estimation results

Evaluation was performed by monitoring the difference between the estimated and actual costs from software development project in order to obtain the deviation.

TABLE XV. ESTIMATE AND ACTUAL COST DEVIATION

| ID | Actual Cost | Estimation Cost | Deviation | % |
|---|---|---|---|---|
| A | 237.657.000 | 266.858.221 | 29.201.221 | 12.29 |
| B | 330.385.000 | 349.704.183 | 19.319.183 | 5.85 |
| C | 190.248.000 | 197.807.122 | 7.739.912 | 3.97 |

## VI. CONCLUSION

This research developed the analogy method for software development cost estimation. This approach builds the project dataset using the use case complexity approach and IDW interpolation technique for effort calculation. Dataset evaluation for effort estimation using the proposed analogy method shows the best results of MMRE of 0.36 and PRED(0.25) of 0.57. The estimation result of this approach is better than UCP algorithmic method using proposed dataset. Meanwhile, the evaluation of cost estimation from three software projects that were conducted in government agencies have percentages of deviation values of 12.29%, 5.85%, and 3.97% with an average of 7.37%. According to the result, this approach can be used to estimate the software development cost in any organization that need to conduct in early stage, such as government agency.

## REFERENCES


[1] M.A. Saputra and A.A Arman, “An analysis of software project management (case study: Government agencies),” ICITSI, Bandung, 2015.

[2] The Standish Group, “CHAOS Report,” The Standish Group, 2016.

[3] D.D. Galorath & M.W. Evans, “Software sizing, estimation, and risk management: When performance is measured performance improves”. Boca Raton, FL: Auerbach,2006.

[4] Tim Sharing Vision, “Pricing Issues in IT Procurement,” 2014.

[5] A.J. Albrecth, “Measuring Application Development Productivity,” IBM Appplication Development Symposium, 1979.

[6] M.R. Braz and S.R. Vergilio, “Software Effort Estimation Based on Use Cases,” IEEE, 2006.

[7] I. Jacobson, M. Christerson, P. Jonsson, dan G. Övergaard, “Object-oriented software engineering: A use case driven approach,” Reading, MA: Addison-Wesley, 1992.

[8] A. Cockburn, “Writing Effective Use Cases,” Boston, 2001

[9] G. Karner, “Resource Estimation For Objectory Projects,” Objective Systems, 1993.

[10] S. Kusumoto, M. Tsuda, K. Inoue, “On a Use Case Points Measurement Tool For Effective Project Management,” Workshop on ATGSE, 2007.

[11] J. Lee, W.T. Lee, and J.Y. Kuo,’Fuzzy Logic As A Basic For Use Case Point Estimation,” IEEE International Conference on Fuzzy Systems, 2011.

[12] J. Keung, “Software Development Cost Estimation using Analogy: A Review,” ASWEC 2009, April, Gold Coast, Australia, 2009.

[13] S. Vicinanza, T. Mukhopadhyay, and M.J. Prietolla, “Software Effort Estimation: An Exploratory Study Of Expert Performance,” Journal of Information System Research, 2:243-262, 1991.

[14] M. Shepperd and C. Scofield, “Estimating Software Project Effort Using Analogies,” Software Engineering, IEEE Transactions, 1997.

[15] F. Walkerden and R. Jeffery, “An Empirical Study Of Analogy-Based Software Effort Estimation,” Empirical Software Engineering, 1999.

[16] L. Angelis and I. Stamelos, “A Simulation Tool For Efficient Analogy Based Cost Estimation,” IEEE Empir Software Eng, pp.35-68, 2000.

[17] H. Burkhard and M.M. Richter, “On The Notion of Similarity in Case Based Reasoning and Fuzzy Theory,” Springer Verlag, 2001.

[18] A. Aamodt and E. Plaza, “Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approach,” Artificial Intelligence Communications, 7:39-59, 1994.

[19] A.P. Subriadi, Sholiq, P.A. Ningrum, “Critical Review of Effort Rate Value in UCP Method For Estimating Software Development Effort,” Journal of Theoretical and Applied Information Technology, 2014

[20] M. Ochodek, J. Nawrocki, and K. Kwarciak, “Simplifying Effort Estimation Based On UCP,” Journal of Information and Software Technology, 2011.

[21] J. Popovic, D. Bojic, and N. Korolija, “Analisys of Task Effort Estimation Accuracy Based on UCP Size,”. IET Software, 2015.

[22] R. Silhavy, P. Silhavy, and Z. Prokopova, “Algorithmic Optimisation Method for Improving Use Case Points Estimation,” PLOS ONE, 2015.

[23] M. Saroha and S. Sahu, “Software Effort Estimation using E-UCP Model,” ICCCA, 2015.

[24] M. Hariyanto and R.S. Wahono, “Estimated Software Development Project with Fuzzy Use Case Points,” Journal of Soft Engineering, 2015.

[25] I.D. Kenestie, Sholiq, "Determining The Effort Rate Rate on UCP Method for Effort Effort Projects Software Development in Education Field," Jurnal Teknik POMITS, 2013.

[26] W. Kurniawan, Sholiq, and T. Sutanto, "Determination Of Effort Rate On Effort Estimation UCP Method For The Development Of The Government Website Software," JSIKA, 2013.

[27] E. Mendes, N. Mosley, and S. Counsell, “A Replicated Assesment of the use of Adaptation rules to Improve Web Cost Estimation,” ISESE, 2003.

[28] P.L. Primandari, and Sholiq, “Effort Distribution to Estimate Cost in Small to Medium Software Development Project with Use Case Points,” ISICO, 2015.