NAYA as a Tool of Software Cost Automatic Analysis

Shofiyah Al Idrus
Informatics Education
State University of Malang
Malang, Indonesia
shofiyah.ft@um.ac.id

Wahyu Nur Hidayat
Informatics Education
State University of Malang
Malang, Indonesia
wahyu.nur.ft@um.ac.id

Achmad Hamdan
Informatics Education
State University of Malang
Malang, Indonesia
achmad.hamdan.ft@um.ac.id

Abstract— **A software project failure can be caused by an inappropriate and ineffective management approach in developing software. One of the failures of software projects that have the greatest capacity is the error in estimating financing and scheduling software. Representations of software financing calculations with good accuracy can be represented by the COCOMO II and Use Case Point (UCP) methods. The COCOMO II and UCP methods have a long steps calculation process with a lot of data collection, this obstacle is the main factor in the cost calculation process in the software and also, the lack of information about software cost calculation methods and techniques makes this section often overlooked by software developers. NAYA application developed by researchers has guided developers to write supporting data on the calculation components that have been provided. This application was developed with an iterative development model for implementation because application development requires feedback from the user as an improvement of specific functions by the system. The advantages NAYA application with COCOMO II in this application are simple data that must be prepared by the user, layout of calculations with minimum wages, and a comprehensive presentation of calculation results. Cost calculation with UCP has the advantage of implementing the Re-UCP method so that the components used are very comprehensive and complete so that it can remind UCP value and effort. NAYA applications can help software developers to calculate and arrange to finance for a software project quickly and accurately.**

Keywords—COCOMO II, cost estimation, UCP

I. INTRODUCTION

A software project failure can be caused by an inappropriate and ineffective management approach in developing software [7][8]. One of the failures of software projects that have the greatest capacity is the error in estimating financing and scheduling software. Software is considered a failure if it exceeds a percentage of 50% of the planning costs and over a predetermined schedule. Therefore, the design of software development requires a process of estimation of size. Estimation is done as a consideration in determining the cost, scheduling, and software development efforts. The accuracy of software estimation and measurement is important for a project [7][8].

Estimated financing is also a benchmark for decisions on whether a project is feasible or not and how much relevant value of software is generated within the specified parameters [12]. Accurate and reliable estimates are difficult to obtain due to the lack of detailed information about the system to be developed, in the early stages of software development [6]. If we give a project more resources than it needs, then the project is likely to cost more, take longer time to deliver than necessary, and delay the use of resources on the next project [8][11].

There are several methods, tools, and techniques that can be put into practice to estimate the cost of the software ranging from traditional modeling to modern-day modeling. The data used in estimating software financing also varies. Each of the supporters of software calculations will complement one another [8][11]. Representations of software financing calculations with good accuracy can be represented by the COCOMO II and Use Case Point (UCP) methods. Both of these methods have complete supporting data both owned by the system process and the systems interact with the user.

The COCOMO II and UCP methods have a long steps calculation process with a lot of data collection, this obstacle is the main factor in the cost calculation process in the software. Also, the lack of information about software cost calculation methods and techniques makes this section often overlooked by software developers.

Based on the problems that have been described, researchers developed an application to help software developers to calculate software costs quickly and accurately. NAYA application developed by researchers has guided developers to write supporting data on the calculation components that have been provided. This application is also equipped with documentation of the results of cost calculations that can be used by developers to discuss with clients related to the project to be developed.

II. METHODOLOGY

Based on the introduction this research is referred to as software development research with an application product as a result. This research used the iterative development model for implementation because application development requires feedback from the user as an improvement of specific functions by the system [9].

A. Iterative Model

The phases of the iterative model have explained in Fig. 1. Moreover, the description of phases in the iterative model explained in Table 1.

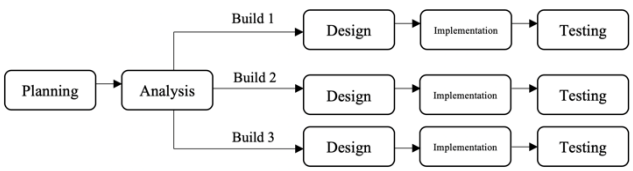

Fig. 1. Iterative Model

978-1-7281-8131-8/20/$31.00 ©2020 IEEE

TABLE I. DESCRIPTION OF THE ITERATIVE MODEL PHASES

Phases	Description
Analysis	User needs are defined to classify functions that have done by this application. User needs are supported by literature studies with COCOMO II and UCP materials.
Design	The development team would be designing the application with wireframes and prototypes.
Implementation	The implementation in these phases has done with a coding program for the result of an application website.
Testing	The testing process in these phases has done with a technique system testing, user testing, and usability testing. System testing is focused on testing the system with system code, whereas the user and usability testing are focused on the satisfaction and related function with user needs. We have done tests with the student target as many as 30 peoples.

III. RESULT AND ANALYSIS

The aims of the development application are how those can help users to perform accurately cost calculations for software development. The accuracy of the calculations in this application is described in the functions and supporting indicators in each section. The functions or features possessed by this application are divided into two broad lines, namely the cost calculation feature with COCOMO II and the cost calculation feature with UCP. Cost calculations were generating automatically with this application.

A. Cost Calculations with COCOMO II

Before cost calculations the estimation we needed data flow diagrams (DFD) as supporting data [4][7]. The application will calculate the cost based on the process in our system. Estimated cost calculation with COCOMO II have done in steps i.e [4][3][16]:

- Collecting data process from DFD
- Calculating Unadjusted Function Point (UFP) with EI, EO, EQ, ILF, and EIF components.
- Calculating complexity weights with DET, RET, and FTR.
- Transform the SLOC value into KSLOC value.
- Calculating Scale Factors (SF),
- Calculating Effort Multipliers (EM).

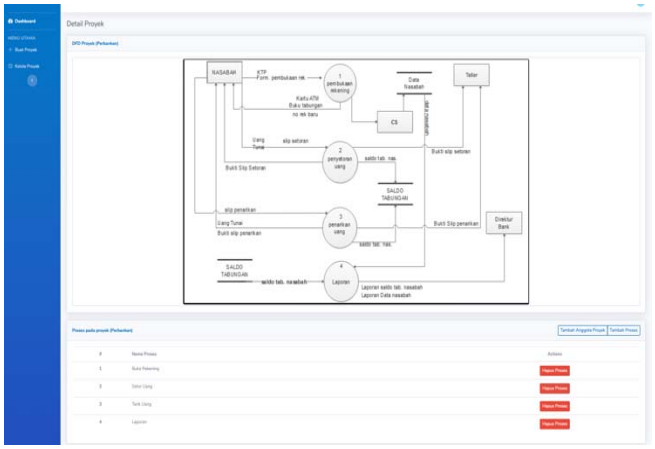

Fig. 1. Collecting data process

In our application, COCOMO II is calculated according to the stages outlined. This application with COCOMO II calculations has project management features so that team members can collaborate and complete data [10].

The advantages of calculating COCOMO II in this application are available salary layouts required by the project developer to provide an accurate calculation. NAYA application will provide predictions in the form of the duration of the project, the large number of people who have to work on the project, and the number of costs required for the project [10].

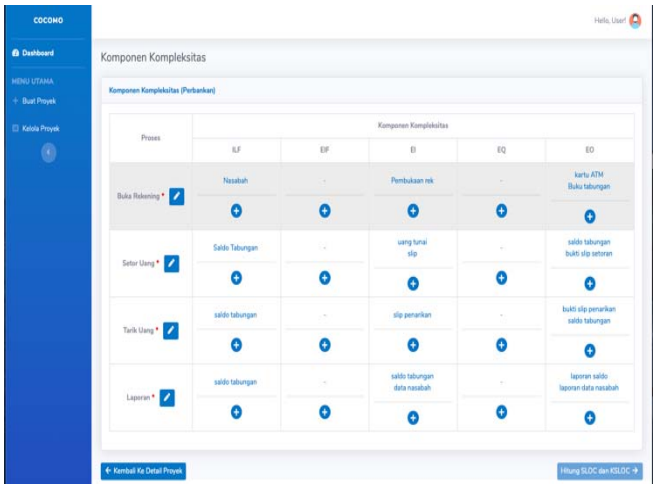

Fig. 2. Calculating UFP

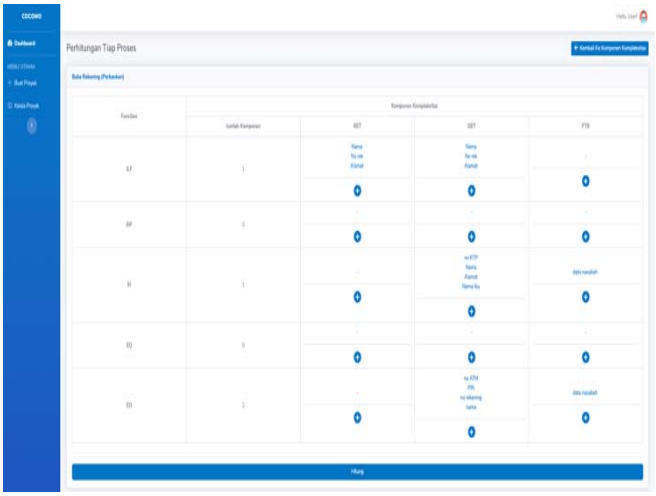

Fig. 3. Calculating Complexity Weight

Fig. 4. Result of Complexity Weight Calculate

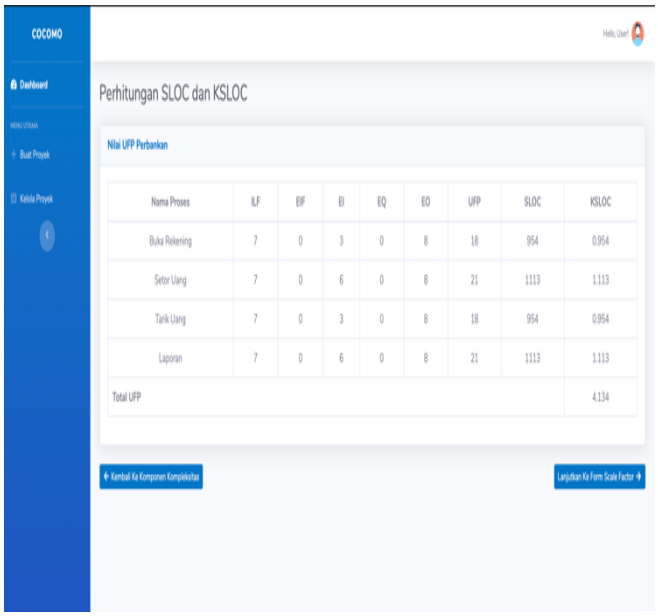

Fig. 5. Transformation SLOC to KSLOC

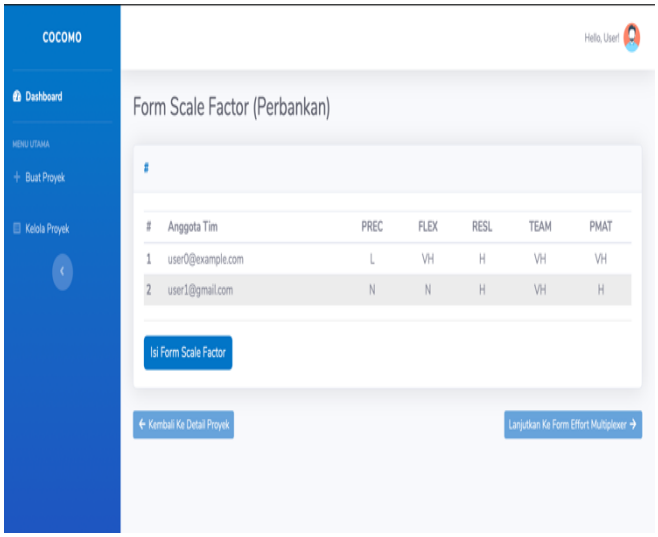

Fig. 6. Calculating SF

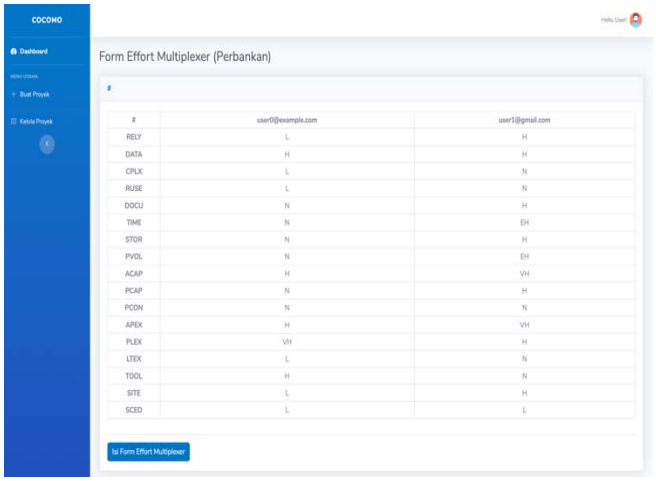

Fig. 7. Calculating EM

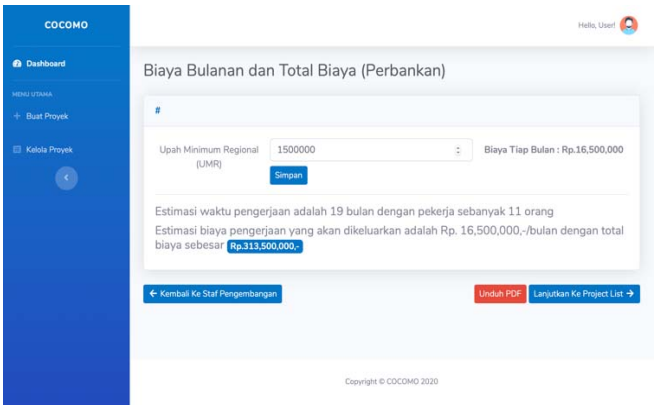

Fig. 8. Result of Cost Estimation

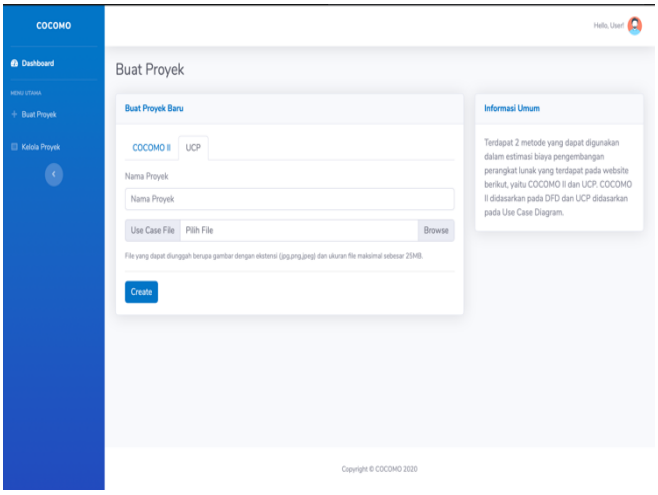

Fig. 9. Collecting data from use case diagrams

Cost estimation calculation can be calculated by the following formula [1][3].

$$PM = A * (Size)^{e} * EM \quad (1)$$

$$TDEV = C * (PM)^{(D+0{,}2*(E-B))} \quad (2)$$

$$Average\ Staff = \frac{PM}{TDEV} \quad (3)$$

$$Monthly\ Cost = Average\ staff * regional\ minimum\ wage\ (in\ our\ city) \quad (4)$$

$$Cost\ total = monthly\ free * TDEV \quad (5)$$

B. Cost Calculations with UCP

Cost calculations with UCP are needed to use case diagrams as supporting data [8]. The application will calculate the cost based on actors or users and activity on the developed system.

The first calculation has done by getting the user actor weight (UAW) value with the formula [2][5].

$$UAW = actor\ count * actor\ weight \quad (6)$$

UCP calculation based on revised UCP (Re-UCP) because this method has shown improvements in estimating the efforts for a software development project with minimum calculation error so that increasing the accuracy of calculations. UAW in Re-UCP is categorized into simple, average, complex, and critical with sequential weight parameters of 1,2, 3, and 4 [13].

Fig. 10. UAW calculating process

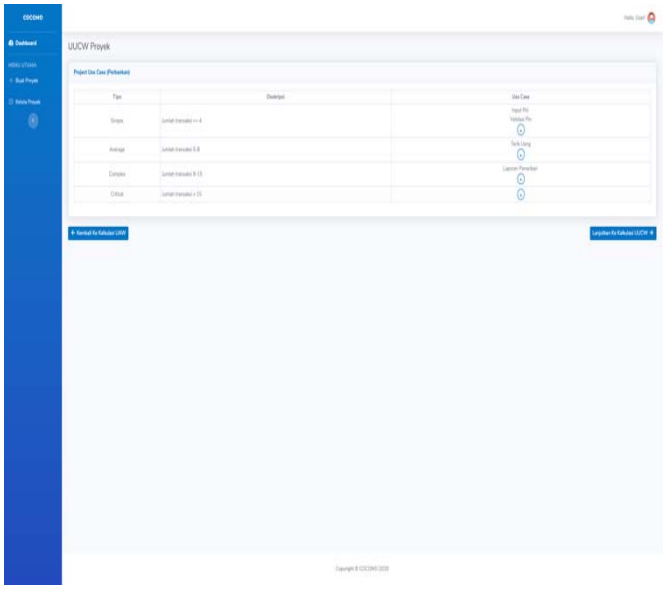

Fig. 11. UCW calculating process

The UAW value results will be continued in the Use Case Weight (UCW). UAW dan UCW is based to obtain Use Case Point (UCP) value [2][5].

UCW = use case count * use case weight……. (7)

UCW in Re-UCP is categorized into simple, average, complex, and critical with sequential weight parameters of 5,10, 15, and 20. The numbers of transactions in the use case scenario will affect the existing estimates. The transactions with large numbers in a use case scenario will affect UCW to be complex and increase UCP and otherwise. Furthermore, transactions in large numbers can be reduced without affecting business processes. NAYA application has been shown the value of UAW and UCW calculations automatically as Fig. 12.

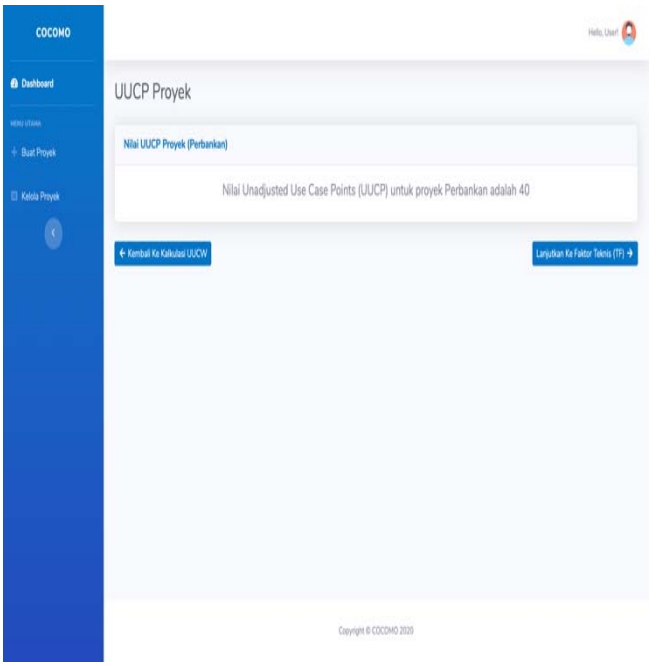

Fig. 12. UCP calculating process

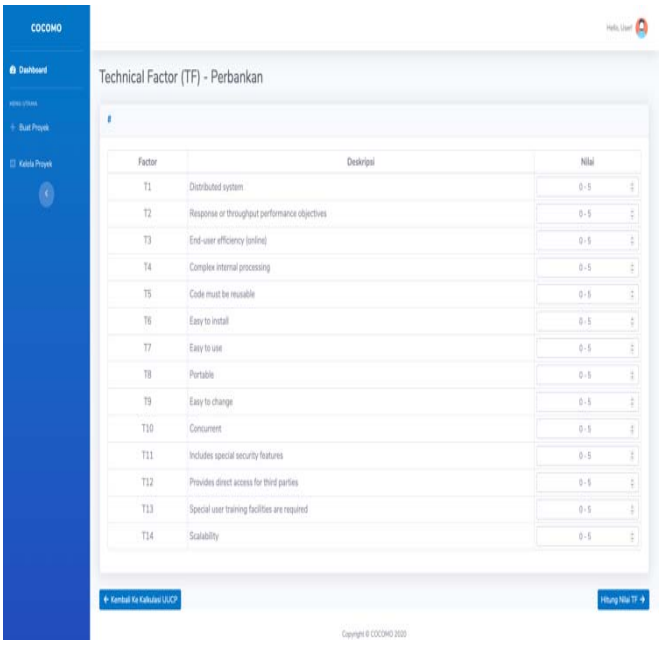

Fig. 13. TCF calculating process

This application the UCP value obtained will be recalculated with the Technical Complexity Factor (TCF) and Environmental Complexity Factor (ECF) values to produce the nominal costs needed in the project being developed [5].

In our development application, we have used fourteen TCF factors, we have added a scalability factor in the calculating process. The scalability factor can improve the value of UCP points and effort [10][11]. TCF evaluated by the software development team subjectively depends on the perception of the complexity of the project simultaneously. The value of TCF factors we can give value among 0-5 numbers. If you have a perception with a doubtful value, the development team can give a value of 3.

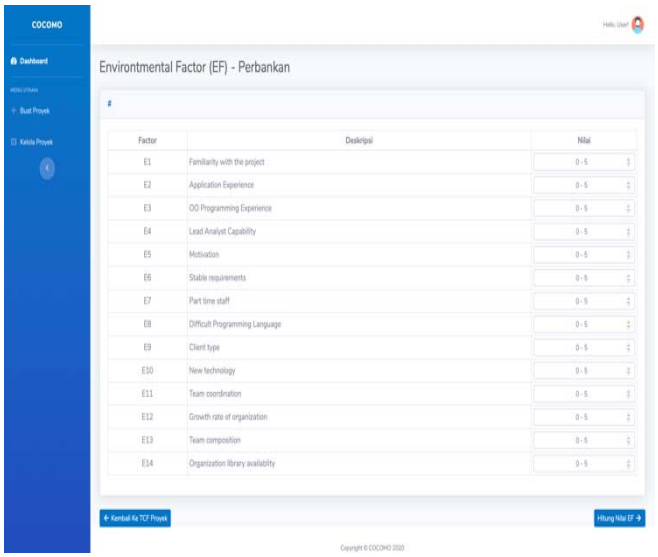

Fig. 14. ECF calculating process

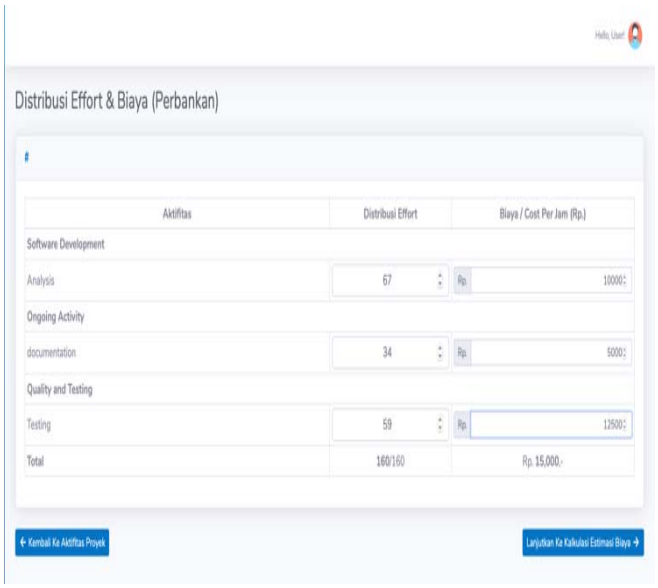

Fig. 15. Effort Calculating

Each value obtained in the TCF will be multiplied by the weighted value for each factor and summed to obtain the total value in the Technical Factor. The total value on the TCF is used to obtain the value of the TCF in the following formula.

$$TCF = 0{,}6 + (0{,}01 * \text{Technical Total Factors}) \ldots. (8)$$

Furthermore, the ECF in our development application was using fourteen factors, which we believe are strongly related to the development environment and can help get a more accurate and precise result. The value of ECF factors we can give value among 0-5 numbers [14].

Each value obtained on the Environmental Factor will be multiplied by the weighted value on each factor and added up to obtain the total value on the Environmental Factor. The total value on Environmental Factor is used to obtain the value of the ECF with the following formula.

$$ECF = 1{,}4 + (-0{,}03 * \text{Environmental Total Factors}) \ldots. (9)$$

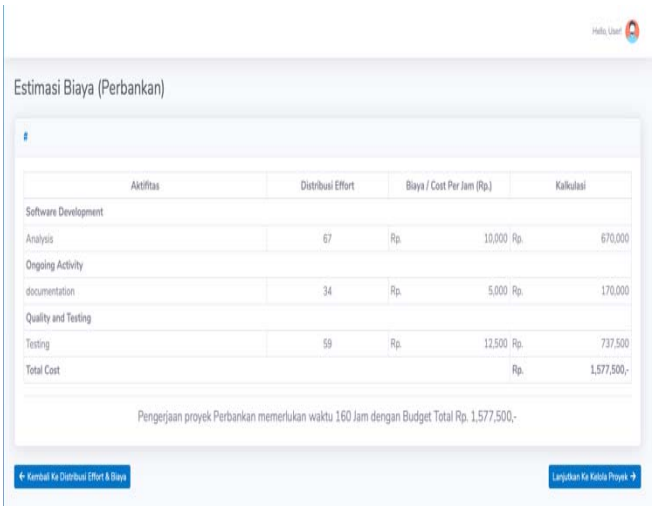

Fig. 16. Cost Calculating Result

The advantages of this application in calculating UCP costs can directly predict the number of hours load required by the project, provide an activity calculation directly according to the division of project classification, and provide an overview of costs associated with the project being developed. The distribution of effort in activities is divided into software development activity (42%), ongoing activity (21%), and quality-testing activity (37%) [15].

IV. Conclusion

The conclusion that can be presented is the calculation of estimated costs in the software manufacturing process is an important element to determine the success of a project. The success of a project can be increased by the accuracy and speed of making cost estimates. NAYA application is an application development that helps software financing calculations with the COCOMO II and UCP methods. The advantages of calculating using COCOMO II with this application are simple data that must be prepared by the user, layout of calculations with minimum wages, and a comprehensive presentation of calculation results.

Cost calculation with UCP has the advantage of implementing the Re-UCP method so that the components used are very comprehensive and complete so that it can remind UCP value and effort. NAYA applications can help software developers to calculate and arrange to finance for a software project quickly and accurately.

Acknowledgment

I would like to show my gratitude for LP2M State University of Malang as the main contributor to funding the implementation software development basis of PNBP 2020. This paper and the research behind it would not have been possible without the exceptional support of them.

Further, I would particularly like to gratitude INOBEL research which has provided the opportunity so that the application-based learning suggestions can be improved and utilized by the student.

References

[1] Amanullah, dkk "A Comparative Study for Software Cost Estimation Using COCOMO-II and Walston-Felix Models, "International Conference on Innovations in Computer Science & Software Engineering. Pakistan, pp. 1-6, 2016

[2] Anwar, dkk, "Intelligent Software Cost Estimation in Centralized and Distributed Environment, "Artifical Intelligence for Modern Systems Journal. Pakistan, vol 1, no 1, pp 19-31, 2017

[3] Borade, "Software Project Effort and Cost Estimation Techniques, "International Journal of Advanced Research in Computer Science and Software Engineering. India, vol 3, no 8, pp 730-739, 2013

[4] Eberendu, "Software Project Cost Estimation: Issues, Problems and Possible Solutions, " International Journal of Engineering Science Invention. Nigeria, vol 3, no 6, pp. 38-43, 2014

[5] Dash, dkk, "Cost Estimation for Distributed Systems Using Synthesized Use Case Point Model, " International Conference on Advances in Computing, Communication, and Control. India, vol 125, pp. 161-169, 2011

[6] Javed, dkk, "Factor Affecting Software Cost Estimation in Developing Countries, "I.J Information Technology and Computer Science. Hongkong, vol 5, pp. 54-59, 2013

[7] Jena, "Survey Report on Software Cost Estimation Using Use Case Point Method, " International Journal of Computer Science & Engineering Technology. India, vol 5, no 4, pp. 280-287, 2014

[8] Jha, dkk, "Estimating Software Development Effort Using UML Use Case Point (UCP) Method with a Modified set of Environmental Factors, " International Journal of Computer Science and Information Technologies. India, vol 5, no. 3, pp 2742-2744, ISSN: 0975-9646, 2014

[9] Joseph & Joey, "Modern System Analysis and Design 8 th Edition, " USA, Pearson Education Global Right, 2017

[10] Kirmani and Wahid, "Impact of Modification Made in Re-UCP on Software Effort Estimation, " Journal of Software Engineering and Application. India, Vol. 8, pp. 276-289, 2015

[11] Kirmani dan Wahid, "Revised Use Case Point (Re-UCP) Model For Software Effort Estimation, "International Journal of Advanced Computer Science and Application. Japan, vol 6, no.3, pp. 65-71, 2015

[12] Klakegg, dkk, "Successive Cost Estimation – Successful Budgeting of Major Projects, "Procedia Social and Behavioural Sciences. Panama, 226,pp. 176-183, 2016

[13] Nagar, "Software Efforts Estimation Using Use Case Point Approach by Increasing Technical Complexity and Experience Factors, " International Journal on Computer Science and Engineering. India, vol 3, no 10, pp. 3337-3345, 2015

[14] Rak, dkk, "Effort Estimating Model for Software Development Projects Based on Use Case Reuse, "Journal Software Evolution and Process. Croatia, DOI: 10.1002/smr.2119, 2019

[15] Saleh, "Effort and Cost Allocation in Medium to Large Software Development Projects, "International Journal of Computers. Kuwait, vol 5, no 1, pp. 74-79, 2011

[16] Shekar, dkk "Review of Various Software Cost Estimation Techniques, "International Journal of Computer Application. USA, vol 141, no 11, pp. 31-34, 2016