Effort Estimation of Software Development Using Fuzzy Use Case Points

Rakhmi Khalida, S.T

October 23, 2016

EFFORT ESTIMATION OF SOFTWARE DEVELOPMENT USING FUZZY USE CASE POINTS

Written by:

RAKHMI KHALIDA, S.T

MASTER PROGRAM IN TECHNOLOGY AND ENGINEERING

GUNADARMA UNIVERSITY

JAKARTA
2016

EFFORT ESTIMATION OF SOFTWARE DEVELOPMENT USING FUZZY USE CASE POINTS

Written by:

RAKHMI KHALIDA, S.T

NIM / NIRM: 92314103

Supervisor:

Dr. Tubagus Maulana Kusuma, SKom., MEngSc

TESIS

Submitted as a partial fulfillment

of the requirements for

Master Degree of Information System Management

Concentrating on Information System Software

MASTER PROGRAM IN TECHNOLOGY AND ENGINEERING

GUNADARMA UNIVERSITY

JAKARTA

2016

Statement of Originality and Publication

I, who undersigned below:

Name	:	Rakhmi Khalida, S.T
NIM / NIRM	:	92314103
Thesis Title	:	EFFORT ESTIMATION OF SOFTWARE DEVELOPMENT USING FUZZY USE CASE POINTS
Date of Thesis Defence	:	
Date of Yudisium	:	

state that this final project is originally my writing and can be published fully by Gunadarma University. All reference in any form were done according to the formal rules and ethics. The author, not Gunadarma University, is fully responsible for the contents of this thesis.

This was stated truthfully and with a full presence of mind.

Jakarta, September 2016

(Rakhmi Khalida, S.T)

Approval Page

Thesis Title	:	**EFFORT ESTIMATION OF SOFTWARE DEVELOPMENT USING FUZZY USE CASE POINTS**
Name	:	**Rakhmi Khalida, S.T**
NIM/NIRM	:	**92314103**
Date of Yudisium	:	13 October 2016

Approved by:

Board of Advisors

Dr. Tubagus Maulana Kusuma, SKom., MEngSc
(Chair Person)

Prof. Dr. Yuhara Sukra, MSc

Graduate Program Coordinator

Dr. Tubagus Maulana Kusuma, SKom., MEngSc
(Director of Master's Program in Technology and Engineering)

Abstract

Rakhmi Khalida, S.T.
EFFORT ESTIMATION OF SOFTWARE DEVELOPMENT USING FUZZY USE CASE POINTS.
Master Thesis, Information System Management, Information System Software, Gunadarma University, 2016.
Keyword : Estimation, Effort Rate, Software Project, Fuzzy, Use Case Points
(xv+ 80+ appendix)

Software effort estimation is one of the key aspects of successful project management. Effort estimation at the early stages of software development is a challenge. Firstly, very little is known about the project. Secondly, there is a threat that the project will not be accepted for further development, so limited resources can be spent on effort estimation. Thus, there is a trade-off between the level of estimation error and the resources assigned to the estimation activities. The common methods used for effort estimation is Use Case Points. Use case points are mainly used for application based objects, because it makes use case as the input. The UCP has weaknesses, which is high difference among the value of weight factor of UUCW. In order to overcome the abovementioned problem, Fuzzy use case points (FUCP) which is the combination between fuzzy logic and use case points, can be used. Applying fuzzy logic to this use case points in the UUCW category, fuzzy use case point has a new weight factor value of UUCW. The implementation of FUCP to calculate effort estimation in ten government based project used in this research has shown that FUCP has the closest value to the actual effort. It is also demonstrated that FUCP outperforms UCP in terms of accuracy by 6,51 % improvements.

References (1983-2016)

Curriculum Vitae

Name	:	**Rakhmi Khalida, S.T**
NIM/NIRM	:	**92314103**
Place, Date of Birth	:	Jakarta, September 4th 1992
Address	:	Jl. Melati 3 Blok F2 No.5/6 Kp Cerewet, RT/RW 08/14, Duren Jaya Bekasi Timur
Email	:	khalidarakhmi9@gmail.com

Educational Background

1997- 1998	:	Cendrawasih Kindergarten, East Bekasi
1998 - 2004	:	Bekasi Jaya Indah XIII Elementary School, East Bekasi
2004 - 2007	:	SMPN 3 Junior High School, East Bekasi
2007 - 2010	:	Bani Saleh Senior High Boarding School, East Bekasi
2010-2014	:	concern in Information Engineering, Faculty Industrial Technology in Gunadarma University, Bekasi, West Java.
2015 - 2016	:	concern in Information System Software, Magister of Information System Management in Gunadarma University, Depok, West Java.

Projects

Oct 2014 - Des 2014	:	DPA Internal Website.
March - Oct 2015	:	Portal website: Sport Science Kemenpora.

Acknowledgements

Praise and gratitude to Allah SWT who has bestowed His mercy and grace so that I can finish this thesis.

The author realizes that the preparation of this thesis can not be separated from the support of people who contributed, both morally and materially to the finish this writing. On this occasion, the authors thank profusely to all those who have helped, especially to:

1. Prof. Dr. E.S. Margianti, SE. MM, as a Rector of Gunadarma University.
2. Prof. Dr. Yuhara Sukra., M.Sc., as Coordinator of Graduate Program.
3. Dr. Tubagus Maulana Kusuma, SKom., MEngSc, as a Director of Master's Program in Technology and Engineering of Gunadarma University and as a supervisor of this thesis who spend some of her times, energy and knowledge to guides the author from the start until this thesis is done
4. Dr. Yuhilza Hanum, SSi., MEng., as Head of Master's Program in Information Management.
5. Beloved parents, H. Nurcholik and Hj. Sulasmi for their exceptional support, love, wisdom and prayers that can not be disclosed by words and encouragement to finish this writing in time..
6. Dr. rer. nat. I Made Wiryana, SKom, SSi, MAppSc and Mrs. Ernianti Hasibuan, Skom.,MSc, who always inspires, give advices and loves during education period.
7. Dr. Eng Romi Satria Wahono, BEng, MEng., as CEO Brainmatics Cipta Informatika who contributed the author for the research of this thesis.
8. Fitri Umi Khalifah, Lulu Puspita, Rifah Roszanah, Nurina Yasin, Gita Rahmi and Cintya Elsa for earning an honest living for us and for supporting and encouraging me to believe myself

9. Karina A Risaf and Asmaya Malihayati a strong and gentle soul who taught me to trust believe in hard work.
10. All FastTrack III classmate for the joy and happiness that we've shared. God bless you all.
11. All Fastrack Family for their support. It is a tremendous honor to be part of this family.

With all humility, the authors recognize that there are so many flaws in this thesis and preparation of the final project. This thesis is still far from perfection, the authors accordingly expect criticism and constructive suggestions to accomplish this thesis. Finally the authors hope this article is useful for personal self writers and readers.

Jakarta, September 2016

Author.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The software is physical abstraction that allows us to talk to the machine hardware [Langer, 2008]. Without such software, then the existing hardware can not be useful or optimal functioning. Establish or develop a software requires the discipline of standardization model of software quality. One of the most important stages in building or developing a software is planning stages. In this stage, the effort and resource estimation, human resources, materials, infrastructure, time, and budget must be taken into account [Barbosa, 2004]. The problems that found in software development projects is over estimates or under-estimates Over estimates will lead to an additional allocation of necessary resources. While under-estimates can indirectly reduce the quality of the product, due to lower costs, then the software can be made not in accordance with the standards. Both of these problems indicates that the estimate is one important factor that must be considered before running a software development project.

Estimation is a measurement that is based on the results of quantitative or numeric levels can be measured by its accuracy [Tockey, 2005]. Software estimation is an activity perform predictions or forecasts about the output of a project by reviewing the schedule, cost, even to the risk will be borne as well as an effort in the project [Evans, 2006]. Effort is the real work that we do in completing a project. The unit is days or man-man-hours [Muhardin, 2011]. Although estimates may not be able to produce a highly accurate results, but these inaccuracies can be minimized by using some method of evaluation in accordance with the project will estimate. From the description of the definition of estimation, effort estimation is an activity perform

to predict or forecast of how many workers and how long it will take to complete the project

The methods used for effort estimation, among others, Function Points [Albrecht, 1983] and Use Case Points [Karner, 1993]. Function Points method has advantages not depend on the programming language that can be applied to different types of applications. This approach is also more predictable because its parameters are calculated based on the number of input and ouput. In the method Use Case Points (UCP) is widely used in object-based applications. Estimation of software projects by using UCP easy to use on the software environment that have complex factors.

UCP method is a method that is able to provide an estimate of effort (person per hour) required to create a project [Karner, 1993]. UCP estimation does not relate to the estimation of the cost factor. Methods of Use Case Points (UCP) was invaluable in the context of early size measurement and estimation of effort, because it makes use case as input but UCP has weaknesses which is high difference among the value of weight factor of UUCW. As an alternative solution of Fuzzy logic used UCP weakness. Fuzzy logic will modify the value of the multiplier on the classification of the use case. Fuzzy logic was introduced by Lotfi Zadeh, Prof. of University of California at Berkeley in 1965. This method is often used to overcome an uncertain on issues that have a lot of answers. Fuzzy inference system provides a way to describe a definitive conclusions from information that is vague, ambiguous and imprecise. Research on the fuzzy inference system Mamdani model as a kind of fuzzy logic can be used to get the default value of multiplier use case classification and expected from the integration of this method gets an effort more accurate or close to the actual effort can be used as reference for software developers to do the effort estimation in software development projects.

1.2 Problem Formulation

Based on the background, The problem of this research are::

1. How to get actual effort value as a reference in software development projects?

2. How to get the value of effort estimation used UCP and FUCP method in software development projects?

3. How it compares UCP and FUCP value which has been obtained on the value of the actual effort?

4. How the UCP and FUCP correlation to actual effort?

1.3 Scope of The Research

Boundary problem in this research are :

1. This research uses ten data goverment software development based on projects, namely :

   (a) Evaluation Kemenpora Website (Website Evaluasi Kemenpora)

   (b) Financal System of Dikti (Sistem Keuangan Dikti)

   (c) SIMAYA System (Sistem SIMAYA)

   (d) Inventory System of Pekalongan City (SIMSEDIA)

   (e) Geographical Information System (GIS) Website of Kemenpora

   (f) Sportscience Website of Kemenpora (Website Sportscience Kemenpora)

   (g) Assesment Library Website of Kemdikbud (Website Perpustakaan Penilaian Kemdikbud)

   (h) Internasional Study Website of Kemdikbud (MINITES Kemdikbud)

   (i) Biodiversity Mobile Apllication (Aplikasi Biodiversity)

   (j) BSN E-learning Website (Website E-learning BSN)

2. The data in this research were obtained by interview, questionnaire and refer to the documents the specifications of software projects that have been completed. Data obtained in the form of a use case diagram is the latest of a software

3. All projects have the actual effort in hours, the value of the effort is the amount of time needed by developers in completing the project, starting from the stage of planning, analysis, design and implementation.

4. Research aimed to get the effort estimation rate in hours, not until the calculation of the estimataion cost.

1.4 Research Objective

The objective of this research are :

1. Get the actual effort values in units of hour in software development projects.
2. Get the value of effort estimation used use case points (UCP) and fuzzy use case points (FUCP) method in software development projects.
3. To know result of the comparison between actual effort with UCP and FUCP.
4. To know the correlation between actual effort with UCP and FUCP

1.5 Research Contribution

This research has contributions to making effort estimation more accurate without increasing the time and money spent on effort estimation and simplifying effort estimation methods without compromising their accuracy.

Chapter 2

Literature Study

2.1 Effort

As the name suggests, 'effort' is the number of work units that is vital to complete an activity. In simpler terms, it is the number of hours we put in, focused on a particular task, to get a certain job done. Effort has to do with how hard you're trying. If something is easy, it doesn't take much effort. If it's hard, it takes a lot of effort.

Effort is about trying to get something done, even if it doesn't always work out. If you want to determine any of the other two, you will need to determine the effort in a project first. Effort is most often expressed in Staff - hours, days or weeks [Monnappa, 2015].

Stakeholders often want to know how much a project will cost. This chiefly depends on the measure of time members of the project spend on the project. In sports, coaches don't like losing, but they really hate it when the players aren't giving much effort. There are also large scale, collective efforts, like the effort to cure cancer.

A simple example to explain this concept is, say you begin to paint your house. You work 6 hours a day for 9 days. Your effort would then be the amount of time you take in a day multiplied by the number of days you work, which would be 54 hours. The effort you put in is 54 hours

2.1.1 Actual Effort

Actual effort is the amount of time needed by programmers in completing the project, starting from the stage of planning, analysis, design to implementation. Do always take the planned effort for a task, put a schedule, assign the

resources, identify the start date and the end date. Most of the time, project team always meet the target end date. Project manager encourage the team members to give the real actual effort. Many opinion says rate people 'good' if they have completed the task before the deadline and neither rate them 'bad' if they have taken more the actual effort. In either of the case, there is a learning that must be carried to your every project.

The only thing that project manager emphasize to the team members is that they have given me the actual and now then own it. Let's give the best faith effort to stick to those end dates. But not to forget in meeting those end dates goals, project team might have spent some extra time from they personal time. There is always a continuous learning and with this learning, project team must ensure that must learn something from this project and take this learning in our next project and to ensure that the mistakes did are not repeated [Kumara, 2011].

2.1.2 Effort Estimation

Software project management process starts with project planning activities. One of the criteria for the assessment of the project so that it can be said to be successful is that the finished product in a timely manner and in accordance with the cost of the plan. In the planning stage, important activity is the approximate time and is also closely related to the production costs of project completion.

Software as a product of the complex and intangible (cannot be assessed visually), require special treatment so that no failure occurred in creating or developing the software. One of the main causes of the failure of the creation or development of software i.e. poor project planning. One way to fix that is by doing project planning estimates.

The first activity of the project planning is estimation. Estimation is the basis for all other project planning activities and project planning provides a road map for success in software engineering, then without the estimation of the project can not run properly.

Important side estimation of project planning is the emergence of the right schedule and budget, though not entirely an estimate will end up with the right. But, without a software project estimation can be regarded as a blind project. [Rizky, 2011]. Estimation cannot be measured mechanically and is based on the results of quantitative or numeric levels can be measured by its accuracy [Tockey, 2006]. The definition of a software estimation activi-

ties do predictions or forecasts about the output of a project by reviewing the schedule, effort, even to the cost of risk will be borne in such project [Evans, 2006].

In the estimation, ideally an estimation beginning with measuring project size. Other metrics such as effort can be seen after the project size is known. Project size is the size of a project. With the project size, we can compare two different applications.

There are many metric to measure the size of the project [Muhardin,2011]. The most commonly metric used is the number of lines of code, namely Source Lines of Code (SLOC) or Non Commenting Source Code (NCSS). Even though SLOC is very accurate in determining the size of the project, but the number of lines of code is difficult to estimate at the beginning of the project. Therefore some creative people and then develop a new method called Function Point, Use Case Points and others. After getting the estimated size of the project, will be used to effort estimation , duration, and cost.

Effort is the real work that we do in completing a project. Unit is mandays or manhour. For example, an application is being estimated to need effort 10 mandays. This means that this application will be completed when it is done one person for 10 continuous days or 5 days if there are two workers. Effort did not know holiday or time off. Schedule is a period of completion of the project. This is usually expressed in units of working days or calendar days. When the duration of the project is expressed 10 calendar days, then if starts December 1, will be completed on 10.

Effort estimation is predict the real work in completing the project by reviewing the schedule, project size, even the project risk will be borne

So, to get the duration, must have the following assumptions:

1. How many people are employed
2. How many days holiday
3. How the allocation of time for non-work such as meeting, presentations, etc.

That assumption, combined with estimates of effort, will generate the estimated duration. Assumptions should refer to historical data because the data is historically very influential in making estimation. Historical data into the guidelines and will be used for conversion to get size, effort and schedule. The following are the needs of the data needed to conduct the estimation :

1. Conversion Table size to mandays.
2. Table distribution of effort every phase
3. Table distribution of effort every role

Estimation Flow :

1. Estimation of Size of Project
2. From the size, use the table 1 to get the effort.
3. From the effort, use the table 2 to get the duration.
4. From the effort, use the table 3 to get the effort per personnel.

2.2 Use Case Points

Use Case Points is a method of effort estimation in software development [Karner, 1993]. This method is aimed at measuring software functional size as early as possible in the development cycle. This method provides an estimate of the actual estimation approach resulting from the experience of the creation or development of software. The UCP method was created to solve for estimating the software size of systems that were object oriented. It is based on similar principles as the Function Point (FP) estimation method, but was designed for the specific needs of object oriented systems and system requirements based on use cases. Use case points (UCP) is used when the Unified Modeling Language (UML) and Rational Unified Process (RUP) methodologies are being used for the software design and development. The concept of UCP is based on the requirements for the system being written using use cases diagram, which is part of the UML set of modeling techniques. Tools used to simplify creating use case diagrams i.e. StarUML 2.7. StarUML will be discussed in points 2.3.

The design of UCP takes three aspects of a software project into account:

1. Use cases diagram
2. Technical qualities
3. Development resources.

In order to obtain UCP for the system one has to start with the assessment of the complexity of actors and use cases and then adjust it with two kinds of factors characterizing the development environment and the technical complexity of the system under development [M Ochodek, 2010]. The UCP is calculated by multiplying the unadjusted use case points (UUCP) by the technical qualities (TCF) and environmental factors (ECF) from development resources as follows :

$$UCP = UUCP * TCF * ECF \tag{2.1}$$

Description :

1. UUCP = Unadjusted Use Case Points
2. TCF = Technical Complexity Factor
3. ECF = Environmental Complexity Factor

If the project complexity is high and the team efficiency is low, there will be a high risk that this project will fail. Typically, by taking the technical and the environmental factors into consideration, the value of the adjusted use case points (UCP) will be 30% more or less than the unadjusted use case points (UUCP). This is evidenced by several studies that have been done before, and resulted in the following statement :

1. UCP has 6% deviation. The effort estimation equation of UCP produce man-367 days, whereas the results of actual effort man-390 days so there is deviation of 6 percent.
2. The UCP has a deviation of 19%, while estimates of the experts have an average deviation of 20% [Anda, 2002].
3. UCP have deviation amounted to 9% [Carrol, 2005].

   On the research of Carrol, mentioned "after applying a considerable process on hundreds of software projects (average 60 man-hours), generate metrics that prove the accuracy of that estimate is less than 9% of the deviation of actual costs for estimated at 95% of the projects.

The use case points model (UCP) is detailed as follows section 2.2.1 until section 2.2.4

2.2.1 Use Case

The main aim of use cases is to present interaction between end-user (called actor) and the described system in terms of user-valued transactions – using natural language [M Ochodex, 2010]. To perform project estimation using use case points, make first a use case diagram. Use case diagram designed for the specific needs of object oriented systems and system requirements based on use cases.

1. Unadjusted Actor Weight (UAW)

   On use case diagram make the first classify the actor according to Table 2.1. UAW value obtained from the classification of existing actors in the use case diagram. Results of multiplying the number of actors according to their classification unadjusted actor called Actor Weight (UAW). The actor in use case diagram can be seen in figure 2.1.

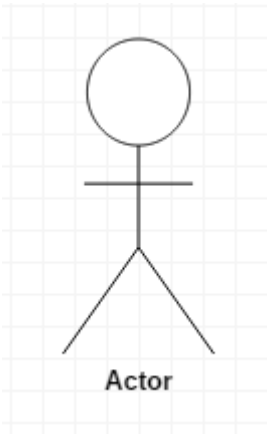


Figure 2.1: Actor in Use Case Diagram [Lee, 2016]

   In the UCP, actors are classified as simple, average or complex. A weight is assigned to each category as follows :

   (a) Simple actor: This is described as another system through an API. Its weight is 1.

      i. Average actor: This is described as another system interacting through a text-based user interface or a protocol. Its weight is 2.

      ii. Complex actor: This is described as a human interacting with system through a graphical user interface (GUI). Its weight is 3.

The UAW can be seen in Table 2.1.

Table 2.1: Unadjusted Actor Weight (UAW)

Actor Type	Description	Weighting Factor
Simple	Interaction system with well defined API	1
Average	Interaction system using a protocol based interface	2
Complex	Human	3

The UAW is calculated as :

$$UAW = (\sum SA * 1) + (\sum AA * 2) + (\sum CA * 3) \qquad (2.2)$$

Where SA, AA, and CA correspond to Simple Actors, Average Actors and Complex Actors, respectively.

(b) Unadjusted Use Case Weight (UUCW)

In addition to actors, each use case is classified based on the number of transactions. The results of the calculation of the number of use case and its classification is called Unadjusted Use Case Weight (UUCW). The use case in use case diagram can be seen in figure 2.2

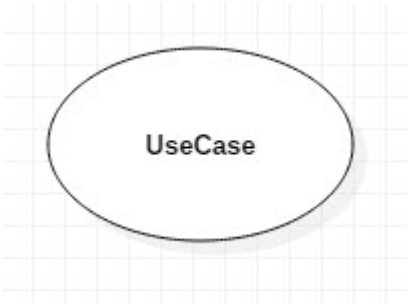


Figure 2.2: Use Case in Use Case Diagram [Lee, 2016]

Use cases are classified based on the number of transactions in the success and alternative scenarios. A weight is assigned to each category as follows :

i. Simple Use Case: A use case is classified as Simple if the number of transactions is ≤ 3. Its weight is 5.

ii. Average Use Case: A use case is classified as Average. If the number of the transactions is between 4 and 7. Its weight is 10.

iii. Complex Use Case: A use case is classified as Complex if the number of transactions is more than 7. Its weight is 15.

The UUCW can be seen in Table 2.2.

Table 2.2: Unadjusted Use Case Weight (UUCW)

Use Case	Description (Transaction)	Weighting Factor
Simple	1 - 3	5
Average	4 - 7	10
Complex	> 7	15

The UUCW is calculated as :

$$UUCW = (\sum SU * 5) + (\sum AU * 10) + (\sum CU * 15) \quad (2.3)$$

Where SU, AU, and CU correspond to Simple Use Case, Average Use Case and Complex Use Case.

(c) Unadjusted Use Case Points (UUCP)

The value of the UAW and UUCW combined to become the value of Unadjusted Use Case Points (UUCP). This is described as :

$$UUCP = UAW + UUCW \quad (2.4)$$

Description :

i. UAW = Unadjusted Actor Weight

ii. UUCW = Unadjusted Use Case Weight

UCP is achieved by multiplying UUCP by the technical complexity factors (TCF) from technical qualities and the environmental complexity factors (ECF) from development resources.

2.2.2 Technical Complexity Factor (TCF)

TCF is a technical factors affecting software development projects. Technical qualities are represented by a Technical Complexity Factor (TCF), which consists of 13 technical qualities (Table 2.3), each with a specific weight, combined into a single factor. To calculate the TCF, an expert must assess the relevance to the project of each technical quality, evaluated on a scale from 0 to 5 (where 0 is 'unrelated' and 5 is 'indispensable'). The weights

are balanced in such a way that a relevance factor of 3 for each quality will produce a TCF equal to 1.

The TCF is thus the sum of all the relevance factors (one for each technical quality) multiplied by their corresponding weights plus two constants, C2 (0.1) and C1 (0.6): Table 2.3 lists the quality factors and their corresponding weights. Karner bases his design for these weights on the constants and weights of the FPA Value Adjustment Factors. These factors contribute to the complexity of the project. The technical factors can be seen in Table 2.3.

The technical complexity factor (TCF) is calculated as follows :

$$TCF = C_1 + C2 \sum_{i=1}^{13} TF \tag{2.5}$$

Where C_1 = 0.6, C_2 = 0.01 and TF is a factor that takes values between "0" and "5". The value "0" means the factor is irrelevant while the value "5" is essential. The value "3" means that the factor is not very essential, neither irrelevant (average). For instance, if all the factors have a value of "3", the TF will be 1.

Table 2.3: Technical Complexity Factors

No	Description	Weight
T_1	Distributed System	2,0
T_2	Response time or throughput performance objectives	1,0
T_3	End-user online efficiency	1,0
T_4	Complex internal processing	1,0
T_5	Reusability of code	1,0
T_6	Easy to install	0,5
T_7	Ease of use	0,5
T_8	Portability	2,0
T_9	Ease of change	1,0
T_{10}	Concurrency	1,0
T_{11}	Special security objectives include	1,0
T_{12}	Direct access for thrid parties	1,0
T_{13}	Special user training required	1,0

2.2.3 Environmental Complexity Factor (ECF)

ECF is a environment factors affecting software development projects. Development resources are represented by Environment Complexity Factors (EcF), also referred to as experience factors. The UCP model identifies eight such factors (Table 2.4) contributing to the effectiveness of the development team. To calculate the EF, an expert must assess the importance of each factor and

classify it on a scale from 0 to 5 (0 meaning 'very weak'; 5 meaning 'very strong'). The selection of the weights is balanced such that a value of 3 for each factor will produce an EF of 1. The EF is the sum of all the factors multiplied by their weights and two constants, C2 (-0.03) and C1 (1.4). These factors contribute to the team efficiency and productivity. The environmental factors are presented in Table 2.4. The environmental complexity factor (ECF) is calculated as follows :

$$ECF = C1 + C2 \sum_{i=1}^{8} EF \tag{2.6}$$

Where C_1 = 1.4, C_2 = -0.03 and EF is a factor which i equivalent to the Fi of the technical factor (i.e. between 0 and 5). The environmental factors can be seen in Table 2.4

Table 2.4: Environmental Factors

No	Description	Weight
E_1	Familiarity with system development process being used	1,5
E_2	Application experience	0,5
E_3	Object-oriented experience	1,0
E_4	Lead analyst capability	0,5
E_5	Motivation	1,0
E_6	Requirements stability	2,0
E_7	Part time staff	-1,0
E_8	Difficulty of programming language	-1,0

2.2.4 UCP Effort

To obtain effort estimation in man-hours one has to multiply UCP by the Person Man Hours (PHM). This is the final stage of the use case points model. The default value for PHM proposed by Karner is 20 hours per UCP. The value of the effort can be known from the UCP multiplied by the value of the PHM (Person Hour Multiplier). The value of effort is calculated as follows :

$$Effort = UCP * PHM \tag{2.7}$$

Description :

1. UCP = Use Case Points
2. PHM = Person Man Hours

Schneider and Winters [G Schneider, 1998] proposed a method for determining the initial value of PHM. Based on their experience, they suggested to count the number of environmental factors by the following rules :

1. F1 = Total weighted value E1 to E6 which has a value <3
2. F2 = Total weighted value E7 to E8 which has a value> 3
3. If F1 + F2 <= 2 then PHM = 20
4. If F1 + F2 = 3 or 4 then PHM = 28
5. If F1 + F2> 4 then the project should be canceled

2.3 StarUML

StarUML is a software modeling platform that supports Unified Modeling Language (UML). It is based on UML version 1.4 and provides eleven different types of diagram, and it accepts UML 2.0 notation [Lee, 2016]. It actively supports the Model Driven Architecture (MDA) approach by supporting the UML profile concept. StarUML excels in customizability to the user's environment and has a high extensibility in its functionality. Using StarUML, one of the top leading software modeling tools, will guarantee to maximize the productivity and quality of your software projects.

1. Uml tool that adapts to the user

   StarUML provides maximum customization to the user's environment by offering customizing variables that can be applied in the user's software development methodology, project platform, and language.

2. True MDA support

   Software architecture is a critical process that can reach 10 years or more into the future. The intention of the Object Management Group (OMG) is to use Model Driven Architecture (MDA) technology to create platform independent models and allow automatic acquisition of platform dependent models or codes from platform independent models. StarUML truly complies with UML 1.4 standards, UML 2.0 notation and provides the UML Profile concept, allowing creation of platform independent models. Users can easily obtain their end products through simple template document.

3. Excellent extensibility and flexibility

   StarUML provides excellent extensibility and flexibility. It provides Add-In frameworks for extending the functionality of the tool. It is designed to allow access to all functions of the model/meta-model and tool through COM Automation, and it provides extension of menu and option items. Also, users can create their own approaches and frameworks according to their methodologies. The tool can also be integrated with any external tools.

4. Key feature

Table 2.5: Key Feature Star UML [Lee, 2016]

Feature	Description
Accurate UML standard model	StarUML strictly adheres to the UML standard specification specified by the OMG for software modeling. Considering the fact that the results of design information can reach 10 years or more into the future, dependence on vendor-specific irregular UML syntax and semantics can be quite risky. StarUML maximizes itself to order UML 1.4 standard and meaning, and it accepts UML 2.0 notation on the basis of robust meta model.
Open software model format	Unlike many existing products that manage their own legacy format models inefficiently, StarUML™ manages all files in the standard XML format. Codes written in easy-to-read structures and their formats can be changed conveniently by using the XML parser. Given the fact that XML is a world standard, this is certainly a great advantage, ensuring that the software models remain useful for more than a decade.
True MDA support	StarUML truly supports UML Profile. This maximizes extensibility of UML, making modeling of applications possible even in areas like finance, defense, ebusiness, insurance, and aeronautics. Truly Platform Independent Models (PIM) can be created, and Platform Specific Model (PSM) and executable codes can be automatically generated in any way.

Applicability of methodologies and platforms	StarUML manipulates the approach concept, creating environments that adapt to any methodologies/processes. Not only the application framework models for platforms like .NET and J2EE, but also basic structures of software models (e.g. 4+1 view-model, etc.) can be defined easily
Excellent extensibility	All functions of the StarUML™ tools are automated according to Microsoft COM. Any language which supports COM (Visual Basic Script, Java Script, VB, Delphi, C++, C#, VB.NET, Python, etc.) can be used to control StarUML™ or develop integrated Add-In elements.

5. System Requirement

   The following are the minimum system requirements for running StarUML :

   (a) Intel® Pentium® 233MHz or higher

   (b) Windows® 2000, Windows XP, or higher

   (c) Microsoft® Internet Explorer 5.0 or higher

   (d) 128 MB RAM (256MB recommended)

   (e) 110 MB hard disc space (150MB space recommended)

   (f) CD-ROM drive

   (g) SVGA or higher resolution monitor (1024x768 recommended)

   (h) Mouse or other pointing device

2.4 Matlab Software

Matlab is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation [Mathworks, 2016]. Typical uses include :

1. Math and computation
2. Algorithm development
3. Modeling, simulation, and prototyping
4. Data analysis, exploration, and visualization Scientific and engineering graphics
5. Application development, including Graphical User Interface building

Matlabis an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran. The name Matlab stands for matrix laboratory. Matlab was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation. Matlab has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, Matlab is the tool of choice for high-productivity research, development, and analysis. Matlab features a family of application-specific solutions called toolboxes. Very important to most users of Matlab, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of Matlab functions (M-files) that extend the Matlab environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others. The Matlab system consists of five main parts :

1. The Matlab language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

2. The Matlab working environment

   This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

3. Handle Graphics

   This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

4. The Matlab mathematical function library

   This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

5. The Matlab Application Program Interface (API)

   This is a library that allows you to write C and Fortran programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

2.4.1 Fuzzy Logic Toolbox

The Fuzzy Logic Toolbox is a collection of functions built on the MATLAB numeric computing environment. It provides tools for you to create and edit fuzzy inference systems within the framework of MATLAB, or if you prefer you can integrate your fuzzy systems into simulations with Simulink, or you

can even build stand-alone C programs that call on fuzzy systems you build with MATLAB. This toolbox relies heavily on graphical user interface (GUI) tools to help you accomplish your work, although you can work entirely from the command line if you prefer. The toolbox provides three categories of tools :

1. Command line functions
2. Graphical, interactive tools
3. Simulink blocks and examples

The first category of tools is made up of functions that you can call from the command line or from your own applications. Many of these functions are MATLAB M-files, series of MATLAB statements that implement specialized fuzzy logic algorithms. You can view the MATLAB code for these functions using the statement type function_name You can change the way any toolbox function works by copying and renaming the M-file, then modifying your copy. You can also extend the toolbox by adding your own M-files. Secondly, the toolbox provides a number of interactive tools that let you access many of the functions through a GUI. Together, the GUI- based tools provide an environment for fuzzy inference system design, analysis, and implementation. The third category of tools is a set of blocks for use with the Simulink simulation software. These are specifically designed for high speed fuzzy logic inference in the Simulink environment.

2.5 Fuzzy Inference System

The concept of fuzzy logic was introduced by Lotfi Zadeh, Prof. of University of California at Berkeley in 1965. Basic fuzzy logic is fuzzy set theory. Fuzzy inference system is computing systems that work on the basis of the principle of fuzzy reasoning. Fuzzy inference system used to map the input value becomes the value of the output using fuzzy logic. Fuzzy inference system (FIS) is conclusion of a set of fuzzy rules, so in the FIS a minimum there must be two rules Fuzzy. FIS a type of fuzzy logic is commonly applied to problems that contain elements of uncertainty, inaccuracies or imprecise, noisy, and so on. Fuzzy logic connecting precision machine with language human language that emphasizes on the meaning or meanings or significance. Fuzzy inference system developed on the basis of human language (natural language).

Fuzzy inference system can perform reasoning with similar principles such as human reasoning on instinct [Kusumadewi, 2010]. There are several types of Fuzzy Inference System (FIS) are recognized Mamdani, Sugeno and Tsukamoto. FIS is the easiest to understand. because it suits human instincts The FIS work based on linguistic norms and have fuzzy algorithm that provides a uniform to enter the mathematical analysis. The fuzzy inference system can be seen in figure 2.3

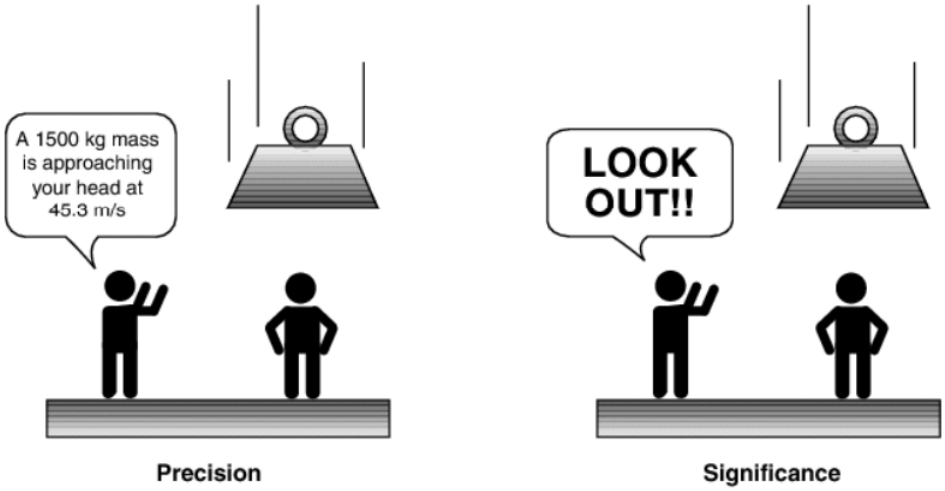


Figure 2.3: Fuzzy Inference System [Kusumadewi, 2010]

Step of FIS can be seen in figure 2.4. The Fuzzy Inference System (FIS) is detailed as follows :

1. Fuzzyfication

   Fuzzyfication is the process of mapping the value of crisp (numeric) into the fuzzy set and determine the degree of membership in fuzzy sets. This must be done because the data is processed based on fuzzy set theory so that the data is not in the form of fuzzy must be converted into fuzzy.

2. Rule Evaluation

   Fuzzy rules are a collection of linguistic statements that describe how the FIS should make a decision regarding classifying an input or controlling an output. Fuzzy rules are expressed in the form of "IF THEN" which is the core of the fuzzy relation. The General form of the rule used for the function of the implication is if x is A then y is B, with x

and y as a scalar, while the A and B is the set of fuzzy. The proposition that is located after the if is called antiseden, and proposition which is located after the then called the consequent. Basic fuzzy logic operator used and, or, not. At the time linked by operator and or not then the degrees of truth calculated with fuzzy matching operations.

3. Defuzzyfication

   Defuzzyfication is process of conversion of the fuzzy data into numerical data that can be sent to the control equipment because the system is set up with the real quantities, not quantity Fuzzy. There are several methods of defuzzification in modeling fuzzy systems, namely :

   (a) Centroid Method

   In this method, a solution is obtained by taking the decisive value is the center point (Z) fuzzy area.

   (b) Bisektor Method

   In this method, the value of the solution obtained by taking a resolute value on fuzzy domain membership value is half of the total value of membership in fuzzy areas.

   (c) Method of Mean of Maximum (MOM).

   Method of Mean of Maximum (MOM). In this method the unequivocal value solution is obtained by taking the average value of a domain that has a maximum membership values.

   (d) Largest method of Maximum (LOM).

   Largest method of Maximum (LOM). In this method, the value of the solution obtained by taking a resolute the greatest value from the domain that has a maximum membership values.

   (e) Methods Smallestof Maximum (SOM).

   Methods Smallestof Maximum (SOM). In this method, a solution is obtained by way of resolute value take the smallest value from a domain that has a maximum membership values.

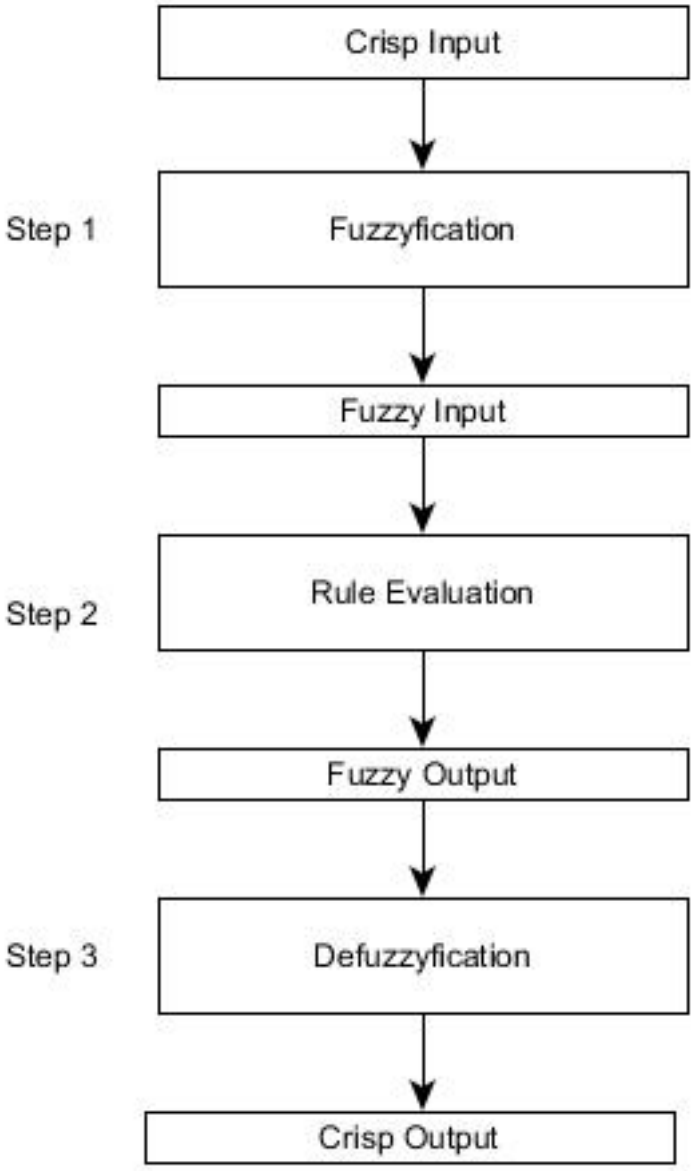


Figure 2.4: Step of FIS [Kusumadewi, 2004]

2.5.1 Fuzzy Use Case Points

The concept fuzzy use case points methods created from the expansion method use case points (UCP) with fuzzy logic Mamdani type. This method is also used to calculate estimates of the project effort. This method is used to obtain the estimated effort of more varied value and investigate the causal relationship between the UCP with FUCP using the evaluation effort estimation model.

Essentially there is no difference in the step of the process calculating the

estimate project effort using fuzzy use case points with using use case points, that the difference is the weight factor when looking for value classification UUCW transaction use case.

Using the same data that is value UAW, TCF, and ECF use the value obtained by using use case points. Do the calculation of estimated effort using fuzzy use case points. Only one category given the influence of fuzzy logic, namely weighting factor UUCW while other categories of data processed using the appropriate method of use case points. Fuzzy logic is applied to the method of use case points will affect the value of the use case points. Fuzzy logic is implemented in use case points affect the result accuracy rate. Based on the weaknesses of the use case points (UCP) which is high difference among the value of weight factor of UUCW. This use case points fuzzy modify the weighting factor UUCW. After the fuzzy logic process step are completed, the new classification weighting factor on UUCW use case has been obtained through the fuzzy logic process. The value of the use case points (UCP) which has been combined with fuzzy called fuzzy use case points (FUCP).

Step calculating effort estimation with fuzzy use case points method can be seen in Figure 2.5.

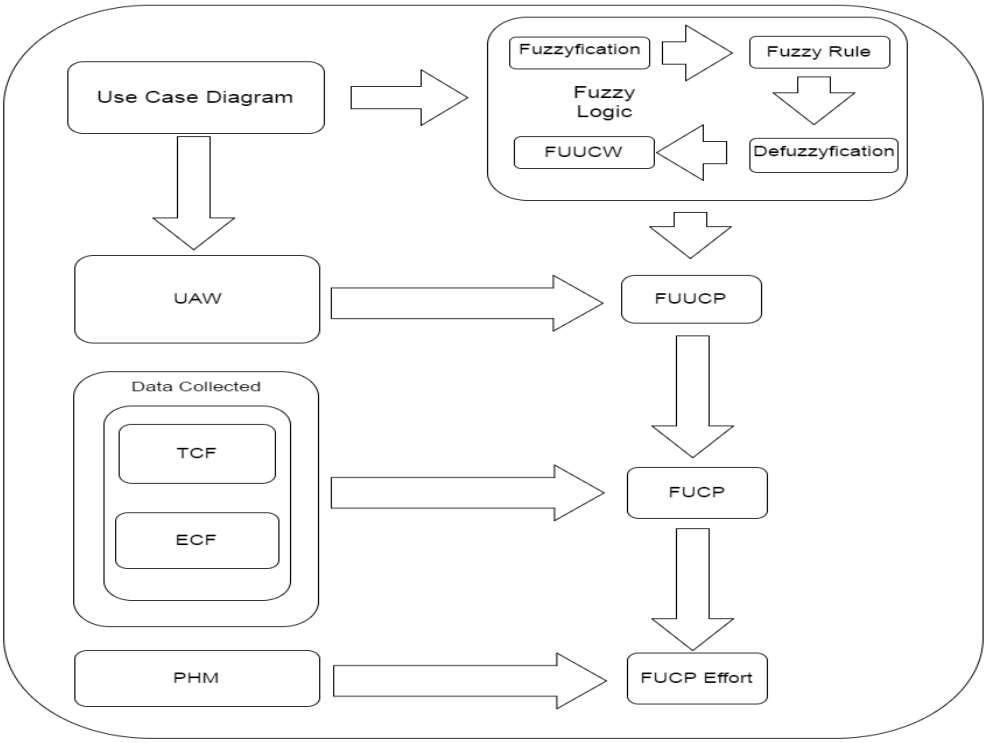


Figure 2.5: Step Calculating Effort Estimation with Fuzzy Use Case Points

Fuzzy use case points will be described in more detail on the process as

follows :

1. UAW (Unadjusted Actor Weight)

   Actors in use case diagram are classified as simple, average or complex. Explanation unadjusted actors weight (UAW) can be seen in section 2.2.1 and calculate UAW can be seen in the formula number 2.2

2. Fuzzy Logic

   Fuzzy logic is applied to the method of use case points will affect the value of the use case points. Fuzzy logic is implemented in use case points affect the result accuracy rate. Based on the weaknesses of the use case points (UCP) which is high difference among the value of weight factor of UUCW. This use case points fuzzy modify the weighting factor UUCW. Explanation fuzzy logic can be seen in section 2.5.

3. FUUCP (Fuzzy Unadjusted Use Case Points)

   The value of the UAW and UUCW combined to become the value of Fuzzy Unadjusted Use Case Points (FUUCP).

   $$UUCP = UAW + FUUCW \tag{2.8}$$

   Description :

   - UAW = Unadjusted Actor Weight
   - FUUCW = Fuzzy Unadjusted Use Case Weight

4. TCF (Technical Complexity Factor)

   TCF is a technical factors affecting software development projects. Explanation Technical Complexity Factor (TCF) can be seen in section 2.2.2.

5. ECF (Environment Complexity Factor)

   ECF is a environment factors affecting software development projects. Explanation Environment Complexity Factor (ECF) can be seen in section 2.2.3.

6. FUCP (Fuzzy Use Case Points)

   FUCP is achieved by multiplying UUCP by the technical complexity factors (TCF) from technical qualities and the environmental complexity factors (ECF).

$$FUCP = FUUCP * TCF * ECF \tag{2.9}$$

Description :

(a) FUUCP = Fuzzy Unadjusted Use Case Points

(b) TCF = Technical Complexity Factor

(c) ECF = Environmental Complexity Factor

7. FUCP Effort

This is the final stage of the use case points model. Karner proposed 20 person-hours for each FUCP. The value of the effort can be known from the FUCP multiplied by the value of the PHM (Person Hour Multiplier). The value of effort is calculated as follows :

$$Effort = FUCP * PHM \tag{2.10}$$

Description :

(a) FUCP = Fuzzy Use Case Points

(b) PHM = Person Man Hours

The value of the PHM is obtained by the following rules :

(a) F1 = Total weighted value E1 to E6 which has a value <3

(b) F2 = Total weighted value E7 to E8 which has a value> 3

(c) If F1 + F2 <= 2 then PHM = 20

(d) If F1 + F2 = 3 or 4 then PHM = 28

(e) If F1 + F2> 4 then the project should be canceled

2.6 The Evaluation Effort Estimation Method

Another problem in determining the effort estimation methods is the difference between estimates and actual. This difference is usually called "Error". If the error results from a variety of different methods studied it is often found that the method which produces the smallest error is the best. Several model exist to evaluate effort estimation methods. In this research, the evaluation model used Root Mean Square Error (RMSE), R Square and Adjusted R Square.

2.6.1 Root Mean Square Error (RMSE)

RMSE are regularly employed in model evaluation studies. The RMSE is more appropriate to represent model performance or effort. RMSE is an alternative method to evaluate the estimation techniques used to measure the accuracy of the estimation model. RMSE is the average value of the sum of squared errors, can also express the magnitude of errors generated by a estimation model. Low RMSE value indicates that the variation value produced by a variation in the estimation model closer observation According Makridakis, at. al, one measure of error in estimation is the middle value or the square root Root Mean Square Error (RMSE).

The root mean square error (RMSE) has been used as a standard statistical metric to measure model performance in meteorology, air quality, and climate research studies. In the field of geosciences, many present the RMSE as a standard metric for model errors e.g., McKeen et al., 2005, Savage et al., 2013, and Chai et al., 2013. One distinct advantage of RMSE is that RMSEs avoid the use of absolute value, which is highly undesirable in many mathematical calculations [Chai, Draxler, 2014].

RMSE is defined by the formula :

$$RMSE = \sqrt{\frac{\sum(Actual\,effort - Effort\,estimastion)^2}{n}} \tag{2.11}$$

Description :

1. Actual effort = Actual effort is the amount of time needed by programmers in completing the project, starting from the stage of planning, analysis, design to implementation.

2. Effort estimation = method of calculation of estimated effort which refers to the use case points and fuzzy use case points..

3. N = the number of projects analyzed

2.6.2 R Square and Adjusted R Square

In statistics, the coefficient of determination, denoted R^2 or r^2 and pronounced "R squared", is a number that indicates the proportion of the variance in the dependent variable that is predictable from the independent variable. Determination of coefficient of linear regression on often refers to how

large the ability all independent variable in explaining the variance of the dependent variables. Simply put the coefficient of determination is calculated by squaring the correlation Coefficient (R). For example, if the value of R is the coefficient 0.80 then determination (R Square) is amounting to 0.80 x 0.80 = 0.64. Means the ability of a independent variable in explaining the variance of the dependent variable is 64,0%. Meaning there are 36% (100%-64%) dependent variables the variance explained by other factors. Based on these interpretations, then it appears that the value R Square is between 0 to 1. R squared is defined by the formula :

$$R^2 = 1 - \frac{SSR}{SST} = 1 - \frac{\sum(\hat{y}_i - \bar{y})}{\sum(y_i - \bar{y})} \tag{2.12}$$

Description :

1. SSR = SS Error
2. SST = SS Total
3. $\hat{y}_i$ =Effort Estimation
4. y_i =Actual Effort
5. $\bar{y}$ =Averaging

The use of R Square (R squared) often cause problems, namely that its value will always increase with the addition of independent variables in a model. This will cause the deviation, because if you want the model with a high R, a study can be recklessly adding independent variable and the value of R will increase, regardless of whether the additional independent variables associated with the dependent variable or not. Therefore, many researchers are advised to use the Adjusted R Square. Interpretation is the same as R Square, but the value of Adjusted R Square can rise or fall with the addition of a new variable, depending on the correlation between the additional independent variables with the dependent variable. Adjusted R Square value may be negative, so if the value is negative, then the value is considered 0, or independent variable totally unable to explain the variance of the dependent variable. Adjusted R square is defined by the formula :

$$Adjusted\, R^2 = 1 - \frac{(1 - R^2) - (N - 1)}{\grave{N} - M - 1} \tag{2.13}$$

Description :

1. R^2= R Square
2. N = Number of data points
3. M = Independent Variables.

2.6.3 Spearman's Rank-Order Correlation

The Spearman's rank-order correlation is the nonparametric version of the Pearson product-moment correlation. Spearman's correlation coefficient, measures the strength of association between two ranked variables.

You need two variables that are either ordinal, interval or ratio. Although you would normally hope to use a Pearson product-moment correlation on interval or ratio data, the Spearman correlation can be used when the assumptions of the Pearson correlation are markedly violated. A second assumption is that there is a monotonic relationship between your variables.

A monotonic relationship is a relationship that does one of the following :

1. As the value of one variable increases, so does the value of the other variable.
2. As the value of one variable increases, the other variable value decreases.

A monotonic relationship is an important underlying assumption of the Spearman rank-order correlation. It is also important to recognize the assumption of a monotonic relationship is less restrictive than a linear relationship (an assumption that has to be met by the Pearson product-moment correlation). The middle image above illustrates this point well: A non-linear relationship exists, but the relationship is monotonic and is suitable for analysis by Spearman's correlation, but not by Pearson's correlation.

There are two methods to calculate Spearman's rank-order correlation depending on whether :

1. Your data does not have tied ranks. The formula for when there are no tied ranks is :

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2.14}$$

2. Your data has tied ranks. The formula to use when there are tied ranks is :

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \tag{2.15}$$

Description :

(a) d_i = difference in paired ranks

(b) n = number of cases

(c) i = paired score

(d) y_i =Effort Estimation

(e) x_i =Actual Effort

(f) $\bar{y}$ =Averaging

(g) $\bar{x}$ =Averaging

2.7 Related Work

Previous research descriptions are described as follows :

1. First proposed method of UCP to software development effort estimation [Karner, 1993]. UCP is better than the estimate of experts, the level of deviation UCP 19% and 20% of the experts [Anda, 2002)]. The results of estimation for 200 projects obtained less than 9% deviation between actual cost and estimation [Carrol, 2005]. The value of everyone's effort-hours (man-hour) in software development that is 8,2.

2. Ali Bou Nasif, Luiz Fernando Capretz and Danny Ho conducted a study to improve the accuracy of the estimation method of Use Case Points technique with Soft Computing in 2010. Background of the research because of the identification of the International Society of Parametrics Analysis and The Standish Group International that the estimation of the ugly as one of the causes of the failure of the software development. [Ali Bou Nasif, 2010]

3. Mohammed Wajahat Kamal and Moataz a. Ahmed do comparison method of calculating effort in software development projects in 2011.

Background the underlying problems they do such research as there are several methods that can be performed to calculate the estimate of effort at the beginning of the project, but it would be hard to do because the initial data less accurate and less detail. They did a comparison of several methods, namely: Use Case Points, Transactions, Paths, Extended Use Case Points, UCPm, Adapted Use Case Use Case Points, Size Points, Fuzzy Use Case Points, Simplified Use Case Points, and the Industrial use of Use Case Points. From the results of testing methods, it turns Fuzzy Use Case Points able to handle information that is not or less precision and transparency. [Kamal and Ahmed, 2011]

4. Mohammad Saber Iraji, Majid Motameni Aboutalebi and Homayun in 2011 to do research to find a method that can automatically determine the level of complexity of the use case at the time of effort estimation using use case points. Background problem underlying use case points because the method has a drawback in the form of uncertainty of factor costs and the determination of its classification less detail. The method suggested by them are using Neuro Fuzzy Use Case Points (NFUCP), and from the results of testing indicate if it's processing data by using more accurate and NFUCP approached the actual effort of developing a software. [Mohammad Saber Iraji, 2011]

5. Renny Sari Dewi, Apol Pribadi Subriadi and Sholiq in 2015 to do research for a new method of UCPabc variation of the extension methods Use Case Point (UCP) and Activity Based Costing (ABC). The underlying background is to find new methods for preparing the financing component of software development projects. From the results of testing of the method UCPabc, the estimated costs of software development has a 2.16% deviation, larger i.e. Rp 192,475,490. The number of such deviations obtained from a comparison of the estimation of UCPabc against the real company actual cost (USD 188,322,484) in five projects of software development in an environment of local governments. [Renny Sari Dewi, 2015]

6. Wahyu Kurniawan, Sholiq, Teguh Susanto in 2013 to do research using use case points for software development eight data of governance website. From the results of testing of the method is correlation between the value of the actual effort and the estimated effort using UCP has a very strong correlation. From this research produced the effort

rate as big as 5178 hours. The effort rate is much smaller than the value given by Karner. Case is possible for several reasons, among others: 1) Enginering software technology that is growing rapidly. 2) Manufacturing website using components. 3) Source of the internet so complete. [Wahyu Kurniawan, 2013]

Chapter 3

Research Methods

3.1 Research Framework

The framework used to map the mindset in this study. General overview of the research framework can be seen in Figure 3.1.

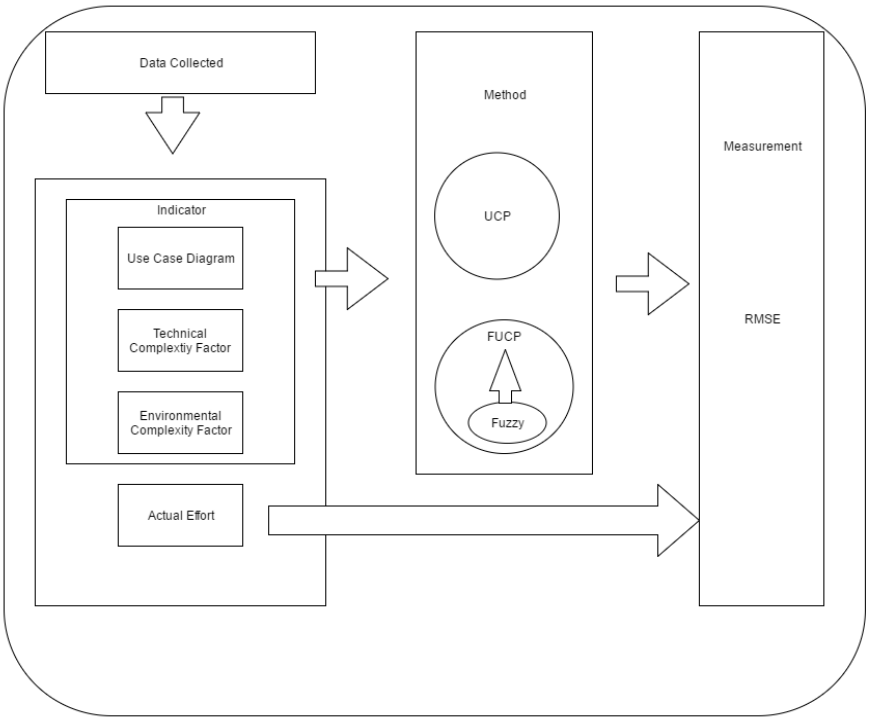


Figure 3.1: Research Framework

Research framework will be described in more detail on the process as follows :

1. Data collected

Data collection was conducted to obtain data required in the research, then used as indicators to help in the research process. The indicators used are :

(a) Use case diagram

Use case diagrams composed of actors and use cases. Actor will be the unadjusted variable Actor Weight (UAW) and the use case will be the unadjusted variable Use Case Weight (UUCW).

(b) Technical complexity factor

Technical Factor is a factor that influences the working techniques of software development.

(c) Environmental complexity factor

Environmental factors is a factor that influences the working of software development.

(d) Actual effort

Actual effort is the amount of time needed by programmers in completing the project, starting from the stage of planning, analysis, design to implementation.

2. Effort estimation calculation method

Calculation effort estimation with method Use Case Points (UCP) and Fuzzy Use Case Points (FUCP). When doing the calculation of estimated effort using the FUCP, there is a fuzzy influence on the calculation of estimated effort.

3. Measurement evaluation effort estimation model

Comparing UCP and FUCP are observed to compare values between two method. The value UCP and FUCP are compare between the smallest value using RMSE. The smaller are RMSE the closer to the value actual effort.

$$Min\{UCP\,,\,FUCP\}$$

3.2 Reseach Method

The research method is needed as a guide to be stages of processing can be run in a focused and systematic. The stages of research workmanship shown

by Figure 3.2.

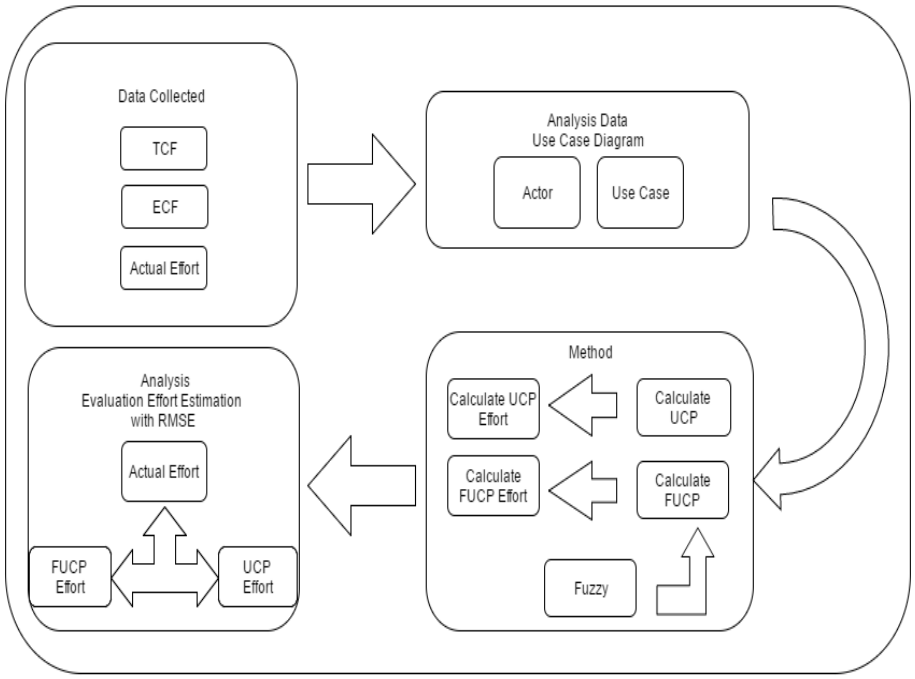


Figure 3.2: Research Method

Caption figure :

1. TCF = Technical complexity factor
2. ECF = Environmental complexity factor
3. UCP = Use Case Points in unit is project size
4. UCP Effort = Predict the real work in completing the project by reviewing the schedule, project size, even the project risk will be borne, unit is mandays or manhour.
5. FUCP = Fuzzy Use Case Points in unit is project size. Fuzzy use case points methods created from the expansion method use case points (UCP) with fuzzy logic Mamdani type.
6. Actual effort = Actual effort is the amount of time needed by programmers in completing the project, starting from the stage of planning, analysis, design to implementation.

7. FUCP Effort = Predict the real work in completing the project by reviewing the schedule, project size, even the project risk will be borne, unit is mandays or manhour. There is a fuzzy influence on the calculation of effort estimation.

Starting from data obtained from data collection techniques, namely technical factors, environmental factors, use case diagrams, and actual effort. The second stage, to analyze the use case diagram, actor and use case on a use case diagram becomes important things that are analyzed in this phase. When analyzing the use case diagram, actor and use case would be the reference for the calculation of estimated effort. Actor and use case analysis can then provides values to variables estimated calculation effort. The third stage calculating effort estimation using use case points (UCP) and fuzzy use case points (FUCP). When doing calculations using FUCP effort estimation, there is a fuzzy influence on the calculation effort etimation. The final stage is the evaluation of effort estimation methods. Activities such measurements are compared with FUCP and UCP method. Activities such evaluation will produce the best methods that appear to approach the value of actual effort.

3.3 Data Collection Technique

Data was collected through interviews, questionnaires, and observations documents. The data collection was conducted to obtain data required in the final project.

1. Interview

   Document specifications are not available on a software project makes data collection more difficult. Many teams of developers do not create a specification document on software development projects, therefore, the interview becomes a solution to get the detail information of software development projects. Interviews were conducted to each project team of software developers. Detail description of software projects that obtained during the interview include :

   (a) Actual effort of software development projects.

   (b) The technology used is currently working on a project.

2. Questionnaire

The questionnaire was given to each project team of software developers. The purpose of distributing questionnaire is to determine factors engineering and environmental factors that affect software development project in accordance with the real situation experienced.

3. Observation document

   Observation document is doing analysis of document specifications or analysis user guide software development projects, if the data is obtained when the interviews and questionnaires did not get the maximum, observation becomes the complement of the data collected.

3.4 Make Use Case Diagram

Before calculating use case points using use case points and fuzzy use case points, which should be done is to make use case diagram. The purpose of making use case diagram is to illustrate how a business system interacts with its environment. Tools used to simplify creating use case diagrams i.e. StarUML 2.7

3.5 Data Collected

The data obtained from data collection techniques are unadjusted Actor Weight (UAW), technical complexity factor (TCF), environmental complexity factor (ECF), and the actual effort. Data collection will be described in more detail as follows.

3.5.1 Unadjusted Actor Weight (UAW)

UAW value obtained from the classification of existing actors in the use case diagram. Classification of the actor can be seen in table 3.1 Actors who have been classified to be multiplied by a weighting factor. After getting the weight of each actor, and then do the accumulation of all the weights actors, resulting in UAW value required. UAW value calculation with the following formula :

$$UAW = (\sum SA * 1) + (\sum AA * 2) + (\sum CA * 3) \tag{3.1}$$

Description :

1. SA = Simple actor
2. AA = Average actor
3. CA = Complex actor

In the UCP, actors are classified as simple, average or complex. A weight is assigned to each category as follows :

1. Simple actor: This is described as another system through an API. Its weight is 1.
2. Average actor: This is described as another system interacting through a text-based user interface or a protocol. Its weight is 2.
3. Complex actor: This is described as a human interacting with system through a graphical user interface (GUI). Its weight is 3.

The UAW can be seen in Table 3.1.

Table 3.1: Unadjusted Actor Weight (UAW)

Actor Type	Description	Weighting Factor	Total Actor	Result
Simple	Interaction system with well defined API	1	SA	1 * SA
Average	Interaction system using a protocol based interface	2	AA	2 * AA
Complex	Human	3	CA	3 * CA
Total UAW				$(\sum SA * 1) + (\sum AA * 2) + (\sum CA * 3)$

Table description :

1. Actor type = type of actor.
2. Description = Description task actor in a use case.
3. Weighting factor = weighting factor the type of actor.
4. SA = number of simple actor
5. AA = number of average actor

6. CA = number of complex actor
7. Result = result of the multiplication between the number of actors with the weighting factors.
8. Total UAW = the result accumulation

3.5.2 Technical Complexity Factor (TCF)

TCF is a technical factors affecting software development projects. Technical factors affecting software development projects and the weight of each technical factors can be seen in table 3.2.

Table 3.2: Technical Complexity Factor (TCF)

Factor Number	Description	Weight	Assigned Value (0-5)	Result
T_1	Distributed System	2,0	A	2,0 * A
T_2	Response time or throughput performance objectives	1,0	B	1,0 * B
T_3	End-user online efficiency	1,0	C	1,0 * C
T_4	Complex internal processing	1,0	D	1,0 * D
T_5	Reusability of code	1,0	E	1,0 * E
T_6	Easy to install	0,5	F	0,5 * F
T_7	Ease of use	0,5	G	0,5 * G
T_8	Portability	2,0	H	2,0 * H
T_9	Ease of change	1,0	I	1,0 * I
T_{10}	Concurrency	1,0	J	1,0 * J
T_{11}	Special security objectives include	1,0	K	1,0 * K
T_{12}	Direct access for thrid parties	1,0	L	1,0 * L
T_{13}	Special user training required	1,0	M	1,0 * M
Total TF				(2,0*A)+...+(1,0*M)
TCF				$TCF = 0,6+(0,01*Total\,TF)$

Table description :

1. Factor number = number of factors
2. Weight = weight of technical factor
3. Assigned value = value of technical factors.

4. A-M = range value provided by the project developers team. The value starts from 0 to 5.

5. Result = result of the multiplication between the weight with the Assigned value.

6. Total TF = total summation of all the factors.

7. TCF = Technical complexity Factor

Technical factors and each weighting factor then rated to get the value of TCF. Scoring each technical factor is done by distributing questionnaires to each software project development team as it requires an objective assessment obtained by the project manager or project team.

The value assigned to each factor depends on the influence of these factors on the progress of software development projects. A value of 0 means no influence, the value of 3 means ordinary, and a score of 5 means that the technical factors have a major influence on the progress of the project. The given value is then multiplied by the weighting on the technical factors to get a total value of Technical Factor (TF), which is then carried accumulation of all TF. The accumulated value of TF and then used to calculate the Technical Complexity Factor (TCF) with the following formula :

$$TCF = 0,6 + (0,01 * Total\,TF) \tag{3.2}$$

3.5.3 Enviromental Complexity Factor (ECF)

In addition to TCF other factors that will count is the ECF. ECF is an environmental factor that affects software development projects. Environmental factors affecting software development projects and the weight of each environmental factor can be seen in Table 3.3.

Table 3.3: Enviromental Complexity Factor (ECF)

Factor number	Description	Weight	Assigned Value (0-5)	Result
E_1	Familiarity with system development process being used	1,5	A	1,5 * A
E_2	Application experience	0,5	B	0,5 * B
E_3	Object-oriented experience	1,0	C	1,0 * C
E_4	Lead analyst capability	0,5	D	0,5 * D
E_5	Motivation	1,0	E	1,0 * E
E_6	Requirements stability	2,0	F	2,0 * F
E_7	Part time staff	-1,0	G	-1,0 * G
E_8	Difficulty of programming language	-1,0	H	-1,0 * H
Total EF				(1,5*A)+...+(-1,0*H)
ECF				$ECF = 1,4 + (-0,03 * Total\ EF)$

Table description :

1. Factor number = number of factors
2. Weight = weight of technical factor
3. Assigned value = value of technical factors.
4. A-H = range value provided by the project developers team. The value starts from 0 to 5.
5. Result = result of the multiplication between the weight with the Assigned value.
6. Total EF = total summation of all the factors.
7. ECF = Environment complexity factor

Not much different from the TCF, ECF is the environmental factors and each has a weighting factor then rated to get the value of the ECF. Scoring each environmental factor is done by distributing questionnaires to each software project development team as it requires an objective assessment obtained by the project manager or project team.

The value assigned to each factor depends on the influence of these factors on the progress of software development projects. A value of 0 means no influence, the value of 3 means ordinary, and a score of 5 means that the technical factors have a major influence on the progress of the project. The value is then multiplied by the weighting given to the environmental factors are to get a total value of Environmental Factor (EF), which then do the accumulation of all the values of EF. The accumulated value of EF and then used to calculate contributions to the Enviromental Factor (ECF) by the following formula :

$$ECF = 1,4 + (-0,03 * Total\ EF) \tag{3.3}$$

3.5.4 Actual Effort

After performing stages data collection, the next thing to do is to calculate the actual effort. Actual effort is the first thing that obtained when conducting interviews. Actual effort obtained through interviews with the developers, especially with programmers and project managers software developers. Interview guidelines actual effort can be seen in Table 3.4.

Table 3.4: Actual Effort

No	Project Name	Person	Time		Working Days	Working Hours	Actual Effort
			Actual Start	Actual End			

In the table, visible the column person, working days, and working hours these. Three things next will be the reference to get the actual effort of a software development project. Formula calculates the actual effort as follows:

$$Actual\,Effort = \sum Person * \sum Working\,Days * Working\,Hours \quad (3.4)$$

3.6 Use Case Points Method

After makes use case diagrams using StarUML 2.7 and then calculate the project effort estimation using use case points (UCP). UCP results will be known after getting UUCP value, the value of TCF and ECF values. UCP value calculation with the following formula :

$$UCP = UUCP * TCF * ECF \quad (3.5)$$

Description :

1. UUCP = Unadjusted Use Case Points
2. TCF = Technical Complexity Factor
3. ECF = Environmental Complexity Factor

To get the value of UUCP, the value of TCF and ECF have to do an analysis that refers to the use case diagram. The values to be obtained from the reference to the use case diagram each software development project, namely :

1. Unadjusted Actor Weight (UAW).

   Explanation unadjusted actors weight (UAW) can be seen in section 3.5.1 and calculate the UAW with a formula that can be seen in the formula number 3.1.

2. Unadjusted Use Case Weights (UUCW)

   In addition to actors, each use case is classified based on the number of transactions. Classification use case can be seen in Table 3.5. Use case that has been classified multiplied by the weighting factor. After getting each weight for each use case, then do the accumulation of all the weights use case, resulting in a value UUCW needed. UUCW value calculation with the following formula :

$$UUCW = (\sum SU * 5) + (\sum AU * 10) + (\sum CU * 15) \quad (3.6)$$

Description :

(a) SU = Simple Use Case

(b) AU = Average Use Case

(c) CU = Complex Use Case

Use cases are classified based on the number of transactions in the success and alternative scenarios. A weight is assigned to each category as follows :

(a) Simple Use Case: A use case is classified as Simple if the number of transactions is ≤ 3. Its weight is 5.

(b) Average Use Case: A use case is classified as Average. If the number of the transactions is between 4 and 7. Its weight is 10.

(c) Complex Use Case: A use case is classified as Complex if the number of transactions is more than 7. Its weight is 15.

The UUCW can be seen in Table 3.5.

Table 3.5: Unadjusted Use Case Weights (UUCW)

Use Case Type	Description (Transaction)	Weighting Factor	Number	Result
Simple	1 - 3	5	SU	5* A
Average	4 - 7	10	AU	10 * B
Complex	> 7	15	CU	15 * C
Total UUCW				$(\sum SU * 5) + (\sum AU * 10) + (\sum CU * 15)$

Table description :

(a) Use case type = type of use case.

(b) Description = description of the transaction amount in a use case.

(c) Weighting factor = weighting factor type use case

(d) SU = number of simple use case

(e) AU = number of average use case

(f) CU = number of complex use case

(g) Result = result of the multiplication between the number of use case with the weighting factors.

(h) Total UUCW = result accumulation.

3. Unadjusted Use Case Points (UUCP)

   After getting the total value of the UAW and the total UCCW, then the next can be calculated by the following formula UUCP value :

$$UUCP = Total\,UAW + Total\,UUCW \tag{3.7}$$

   Description :

   (a) Total UAW = Total Unadjusted Actor Weight

   (b) Total UUCW = Total Unadjusted Use Case Weights

4. Technical Complexity Factor (TCF)

   Explanation of technical complexity factor (TCF) can be seen in section 3.5.2 and calculate TCF with a formula that can be seen in the formula number 3.2.

5. Enviromental Complexity Factor (ECF)

   Explanation of environmental complexity factor (ECF) can be found in Section 3.5.3 and calculate ECF with the formula that can be seen in the formula number 3.3.

3.7 Fuzzy Use Case Points Method

The concept fuzzy use case points methods created from the expansion method use case points (UCP) with fuzzy logic Mamdani type. This method is also used to calculate estimates of the project effort. This method is used to obtain the estimated effort of more varied value and investigate the causal relationship between the UCP with FUCP using the method of evaluation effort.

Fuzzy logic is applied to the method of use case points will affect the value of UCP. Fuzzy logic is implemented on UCP affect the result accuracy rate. Based on the weaknesses of the UCP which is high difference among the value of weight factor of UUCW. Fuzzy use case points modify the weighting factor UUCW. The value of the use case points (UCP) which has been combined with fuzzy called fuzzy use case points (FUCP).

In the fuzzy use case points method applicable rule that maximum classification complexity of the transaction use case is 14 transactions. It is based on the use case diagrams that were analyzed in ten of software development projects.

Tools that are used to facilitate the calculation of fuzzy logic is Matlab R2012b. Tools that are used to facilitate the calculation of fuzzy logic is Matlab R2012b. Fuzzy logic toolbox in processing data do stages as shown below, the process can be seen in Figure 3.3.

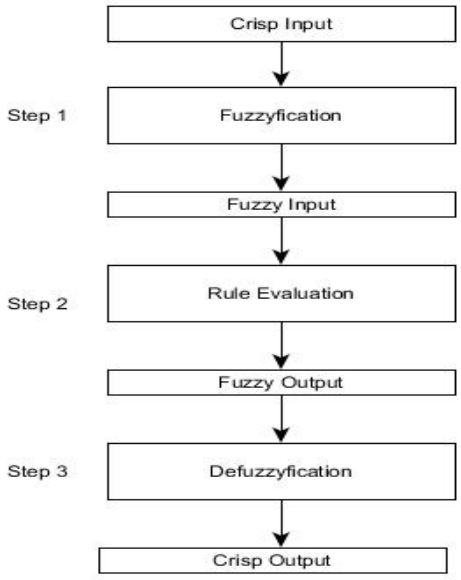


Figure 3.3: Fuzzy Logic [Kusumadewi, 2004]

Explanation of fuzzy logic as follows :

1. Crisp Input

   The raw data that will be used as input for fuzzy. The raw data is used as input is the number of transactions on the use case.

2. Fuzzyfication

   The process for changing the input of crisp becomes fuzzy (linguistic variables) is usually presented in fuzzy associations with a member function. Data crisp processed into fuzzy. Membership function which presents the classification of the transaction on the fuzzy use case points that is simple, average and complex. Data will be presented with three crisp triangular curve in accordance with the membership function. Fuzzyfication can be seen in Figure 3.9

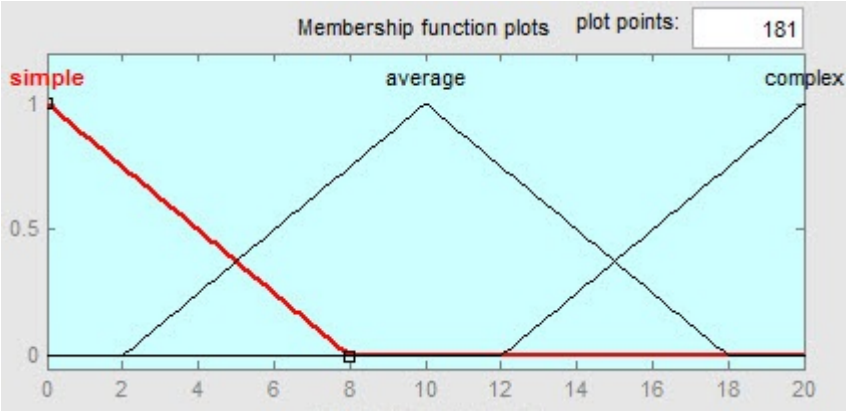


Figure 3.4: Membership Function

3. Rule Evaluation (Inference system)

   Rules that explain the relationship between variables input and output. Variables that are processed and the resulting shape fuzzy (linguistic variables). To explain the relationship between inputs and outputs usually use the "IF-THEN". Fuzzy logic in use case points requires rule evaluation as follows:

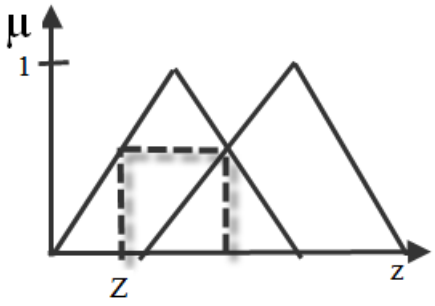
   (a) If the transaction is simple then the multiplier is simple.

   (b) If the transaction is average then the multiplier is average.

   (c) If the transaction is complex, then the multiplier is complex.

4. Defuzzyfication

   The process of converting fuzzy variables (linguistic variables) into the data bound (crisp) which can be further processed by the system. In deffuzyfication there are several methods that can be used. Fuzzy logic is implemented for use case points using centroid method. Formula centroid deffuzyfication process can be seen in figure 3.5 below.



$$Z = \frac{\int \mu_X(z).z\,dz}{\int \mu_x(z)\,dz}$$

Moment

Wide Area

Figure 3.5: Centroid Formula [Wardana, 2015]

Output fuzzy will be processed deffuzyfication. The formula states that to seek regional center implication, we have to divide the moment with wide area membership function. Wide area can be calculated by dividing the area of a triangle on a curve into rectangular areas or small triangular area. The division of the area can be seen as a figure 3.6.

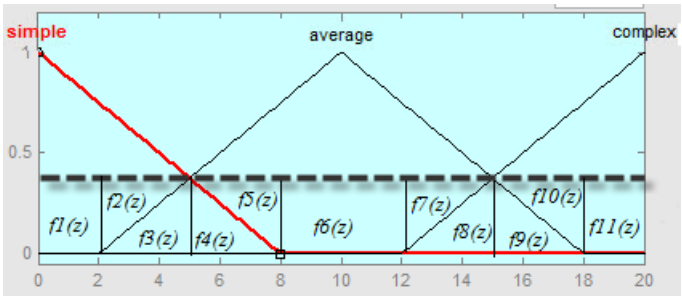


Figure 3.6: Division Area

Areas that have been divided, it can be inserted into an equation, the equation can be seen as an equation 3.7.

$$f(Z) = \left\{ \begin{array}{c} f_1(z),\ for\, 0 \leq z < 2 \\ f_2(z),\ for\, 2 \leq z < 5 \\ f_3(z),\ for\, 2 \leq z < 5 \\ f_4(z),\ for\, 5 \leq z < 8 \\ f_5(z),\ for\, 5 \leq z < 8 \\ f_6(z),\ for\, 8 \leq z < 12 \\ f_7(z),\ for\, 12 \leq z < 15 \\ f_8(z),\ for\, 12 \leq z < 15 \\ f_9(z),\ for\, 15 \leq z < 18 \\ f_{10}(z),\ for\, 15 \leq z < 18 \\ f_{11}(z),\ for\, 18 \leq z < 20 \end{array} \right\} \tag{3.8}$$

While the Moment can be obtained by multiplying each equation with a value of z, then do an integral process, as shown in the equation 3.8.

$$Z = \frac{\int_0^2 f_1(z)\,.z\,.\,dz}{\int_0^2 f_1(z)\,.\,dz} + \frac{\int_2^5 f_2(z)\,.z\,.\,dz}{\int_2^5 f_2(z)\,.\,dz} + ... + \frac{\int_{18}^{20} f_{11}(z)\,.z\,.\,dz}{\int_{18}^{20} f_{11}(z)\,.\,dz} \tag{3.9}$$

Description :

(a) Z = Centroid Formula

(b) (z).zdz = Moment

(c) (z)dz = Wide Area

5. Crisp Output

   The exact value for the result fuzzy processes required to process the data on a system that has been designed.

After the fuzzy logic process stages are completed, the new classification weighting factor on UUCW use case has been obtained through the fuzzy logic process. There is no difference in the stages of the process of calculating the estimate project effort using fuzzy use case points by using use case points, that the difference is the value of the weight factor of the classification of the transaction use case UUCW. Can be seen the difference the weighting factor classification transactions use case use case method on UUCW point with fuzzy use case points or FUUCW hereafter referred to in the table 3.6 and table 3.7 below :

Table 3.6: Classification Weighting Factor in Use Case Points Method

Use Case	Description (Transaction)	Weighting Factor
Simple	1	5
Simple	2	5
Simple	3	5
Average	4	10
Average	5	10
Average	6	10
Average	7	10
Complex	8	15
Complex	9	15
Complex	10	15
Complex	11	15
Complex	12	15
Complex	13	15
Complex	14	15

Table 3.7: Classification Weighting Factor in Fuzzy Use Case Points Method

Use Case	Description (Transaction)	Weighting Factor
Simple	1	4,79
Simple	2	4,67
Simple	3	5,70
Average	4	6,72
Average	5	7,70
Average	6	8,79
Average	7	10,30
Complex	8	11,60
Complex	9	12,50
Complex	10	13,70
Complex	11	15,00
Complex	12	15,00
Complex	13	15,00
Complex	14	15,00

calculation value FUCP with the following formula :

$$FUCP = FUUCP * TCF * ECF \tag{3.10}$$

Description :

1. FUUCP = Fuzzy Unadjusted Use Case Points

2. TCF = Technical Complexity Factor

3. ECF = Environmental Complexity Factor

To get FUUCP value, TCF value and ECF value have to do an analysis that refers to the use case diagram. Stages of the process calculating estimated project effort using fuzzy use case points are described as follows :

1. Unadjusted Actor Weight (UAW)

   Explanation unadjusted actors weight (UAW) can be seen in section 3.5.1 and calculate the UAW with a formula that can be seen in the formula number 3.1.

2. Fuzzy Unadjusted Use Case Weights (FUUCW)

   In addition to actors, each use case is classified based on the number of transactions. Use case that has been classified in accordance with the weight factor. After getting each weight for each use case, then do the accumulation of all the weights factor use case, resulting in a value FUUCW needed. FUUCW calculation formula is :

   $$FUUCW = Total\,Weight\,Factor\,Use\,Case \tag{3.11}$$

   Use cases are classified based on the number of transactions in the success and alternative scenarios. A weight is assigned to each category as follows :

   (a) Simple Use Case: A use case is classified as Simple if the number of transactions is ≤ 3. Its weight is 5.

   (b) Average Use Case: A use case is classified as Average. If the number of the transactions is between 4 and 7. Its weight is 10.

   (c) Complex Use Case: A use case is classified as Complex if the number of transactions is more than 7. Its weight is 15.

   The FUUCW can be seen in Table 3.8.

Table 3.8: Classification Weighting Factor FUUCW in Fuzzy Use Case Points Method

Use Case	Description (Transaction)	Weighting Factor
Simple	1	4,79
Simple	2	4,67
Simple	3	5,70
Average	4	6,72
Average	5	7,70
Average	6	8,79
Average	7	10,30
Complex	8	11,60
Complex	9	12,50
Complex	10	13,70
Complex	11	15,00
Complex	12	15,00
Complex	13	15,00
Complex	14	15,00
Total UUCW		4,79+4,67+...+15+15

3. Fuzzy Unadjusted Use Case Points (FUUCP)

   After getting total value UAW and total UUCW, then the next can be calculated FUUCP value with the following formula :

$$FUUCP = Total\,UAW + Total\,FUUCW \tag{3.12}$$

   Description :

   (a) Total UAW = Total Unadjusted Actor Weight

   (b) Total FUUCW = Total Fuzzy Unadjusted Use Case Weights

4. Technical Complexity Factor (TCF)

   Explanation of technical complexity factor (TCF) can be seen in section 3.5.2 and calculate TCF with a formula that can be seen in the formula number 3.2.

5. Enviromental Complexity Factor (ECF)

Explanation of environment complexity factor (ECF) can be seen in section 3.5.3 and calculate ECF with a formula that can be seen in the formula number 3.3.

3.8 Use Case Points (UCP) Effort

UCP effort is a technique to change value of UCP into value in the form person man hours (PHM). UCP effort illustrates size of the workforce required in hours. UCP effort can be calculated after UCP results have been obtained.

Based on Schneider and Winters to the estimated effort, PHM value is influenced by the Environmental Complexity Factor in UCP. Environmental Complexity Factor (ECF) on UCP can be seen in Table 3.3 Value PHM obtained by the following rules :

1. F1 = Total weighted value E1 to E6 which has a value <3
2. F2 = Total weighted value E7 to E8 which has a value> 3
3. If F1 + F2 <= 2 then PHM = 20
4. If F1 + F2 = 3 or 4 then PHM = 28
5. If F1 + F2> 4 then the project should be canceled

UCP effort calculation with the following formula :

$$Effort = UCP * PHM \tag{3.13}$$

Description :

1. UCP = Use Case Points
2. PHM = Person Man Hours

3.9 Fuzzy Use Case Points (FUCP) Effort

FUCP effort and UCP effort is not much different, both techniques change the value estimation of the effort of the project. FUCP effort is a technique to change value ofF UCP into value in the form person man hours (PHM). FUCP effort illustrates size of the workforce required in hours. FUCP effort can be calculated after FUCP results have been obtained.

Based on Schneider and Winters to the estimated effort, PHM value is influenced by the Environmental Complexity Factor in FUCP. Environmental Complexity Factor (ECF) on FUCP can be seen in Table 3.3. Value PHM obtained by the following rules :

1. F1 = Total weighted Value E1 to E6 which has a value <3
2. F2 = Total weighted Value E7 to E8 which has a value> 3
3. If F1 + F2 <= 2 then PHM = 20
4. If F1 + F2 = 3 or 4 then PHM = 28
5. If F1 + F2> 4 then the project should be canceled

FUCP effort calculation with the following formula :

$$Effort = FUCP * PHM \tag{3.14}$$

Description :

1. FUCP = Fuzzy Use Case Points
2. PHM = Person Man Hours

3.10 Analysis Evaluation Effort Estimation Method

Analysis is a statistical technique used to test whether there is any correlation between the UCP with method FUCP method and compare the value of UCP effort with FUCP effort. Evaluation method effort estimation used to perform analysis of the correlation that is Root Mean Square Error (RMSE), R square, adjusted R square and Spearman's rank order correlation which is a very common criterion used to evaluate the software effort estimation model.

When using Root Mean Square Error (RMSE) to evaluation between actual with estimation, good results are shown with a low value. A low value indicates that value of UCP and FUCP approached the accuracy value actual effort. Low value indicates that FUCP do the repair method UCP in doing calculation effort estimation. Low values indicate an increase in accuracy approach actual effort. Specifically for R square and adjusted R square to be good models if the indicator measuring goodness of fit models high value or

closer to 1. In Spearman's rank-order correlation a monotonic relationship is a relationship that does one of the following as the value of one variable increases, so does the value of the other variable or as the value of one variable increases, the other variable value decreases.

In table 3.9 shows the evaluation formula was conducted on four different criteria, these include R square, adjusted R square, RMSE and Spearman's rank-order correlation

Table 3.9: Models Evaluation

Models Evaluation	Formula
R square	$R^2 = 1 - \frac{SSR}{SST} = 1 - \frac{\sum(\hat{y}_i - \bar{y})}{\sum(y_i - \bar{y})}$
Adjusted R square	$Adjusted\, R^2 = 1 - \frac{(1-R^2)-(N-1)}{N-M-1}$
RMSE	$RMSE = \sqrt{\frac{\sum(Actual\, effort - Effort\, estimastion)^2}{n}}$
Spearman's Rank	$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$ *or* $\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$

Chapter 4

Result and Analysis

4.1 Data Collection

The data obtained will be used as an indicator that helps in the process of research.

4.1.1 Technology Project Requirements

Technology project requirements is the technology used in the making of the project. List of technology project requirements refers to the specification documents the making of the project, but some of the developers do not have a specification document, so that the project needs the data obtained from the interviews, observations website, as well as utilize the user guide is obtained from the developer. To facilitate the process of preparing the tables in this study, then do the code project as initialization to each software project. Browse all software projects can be seen in table 4.1.

Table 4.1: Technology Requirements All of Software Project

Project Code	Project	Person	Technology	Framework
A	Evaluation Kemenpora Website (Website Evaluasi Kemenpora)	4	PHP, HTML, MySQL, Apache	Cframe
B	Financal System of Dikti (Sistem Keuangan Dikti)	4	PHP, HTML, MySQL, Apache	Cframe
C	SIMAYA System (Sistem SIMAYA)	4	Node JS, HTML, Mongo DB	Express JS
D	Inventory System of Pekalongan City (SIMSEDIA)	6	PHP, HTML, MySQL, Apache	Yframe
E	Geographical Information System (GIS) Website of Kemenpora	2	PHP, HTML, Maria DB, Apache	Cframe
F	Sportscience Website of Kemenpora	4	PHP, HTML, MySQL, Apache	-
G	Assesment Library Website of Kemdikbud (Website Perpustakaan Penilaian Kemdikbud)	2	PHP, HTML, MySQL, Apache	Code Igniter
H	Internasional Study Website of Kemdikbud (MINITES Kemdikbud)	2	PHP, HTML, MySQL, Apache, Moodle	-
I	Biodiversity Mobile Apllication (Aplikasi Biodiversity)	4	Android Studio, Java, API	-
J	BSN E-learning Website (Website E-learning BSN)	6	PHP, HTML, JQuery, MySQL, Apache, Smarty	Codekir

4.1.2 Use Case Diagram

Use case diagram composed of use cases and actors describe what can be done by the system. At this stage of use case diagram illustrates the identification of the business process system. After knowing the actors that play a role in the system, then that is getting any process that can be performed using the system, and is further illustrated in a use case.

In this study as a discussion, the Assesment Library Website development project is taken as a data example for this research. The main page Assesment Library Website can be seen in figure 4.1.

Figure 4.1: Main Page Assesment Library Website [Penilaian Homepage, 2014]

Use case diagrams Assesment Library Website can be seen in figure 4.2.

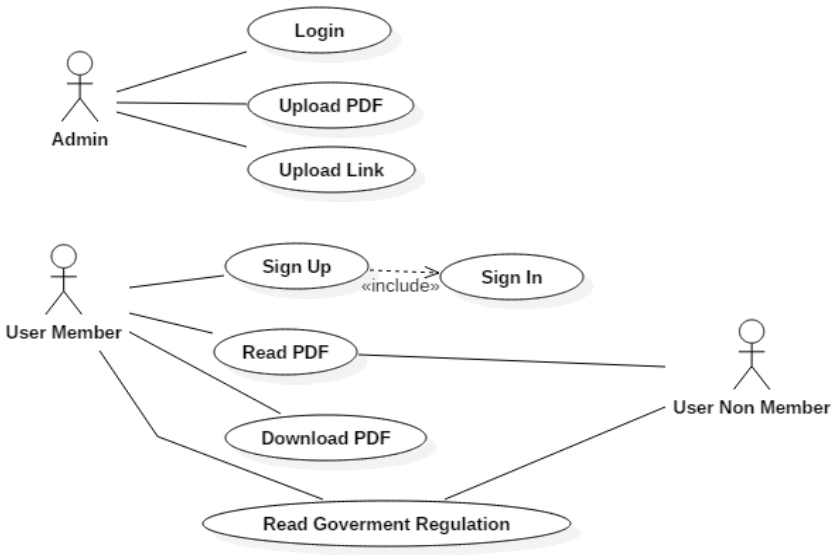


Figure 4.2: Use Case Diagrams Assesment Library Website

From Figure 4.2, it is known that there are three actors and eight use cases. Actor on the use case diagram will be used as a variable unadjusted Actor Weight (UAW). The actor will be clarified by type. The kind of actor can be seen in Table 2.1.

Use case on a use case diagram will be used as a variable unadjusted Use Case Weight (UUCW). Each use case is classified based on the number of transactions, or type. Type of use case can be seen in Table 2.2.

Overall use case diagram software development projects can be seen on appendix with a description of the location can be seen in Table 4.2 as follows :

Table 4.2: Description The Layout Use Case Diagram All of The Software Projects

Project Code	Related Document
A	Appendix B.1
B	Appendix B.2
C	Appendix B.3
D	Appendix B.4
E	Appendix B.5
F	Appendix B.6
G	Appendix B.7
H	Appendix B.8
I	Appendix B.9
J	Appendix B.10

Description of the project as the initialization code for each software project can be seen in Table 4.1.

4.1.3 Unadjusted Actor Weight (UAW) Value

Based on table 2.1 of each actor in the system are classified as simple, average, complex. Based on the use case diagram Figure 4.2 use case diagrams Assesment Library Website, classification actors involved to generate value unadjusted Actor Weight (UAW) can be seen in Table 4.3.

Table 4.3: UAW Assesment Library Website

Actor Type	Description	Weighting Factor	Total Actor	Result
Simple	Interaction system with well defined API	1	0	0
Average	Interaction system using a protocol based interface	2	0	0
Complex	Human	3	3	9
Total UAW				9

All actors involved in the Assesment Library Website is a human, then the type of actor is complex. Based on table 4.3, the value of the UAW on the Assesment Library Website is nine. The results obtained from the formula UAW that can be seen in the formula number 2.2. UAW calculation Assesment Library Website is described as follows :

$$
\begin{aligned}
UAW &= (\sum SA * 1) + (\sum AA * 2) + (\sum CA * 3) \\
UAW &= (1 * 0) + (2 * 0) + (3 * 3) \\
UAW &= 9
\end{aligned}
$$

The overall results UAW values software development projects can be seen in Table 4.4.

Table 4.4: Value UAW All of Software Project

Project Code	UAW Value
A	6
B	9
C	12
D	6
E	6
F	12
G	9
H	6
I	3
J	6

Description of the project as the initialization code for each software project can be seen in Table 4.1.

4.1.4 Technical Complexity Factor (TCF) Value

Technical complexity factor is the value of technical factors affecting software development projects. TCF value obtained from questionnaires distributed to software project development team. TCF value will be used in the calculation of effort estimation. TCF value Assesment Library Website can be seen in table 4.5.

Table 4.5: Technical Complexity Factor (TCF) Assesment Library Website

Factor Number	Description	Weight	Assigned Value (0-5)	Result
T_1	Distributed System	2,0	5	10,00
T_2	Response time or throughput performance objectives	1,0	3	3,00
T_3	End-user online efficiency	1,0	5	5,00
T_4	Complex internal processing	1,0	1	1,00
T_5	Reusability of code	1,0	5	2,50
T_6	Easy to install	0,5	5	2,50
T_7	Ease of use	0,5	5	10,00
T_8	Portability	2,0	5	5,00
T_9	Ease of change	1,0	5	5,00
T_{10}	Concurrency	1,0	5	5,00
T_{11}	Special security objectives include	1,0	4	4,00
T_{12}	Direct access for thrid parties	1,0	3	3,00
T_{13}	Special user training required	1,0	1	1,00
Total TF				57
TCF				1,17

The value of technical complexity factor Assesment Library Website shown in Table 4.5 is 1,17. Based on the calculation formula number 2.5 technical complexity factor (TCF) Assesment Library Website is described as follows :

$$TCF = 0,6 + (0,01 * Total\,TF)$$
$$TCF = 0,6 + (0,01 * 57)$$
$$TCF = 1,17$$

Overall technical complexity factor (TCF) values software development project can be seen in Table 4.6.

Table 4.6: TCF Value All of Software Project

Project Code	TCF Value
A	1,21
B	1,18
C	1,19
D	1,14
E	1,21
F	1,13
G	1,17
H	1,16
I	1,13
J	1,04

4.1.5 Environmental Complexity Factor (ECF) Value

Environmental complexity factor is the value of environmental factors affecting software development projects. ECF value obtained from questionnaires distributed to software project development team. ECF value will be used in the calculation of effort estimation . ECF value Assesment Library Website can be seen in Table 4.7.

Table 4.7: ECF Value Assesment Library Website

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	5	7,50
E_2	Application experience	0,5	4	2,00
E_3	Object-oriented experience	1,0	5	5,00
E_4	Lead analyst capability	0,5	4	2,00
E_5	Motivation	1,0	4	4,00
E_6	Requirements stability	2,0	5	10,00
E_7	Part time staff	-1,0	5	-5,00
E_8	Difficulty of programming language	-1,0	1	-1,00
Total EF				24,50
ECF				0,67

Environmental complexity factor value Assesment Library Website shown in Table 4.7 is 0,665, then rounding up 2 decimal places then to 0,67. Based on the calculation formula number 2.6 the environmental complexity factor (ECF) Assesment Library Website is described as follows :

$$\begin{aligned} ECF &= 1,4 + (-0,03 * Total\,EF) \\ ECF &= 1,4 + (-0,03 * 24,50) \\ ECF &= 0,67 \end{aligned}$$

Overall environmental complexity factor (ECF) values software development project can be seen in Table 4.8.

Table 4.8: ECF Value All of Software Project

Project Code	ECF Value
A	0,77
B	0,77
C	0,74
D	0,71
E	0,67
F	0,77
G	0,67
H	0,85
I	0,70
J	1,00

4.1.6 Actual Effort Value

Actual effort value is the amount of time required by the programmer in completing the project, starting from the stage of planning, analysis, design to implementation. Actual value effort expressed in hours.

The data required to calculate the actual effort value is the person, working days, and working hours. On the Assesment Library Website has a person is two people, working days Assesment Library Website is 101 days and do the project for five hours each day. Based on the formula number 3.4, following the calculation actual effort value Assesment Library Website is described as follows :

$$\begin{aligned} Actual\,Effort &= \sum Person * \sum Working\,Days * Working\,Hours \\ Actual\,Effort &= 2 * 101 * 5 \\ Actual\,Effort &= 1010\,hours \end{aligned}$$

The overall results of the calculation actual effort value of software development projects can be seen in Table 4.9.

Table 4.9: Actual Effort Value All of Software Project

Project	Person	Working Days	Working Hours	Actual Effort
Evaluasi Kemenpora System	4	81	5	2280
Keuangan Dikti System	4	121	5	2420
SIMAYA System	4	528	5	4440
SIMSEDIA System	6	183	5	5490
GIS Kemenpora Website	2	129	5	1290
Sport Science Website	4	174	5	3480
Assesment Library Website	2	80	5	1010
Minites Website	2	80	5	2720
Mobile Biodiversity Application	4	42	5	840
BSN Website Portal	6	122	5	5310

4.2 Use Case Points Calculation

Effort estimation of software development can be calculated using use case points should follow these steps :

4.2.1 Unadjusted Actor Weight (UAW)

UAW is a classification type of actors that interact with the system. Based on the type of actor table 2.1 are classified as simple, average, complex, calculation UAW value Assesment Library Website can be seen in section 4.1.3 in Table 4.3.

4.2.2 Unadjusted Use Case Weight (UUCW)

Based on the type of use case table 2.2 are classified as simple, average, complex. Based on the use case diagram figure 4.2 Assesment Library Web-

site, classifications use case involved generating unadjusted value Use Case Weight (UUCW) can be seen in Table 4.10.

Table 4.10: UUCW Value Assesment Library Website

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	5	25
Average	4-7 Transaction	10	2	20
Complex	> 7 Transaction	15	1	15
Total				60

According to the table 4.10 UUCW value on the Assesment Library Website is 60. The results obtained from the formula UUCW which can be seen in the formula number 2.3. Calculation UUCW Assesment Library Website is described as follows :

$$UUCW = (\sum SU * 5) + (\sum AU * 10) + (\sum CU * 15)$$
$$UUCW = (5 * 5) + (2 * 10) + (1 * 15)$$
$$UUCW = 60$$

Overall results UCCW value software development projects can be seen in Table 4.11.

Table 4.11: UUCW Value All of Software Project

Project Code	UUCW Value
A	130
B	135
C	200
D	375
E	115
F	200
G	60
H	175
I	50
J	395

4.2.3 Unadjusted Use Case Points (UUCP)

UAW and UUCP value is used as a material for calculating the value of UUCP. Based on the calculation formula number 2.4 UUCP Assesment Library Website is described as follows :

UUCP = UAW + UUCW
UUCP = 9 + 60
UUCP = 69

The overall results UUCP value of software development projects can be seen in Table 4.12.

Table 4.12: UUCP Value All of Software Project

Project Code	UUCP Value
A	136
B	149
C	212
D	381
E	121
F	217
G	69
H	181
I	53
J	401

4.2.4 Technical Complexity Factor (TCF)

The value of technical complexity factor becomes one important part that is used to calculate effort estimation using UCP. The calculation technical complexity factor (TCF) value Assesment Library Website can be seen in section 4.1.4 in table 4.5.

4.2.5 Environmental Complexity Factor (ECF)

The value of environmental complexity factor becomes one important part that is used to calculate effort estimation using UCP. The calculation environmental complexity factor (TCF) value Assesment Library Website can be seen in section 4.1.5 in table 4.7.

4.2.6 Use Case Points (UCP)

After successfully calculate the value of the UAW, UUCW, UUCP, TCF, and ECF furthermore calculate the UCP value . Based on the materials collected, the value of the UAW in section 4.1.3, the value UUCW in section 4.2.2, the value of UUCP in section 4.2.3, the value in section 4.1.4 TCF and ECF values in section 4.1 .5 and based on the formula numbers 2.1 then the value of UCP Assesment Library Website is 54.0891, and rounding up 2 decimal point then becomes 54.09. UCP calculation Assesment Library Website is described as follows :

$$UCP = UUCP * TCF * ECF$$
$$UCP = 69{,}00 * 1{,}17 * 0{,}67$$
$$UCP = 54{,}0891 \approx 54{,}09$$

The overall results UCP value software development projects can be seen in Table 4.13.

Table 4.13: UCP Value All of Software Project

Project Code	UAW	UUCW	UUCP	TCF	ECF	UCP
A	6	130	136	1,21	0,77	126,72
B	9	135	149	1,18	0,77	135,39
C	12	200	212	1,19	0,74	186,39
D	6	375	381	1,14	0,71	308,39
E	6	115	121	1,21	0,67	98,10
F	12	200	217	1,13	0,77	188,82
G	9	60	69	1,17	0,67	54,09
H	6	175	181	1,16	0,85	178,47
I	3	50	53	1,13	0,70	41,93
J	6	395	401	1,04	1,00	417,04

4.3 Fuzzy Use Case Points

The method of calculating an effort estimation using fuzzy use case points are modification of the use case points. Essentially there is no difference in the stages of the process calculate an efforts estimation use fuzzy use case points with the using use case points but that the difference is the weight factor value classification transactions use case UUCW.

Using the same data from the Assesment Library Website, calculation effort estimation using fuzzy use case points. UAW, TCF, and ECF value use the value obtained by using use case points. Only one category given the influence of fuzzy logic, namely weighting factor UUCW while other categories of data processed using the same method of use case points. UUCW value for the Assesment Library Website using fuzzy logic influence can be seen in section 4.3.2. Effort estimation of software development can be calculated fuzzy using use case points should follow these steps :

4.3.1 Unadjusted Actor Weight (UAW)

UAW is a classification type of actors that interact with the system. The value of the UAW using the same data as the effort estimation calculation use case

points method. Based on the type of actor table 2.1 are classified as simple, average, complex, calculation UAW value Assesment Library Website can be seen in section 4.1.3 in Table 4.3.

4.3.2 Fuzzy Unadjusted Use Case Weight (FUUCW)

UUCW weighting factor on use case points method has weaknesses which is high difference among the value of weight factor of UUCW. FUUCW is UUCW weighting factor that has been given the influence of fuzzy logic to correct the difference in value between the weighting factors. Comparison UUCW weighting factor on the method of use case points (UCP) and weighting factors FUUCW on fuzzy methods use case points (FUCP) can be seen in Table 3.7 and Table 3.8

In the fuzzy use case points (FUCP) method be applied a rule that the maximum classification complexity of the transaction use case is 14 transactions. It is based on the use case diagrams that were analyzed in ten of software development projects.

Based on the use case diagram Figure 4.2 Assesment Library Website has 8 use case. A detailed explanation of the classification of the use case involved generating Fuzzy unadjusted value Use Case Weight (FUUCW) can be seen in Table 4.14.

Table 4.14: FUUCW Value Assesment Library Website

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Login	3	5,70	5,70
2	Upload PDF	7	10,30	10,30
3	Upload Link	4	6,72	6,72
4	Sign Up	10	13,70	13,70
5	Sign In	3	5,70	5,70
6	Read PDF	1	4,79	4,79
7	Download PDF	1	4,79	4,79
8	Read Goverment Regulation	1	4,79	4,79
FUUCW Value				56,49

Based on table 4.14, the value of FUUCW on the Assesment Library Website is 56.49. The calculation of the value of the FUUCW Assesment Library Website is explained as follows :

FUUCW = $\sum Value$
FUUCW = 5,70 + 10,30 + 6,72 + 13,70 + 5,70 + 4,79 + 4,79 + 4,79
FUUCW= 56,49

The overall results of the FUUCW value can be seen in appendix with description of the layout can be seen in table 4.15 as follows :

Table 4.15: Description The Layout FUUCW Value All of The Software Projects

Project Code	Document Related FUUCW Value
A	Appendix D.1
B	Appendix D.2
C	Appendix D.3
D	Appendix D.4
E	Appendix D.5
F	Appendix D.6
G	Appendix D.7
H	Appendix D.8
I	Appendix D.9
J	Appendix D.10

4.3.3 Fuzzy Unadjusted Use Case Points (FUUCP)

FUUCP is the accumulated value of the UAW and FUUCW. Differences FUUCP with UCP is FUUCW values that affect the calculation, although basically have the same pattern of calculation formula. Based on the values calculated by formula number 2.9 FUUCP Assesment Library Website.

$$\text{FUUCP} = \text{UAW} + \text{FUUCW}$$
$$\text{FUUCP} = 9{,}00 + 56{,}49$$
$$\text{FUUCP} = 65{,}49$$

The overall results FUUCP value of software development projects can be seen in Table 4.16.

Table 4.16: FUUCP Value All of Software Project

Project Code	FUUCP Value
A	123,82
B	126,29
C	165,38
D	327,13
E	88,15
F	192,77
G	65,49
H	161,67
I	39,56
J	240,60

4.3.4 Technical Complexity Factor (TCF)

The value of the TCF using the same data as with effort estimation calculation use case points method. The value of technical complexity factor becomes one important part that is used to calculate effort estimation using UCP. The calculation technical complexity factor (TCF) value Assesment Library Website can be seen in section 4.1.4 in table 4.5.

4.3.5 Environmental Complexity Factor (ECF)

The value of the ECF using the same data as with effort estimation calculation use case points method. The value of environmental complexity factor becomes one important part that is used to calculate effort estimation using UCP. The calculation environmental complexity factor (TCF) value Assesment Library Website can be seen in section 4.1.5 in table 4.7.

4.3.6 Fuzzy Use Case Points (FUCP)

After successfully calculate the value of the UAW, FUUCW, FUUCP, TCF, and ECF furthermore calculate the value FUCP. Based on the materials collected, the value of the UAW in section 4.1.3, the value FUUCW in section 4.3.2, the value FUUCP in section 4.3.3, the value in section 4.1.4 TCF and ECF values in section 4.1.5 and based on the formula numbers 2.9 then FUCP value Assesment Library Website is 51.337611 votes, then rounding up 2 decimal point then becomes 51.34.

Calculation FUCP Assesment Library Website is described as follows :

$$\text{FUCP} = \text{FUUCP} * \text{TCF} * \text{ECF}$$
$$\text{FUCP} = 65{,}49 * 1{,}17 * 0{,}67$$
$$\text{FUCP} = 51{,}337611 \approx 51{,}34$$

The overall results FUCP value software development projects can be seen in Table 4.17.

Table 4.17: FUCP Value All of Software Project

Project Code	UAW	FUUCW	FUUCP	TCF	ECF	FUCP
A	6	117,82	123,82	1,21	0,77	115,37
B	9	117,29	126,29	1,18	0,77	114,75
C	12	153,38	165,38	1,19	0,74	145,64
D	6	321,13	327,13	1,14	0,71	264,78
E	6	82,15	88,15	1,21	0,67	71,47
F	12	180,77	192,77	1,13	0,77	167,73
G	9	56,49	65,49	1,17	0,67	51,34
H	6	155,67	161,67	1,16	0,85	159,41
I	3	36,56	39,56	1,13	0,70	31,30
J	6	234,60	240,60	1,04	1,00	250,23

4.4 Calculation UCP Effort

UCP effort illustrates size of the workforce required in hours. UCP effort can be calculated after UCP results have been obtained. UCP value according to Section 4.2.6 is 54.09. Based on the formula UCP number 2.7 calculation UCP effort requires the value of PHM. PHM is a person man hours are influenced by environmental factors. The amount of the PHM is determined based on value assigned by a team of developers on factors ECF. Rules determining PHM great value can be found in section 2.2.4. Based on data from table 4.7 it can be seen the value of PHM is 20. Calculation of UCP effort Assesment Library Website is described as follows :

$$\text{UCP Effort} = \text{UCP} * \text{PHM}$$
$$\text{UCP Effort} = 54{,}09 * 20$$
$$\text{UCP Effort} = 1081,\ 79 \approx 1082 \text{ hour}$$

Overall results UCP effort value of software development projects can be seen in Table 4.18.

Table 4.18: UCP Effort Value All of Software Project

Project Code	UCP	PHM	UCP Effort
A	126,72	20	2535
B	135,39	20	2708
C	186,39	20	3734
D	308,39	20	6168
E	98,10	20	1962
F	188,82	20	3777
G	54,09	20	1082
H	178,47	20	2570
I	41,93	20	839
J	417,04	20	8341

4.5 Calculation FUCP Effort

FUCP effort illustrates size of the workforce required in hours. FUCP effort can be calculated after FUCP results have been obtained. FUCP value according to Section 4.3.6 is 51,34. Based on the formula FUCP number 2.10 calculation FUCP effort requires the value of PHM. PHM is a person man hours are influenced by environmental factors. The amount of the PHM is determined based on value assigned by a team of developers on factors ECF. Rules determining PHM great value can be found in section 2.5.1 number 7. Based on data from table 4.7 it can be seen the value of PHM is 20. Calculation of FUCP effort Assesment Library Website is described as follows :

$$\text{FUCP Effort} = \text{FUCP} * \text{PHM}$$
$$\text{FUCP Effort} = 51{,}34 * 20$$
$$\text{FUCP Effort} = 1026,76 \approx 1027 \text{ hour}$$

Overall results FUCP effort value of software development projects can be seen in Table 4.19.

Table 4.19: FUCP Effort Value All of Software Project

Project Code	FUCP	PHM	FUCP Effort
A	115,37	20	2308
B	114,75	20	2295
C	145,64	20	2913
D	264,78	20	5296
E	71,47	20	1430
F	167,73	20	3355
G	51,34	20	1027
H	159,41	20	3189
I	31,30	20	626
J	250,23	20	5005

4.6 Evaluation Effort Method

Evaluation of estimation methods used to test the accuracy of a model estimation. Evaluation of effort estimation method uses statistical techniques Root Mean Square Error (RMSE), R square, adjusted R square and Spearman's rank order correlation which is a very general model used to evaluate the estimation model. RMSE value is the average value of the square of the difference between the actual value and the estimated value.

When using the RMSE for evaluation, good results shown by a low value. A low value indicates that the effort estimation method approaches the accuracy of the actual value effort. Specifically for R square and adjusted R square to be good models if the indicator measuring goodness of fit models high value or closer to 1. In Spearman's rank-order correlation a monotonic relationship is a relationship that does one of the following as the value of one variable increases, so does the value of the other variable or as the value of one variable increases, the other variable value decreases.

4.6.1 Evaluation Model for UCP Method

This section presents the evaluation of UCP method. A comparison is performed between UCP method with actual effort. UCP method and actual effort is used as material for calculating the evaluation based model. The implementation of UCP to calculate effort estimation in ten projects used in this research has shown that UCP has RMSE is 699,4. Display train data with a fitted polynomial, degree is one, and 95% prediction bounds intervals. Figure 4.3 shows the evaluation result of UCP method and actual effort.

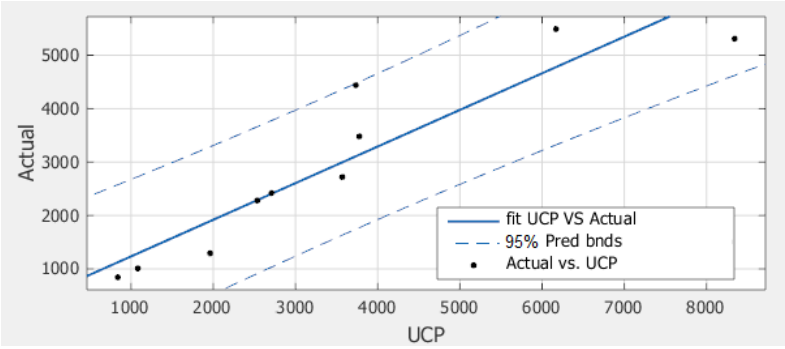


Figure 4.3: Polynomial Graph UCP Method Versus Actual Effort

4.6.2 Evaluation Model for FUCP Method

This section presents the evaluation of FUCP method. A comparison is performed between FUCP method with actual effort. FUCP method and actual effort is used as material for calculating the evaluation based model. The implementation of FUCP to calculate effort estimation in ten projects used in this research has shown that FUCP has RMSE is 547,1. Display train data with a fitted polynomial, degree is one, and 95% prediction bounds intervals. Figure 4.4 shows the evaluation result of FUCP method and actual effort.

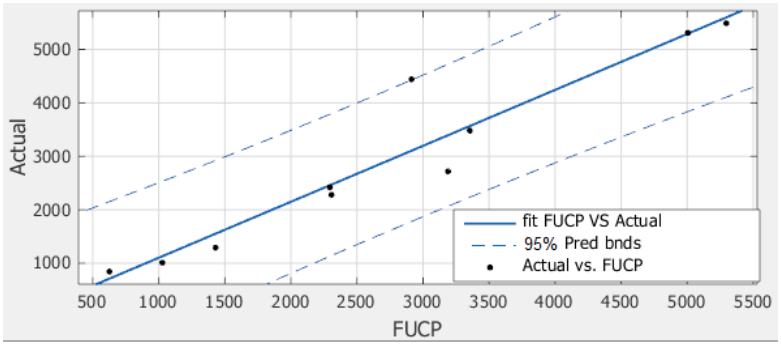


Figure 4.4: Polynomial Graph FUCP Method Versus Actual Effort

4.7 Correlation Evaluation Effort Method

Comparing UCP and FUCP are observed to compare values between two method with actual effort. Table 4.20 shows the evaluation results of UCP and FUCP method with actual effort was conducted on four different criteria. The implementation of FUCP to calculate effort estimation in ten projects used in this research has shown that FUCP has the closest value to the actual effort. It is also demonstrated that FUCP outperform. FUCP in terms of

accuracy by 6,51 % improvements based on adjusted R square and low value of RMSE the closer to the value actual effort.

Table 4.20: Correlation Evaluation Effort Method

Models Evaluation	UCP	FUCP
R Squaare	0.8510	0,9088
Adjusted R Square	0.8324	0.8975
RMSE	699.4	547.1
Spearman's Rank	0.9758	0.9515

Chapter 5

Conclusion and Suggestion

5.1 Conclusion

The conclusion that can be drawn from this study which is as follows :

1. The data were obtained by interview, questionnaire and refer to the documents the specifications of software projects that have been completed and the observation made in this study shows the value of actual effort with effort estimation has a very strong correlation, however, the main advantage of the effort estimation is lower error variability and limited resources can be spent on effort estimation . It is based on the details :

   (a) Total actual effort on ten software goverment is 29280 hours.

   (b) Actual effort value of the samples, namely, the Assesment Library Website is 1010 hours.

   (c) Effort estimation at ten software development using use case points (UCP) have a value of 34716 hours.

   (d) Effort estimation value of the samples using use case points (UCP), namely, the Assesment Library Website is 1082 hours.

   (e) Effort estimation at ten software goverment using fuzzy use case points (FUCP) have a value of 27444 hours.

   (f) Effort estimation value of the samples using fuzzy use case points (FUCP), namely, the Assesment Library Website is 1027 hours.

2. In this study that applying fuzzy logic to this use case points in the UUCW category provide a significant improvement in the accuracy of effort estimation.

3. The implementation of FUCP to calculate effort estimation in ten projects goverment used in this research has shown that FUCP has the closest value to the actual effort. It is also demonstrated that FUCP outperforms UCP in terms of accuracy by 6,51 % improvements based on adjusted R square and low value of RMSE the closer to the value actual effort.

4. The correlation calculation effort estimation between UCP and FUCP method with actual effort is FUCP method gives good results for effort estimation in software development projects because FUCP has smaller RMSE is 547,1 than UCP has 699,4. A low value indicates that the effort estimation method approaches the accuracy of the actual value effort.

5.2 Suggestion

Some things are expected to be developed in the future to calculation effort estimation of software development is as follows :

1. Fuzzy use case points (FUCP) method still need to be tested on a software development project are more variation.

2. Activity diagram can be used to provide more effort estimation with better accuracy than UCP calculated based on sequence diagram and use case diagram

3. The accuracy of UAW and UUCW investigated can be used to fix the value of UCP.

4. If you want to know the variation increased accuracy of effort estimation method with the actual effort, not only used software development goverment based on project.

Bibliography

[Ali Bou Nasif, 2013] Ali Bou Nasif, Luiz Fernando Capretz, D. H. (2013). Towards an early software estimation using log-linear regression and a multilayer perceptron model.

[Allan Albrecht, 1983] Allan Albrecht, G. (1983). Software function and source lines of code and development effort prediction : A software science validation.

[Anda, 2002] Anda, B. (2002). Comparing effort estimates based on use cases with expert estimates.

[Barbosa da Silva, 2008] Barbosa da Silva, C.Monteiro., L. D. S. d. C. A. M. (2008). Applying the use case points effort estimation technique to avionics systems.

[Carroll, 2005] Carroll, E. (2005). Software based on use case points.

[Chai and Draxler, 2014] Chai, T. and Draxler, R. R. (2014). Root mean square error (rmse) or mean absolute error (mae)? â arguments against avoiding rmse in the literature.

[Evans, 2006] Evans, D. D. G. M. W. (2006). *Software Sizing, Estimation, and Risk Management*.

[G. Schneider, 1998] G. Schneider, J. W. (1998). *Applying Use Cases: A Practical Guide*.

[Kamal and Ahmed, 2011] Kamal, M. W. and Ahmed, M. A. (2011). A proposed framework for use case based effort estimation using fuzzy logic: Building upon the outcomes of a systematic literature review.

[Karner, 1993] Karner, G. (1993). Resource estimation for objectory projects. objective systems sf ab, rational software.

[Kumara, 2011] Kumara, R. (2011). Planned vs actual efforts.

[Kusumadewi, 2010] Kusumadewi, S. (2010). *Aplikasi logika fuzzy untuk pendukung keputusan*.

[Langer, 2008] Langer, A. M. (2008). *Analysis and Design of Information Systems*.

[Lee, 2016] Lee, M. (2016). Staruml 5.0 user guide.

[M Ochodek, 2010] M Ochodek, J Nawrocki, K. K. (2010). Simplifying effort estimation based on use case points.

[MathWorks, 2016] MathWorks, I. (2016). Matlab documentation.

[Mohammad Saber Iraji, 2012] Mohammad Saber Iraji, Aboutalebi, M. . M. H. (2012). Effort estimate with neuro fuzzy use case point based on exact weights.

[Monnappa, 2015] Monnappa, A. (2015). Project management series: Effort vs duration vs elapsed time.

[Muhardin, 2011] Muhardin, E. (2011). Estimasi proyek software.

[Penilaian, 2014] Penilaian, P. (2014).

[Renny Sari Dewi, 2015] Renny Sari Dewi, Apol Pribadi Subriadi, S. (2015). Use case point - activity-based costing: Metode baru untuk mengestimasi biaya pengembangan perangkat lunak.

[Rizky, 2008] Rizky, S. (2008). *Konsep Dasar Rekayasa Perangkat Lunak Software Reengineering*. Prestasi Pustaka.

[Tockey, 2005] Tockey, S. (2005). Return on software: Maximizing the return on your software investment.

[Wahyu Kurniawan, 2013] Wahyu Kurniawan, Sholiq, T. S. (2013). Penentuan effort rate pada estimasi effort menggunakan metode use case point untuk pengembangan perangkat lunak website kepemerintahan. Master's thesis.

[Wardana, 2015] Wardana, I. N. K. (2015).

Appendix A

Actual Effort All Project Software Development

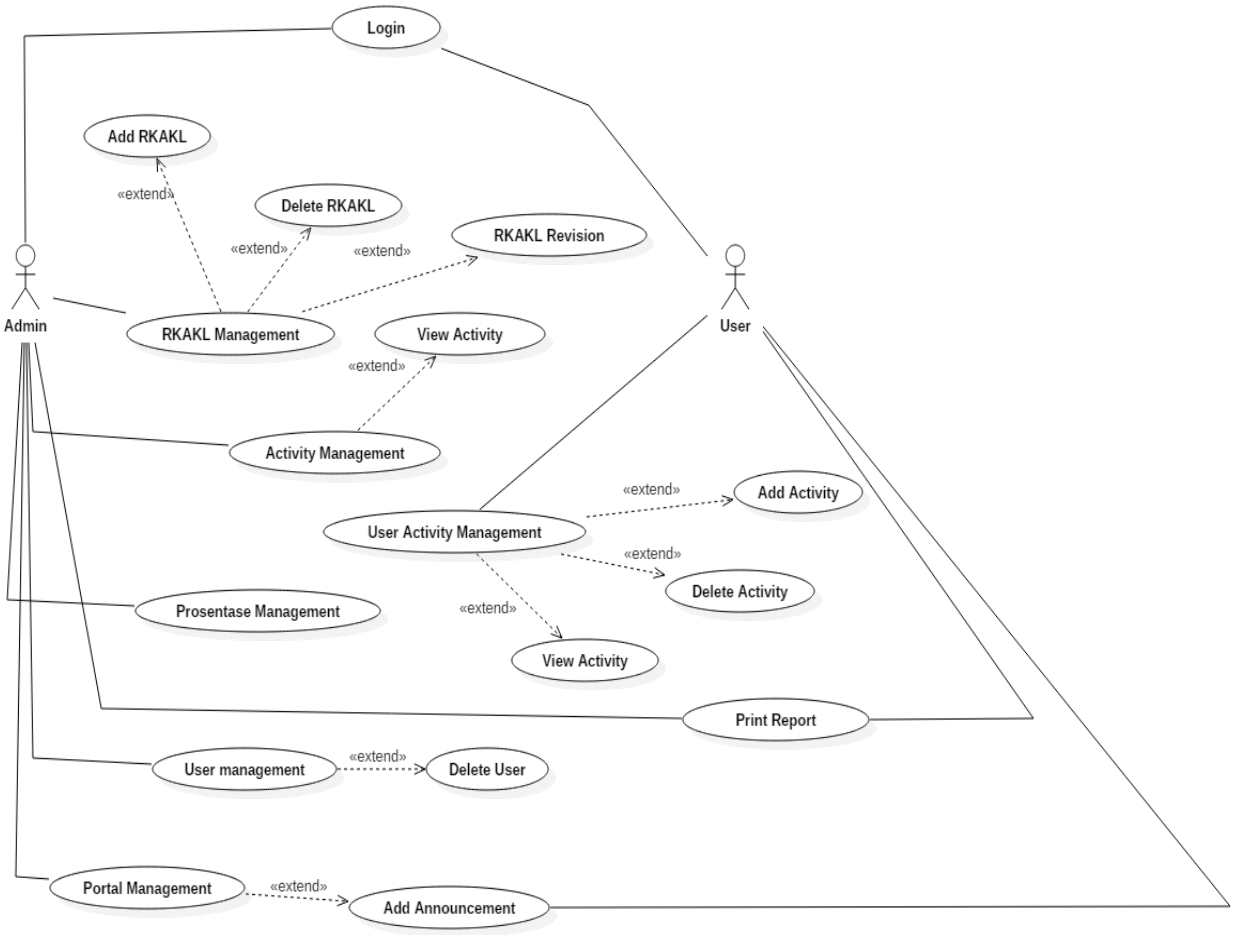
Project	Person	Time		Working Days	Working Hours	Actual Effort
		Start	End			
Evaluation Kemenpora Website (Website Evaluasi Kemenpora)	4	01/12/2015	24/03/2016	114	5	2280
Financal System of Dikti (Sistem Keuangan Dikti)	4	01/01/2016	01/05/2016	121	5	2420
SIMAYA System (Sistem SIMAYA)	4	01/09/2015	10/04/2016	222	5	4440
Inventory System of Pekalongan City (SIMSEDIA)	6	01/07/2015	31/12/2015	183	5	5490

Geographical Information System (GIS) Website of Kemenpora	2	01/11/2014	10/03/2015	129	5	1290
Sportscience Website of Kemenpora	4	27/02/2015	20/08/2015	174	5	3480
Assesment Library Website of Kemdikbud (Website Perpustakaan Penilaian Kemdikbud)	2	01/11/2015	10/02/2016	101	5	1010
Internasional Study Website of Kemdikbud (MINITES Kemdikbud)	2	02/07/2015	30/03/2016	272	5	2720
Biodiversity Mobile Apllication (Aplikasi Biodiversity)	4	30/12/2015	10/02/2016	42	5	840
BSN E-learning Website (Website E-learning BSN)	6	01/03/2015	25/08/2015	233	5	5310

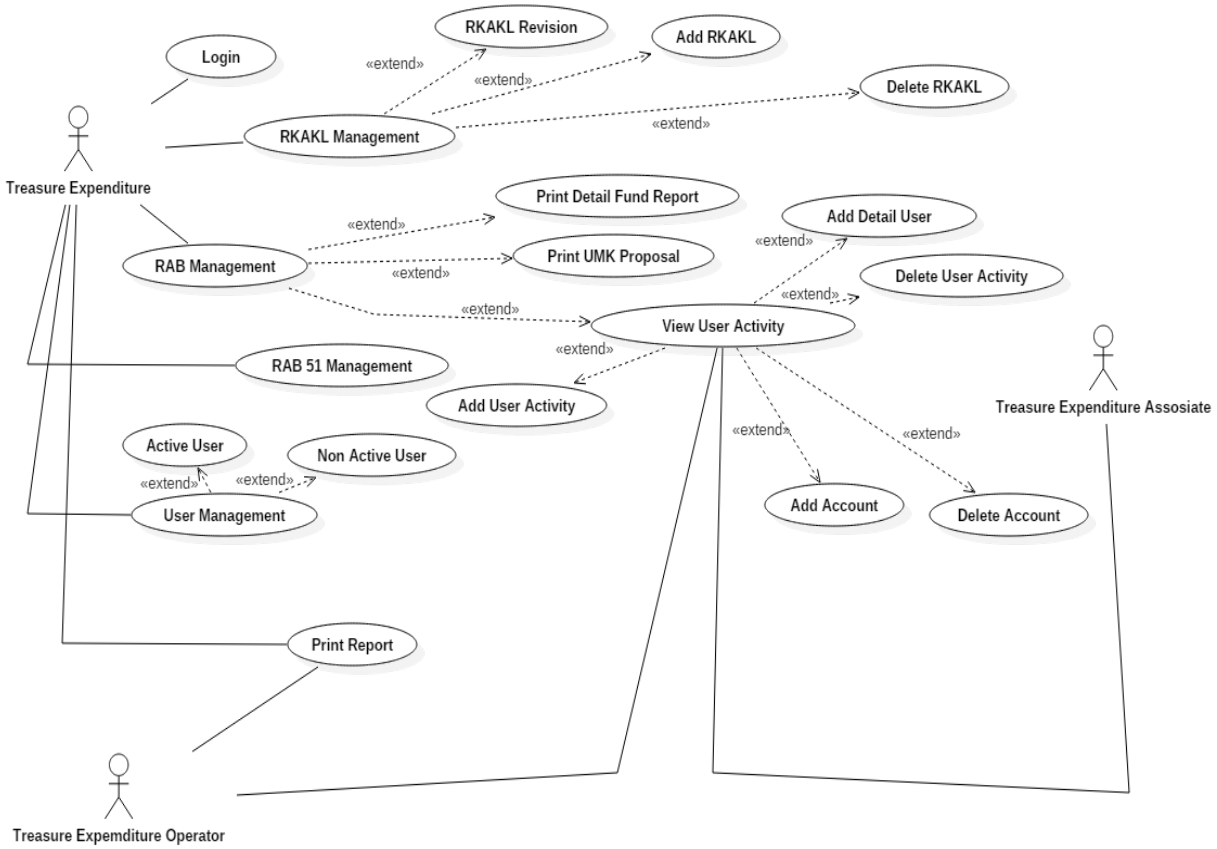
Appendix B

Use Case Diagram

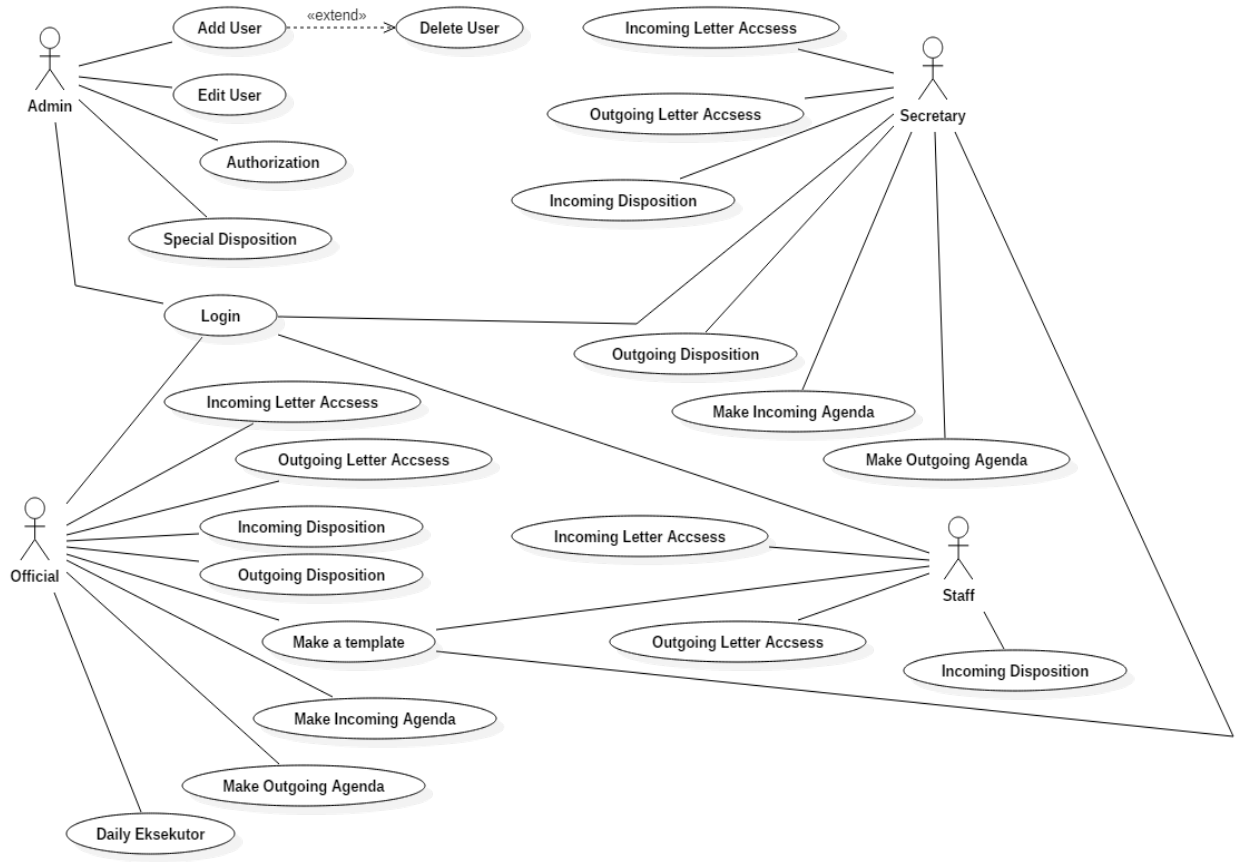
B.1 Use Case Diagram Evaluation Kemenpora Website (Website Evaluasi Kemenpora)



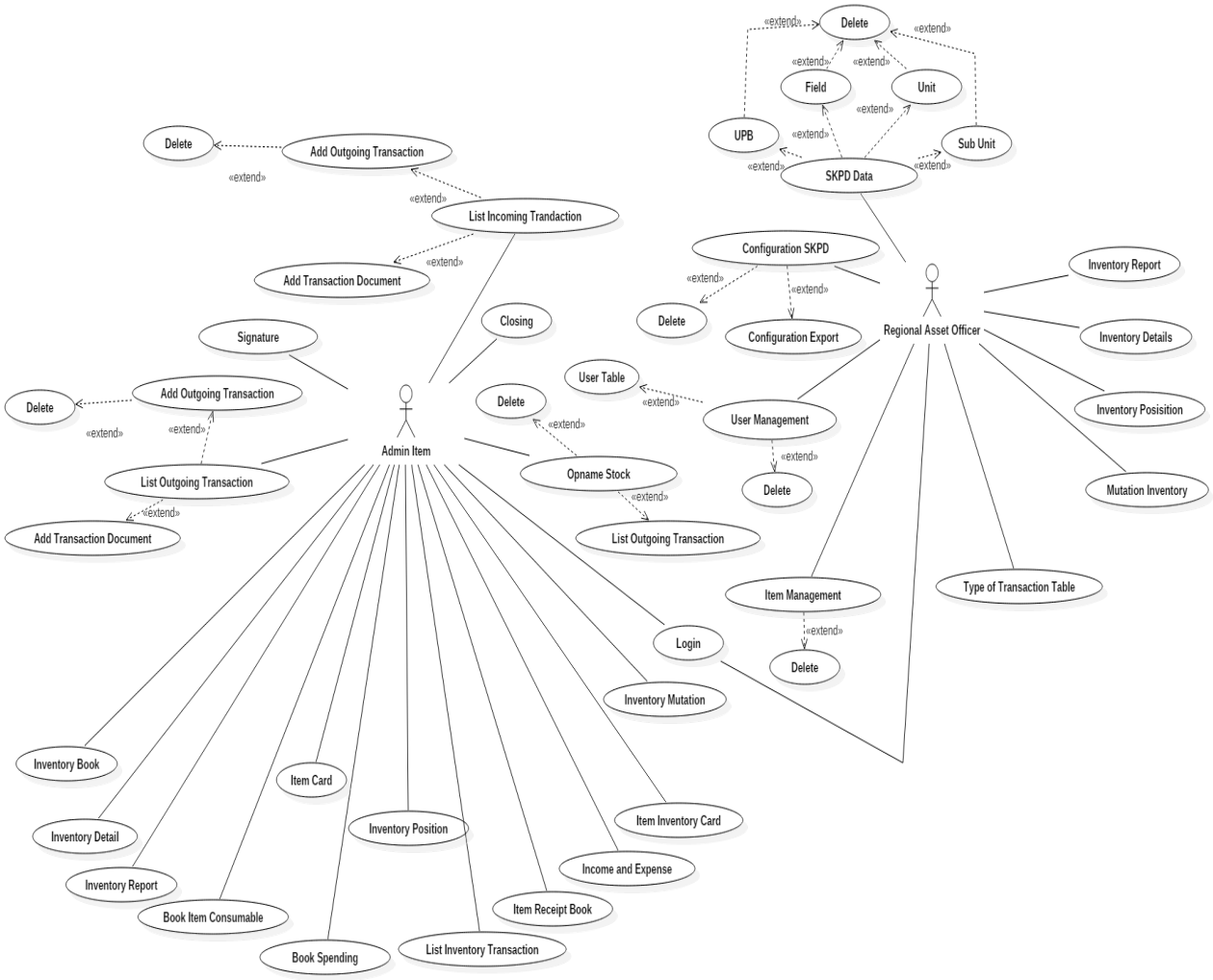
B.2 Use Case Diagram Financal System of Dikti (Sistem Keuangan Dikti)



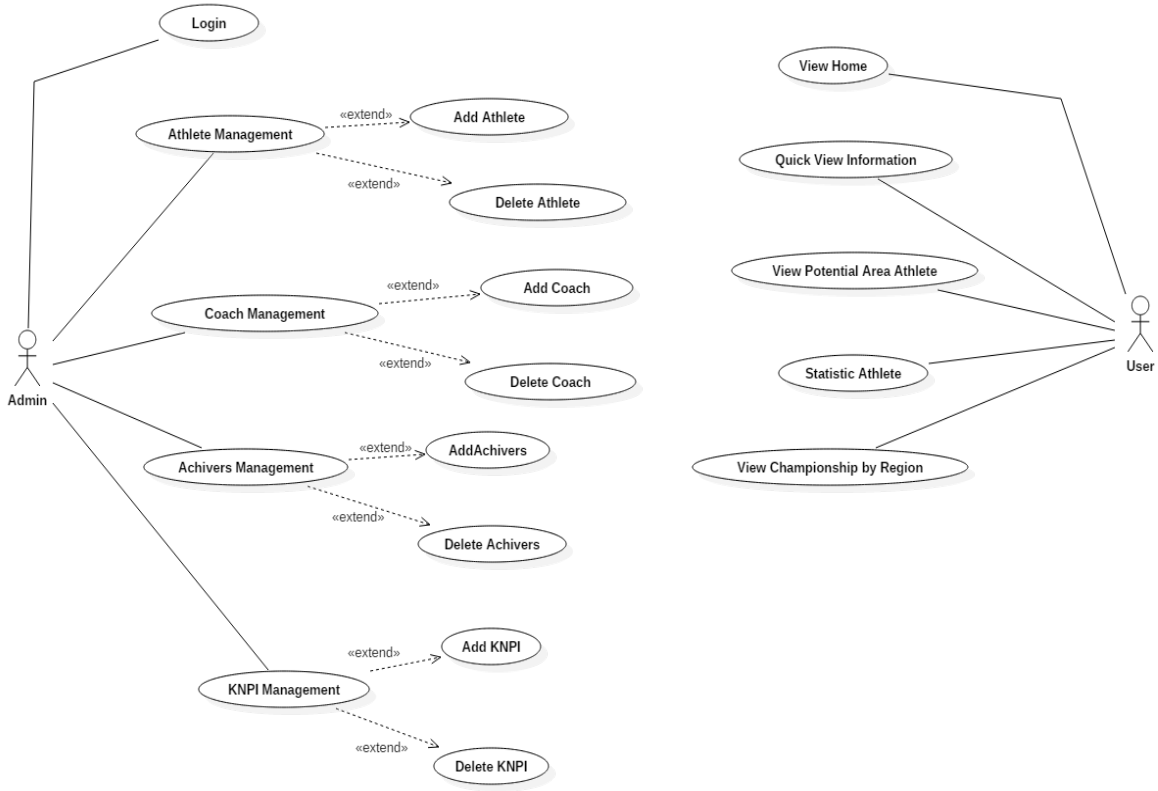
B.3 Use Case Diagram SIMAYA System (Sistem SIMAYA)



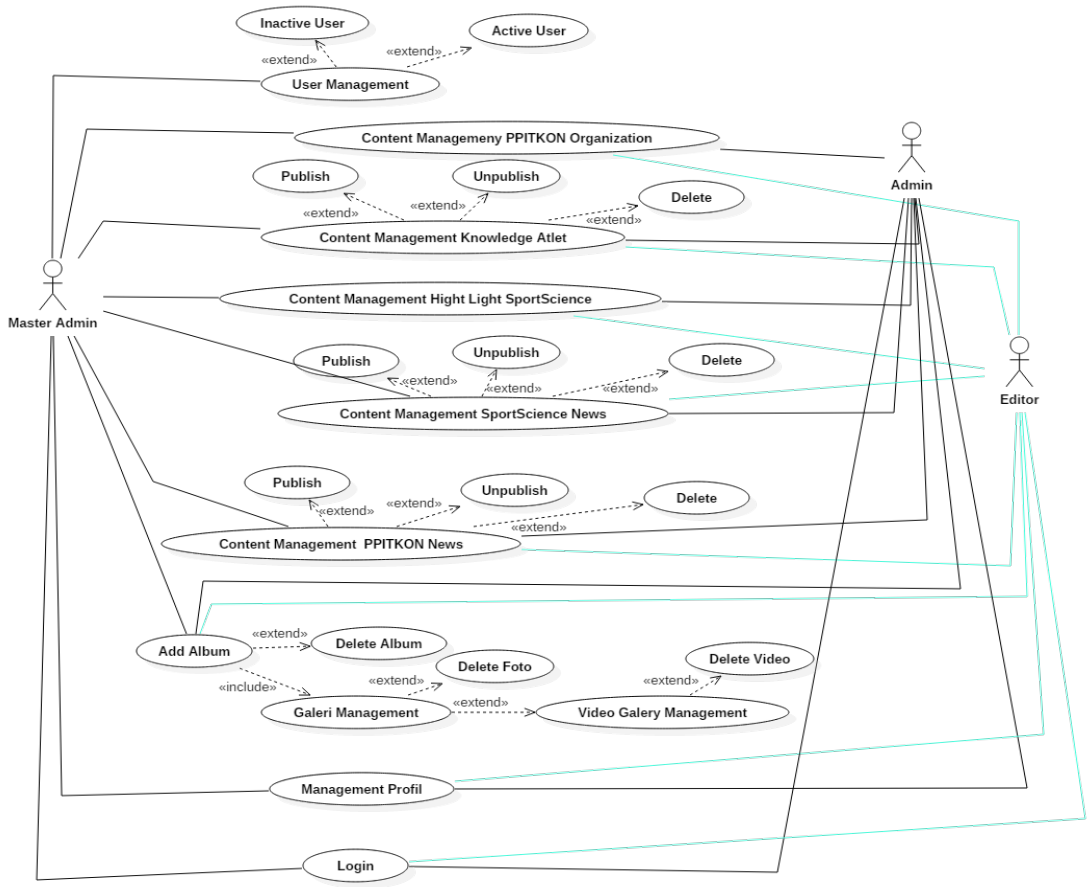
B.4 Use Case Diagram Inventory System of Pekalongan City (SIMSEDIA)

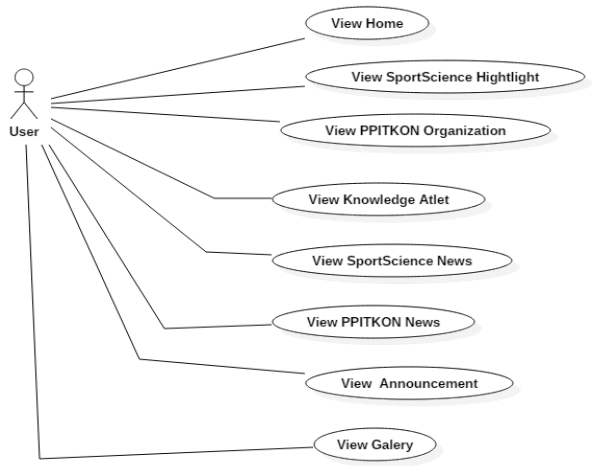


B.5 Use Case Diagram Geographical Information System (GIS) Website of Kemenpora

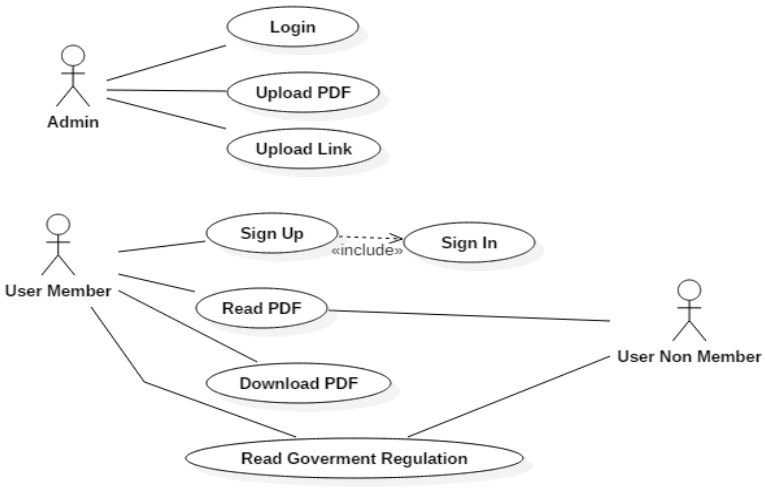


B.6 Use Case Diagram Sportscience Website of Kemenpora (Website Sportscience Kemenpora)

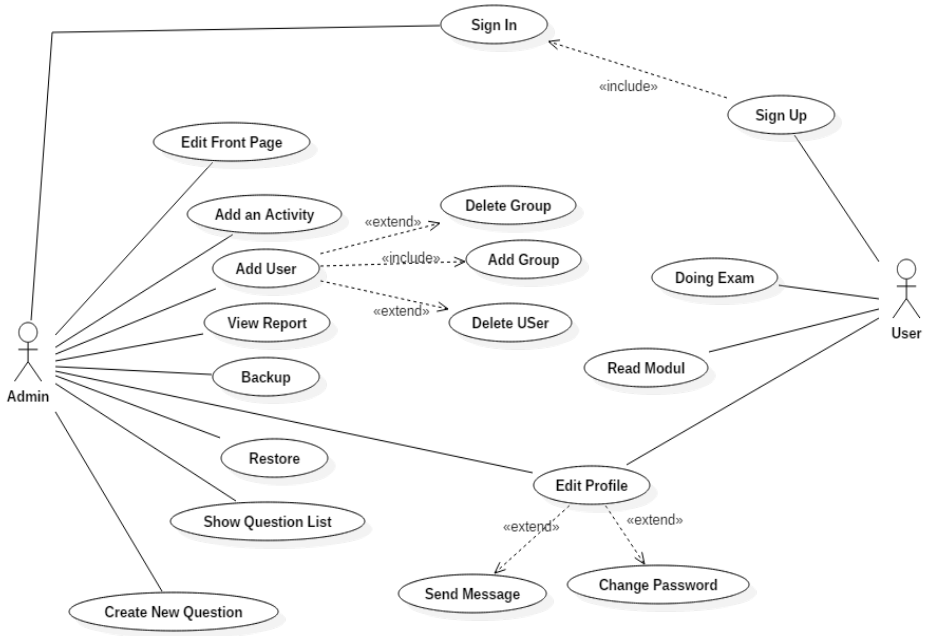




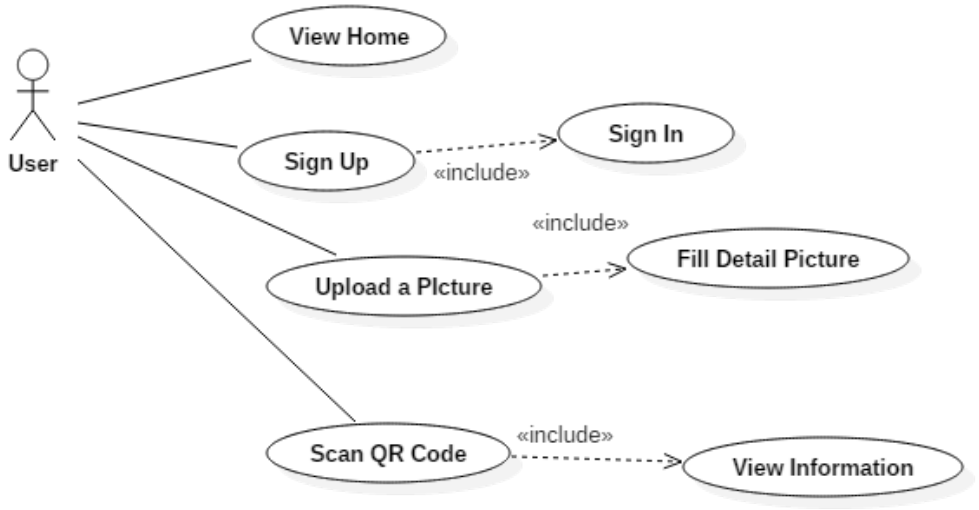
B.7 Use Case Diagram Assesment Library Website of Kemdikbud (Website Perpustakaan Penilaian Kemdikbud)



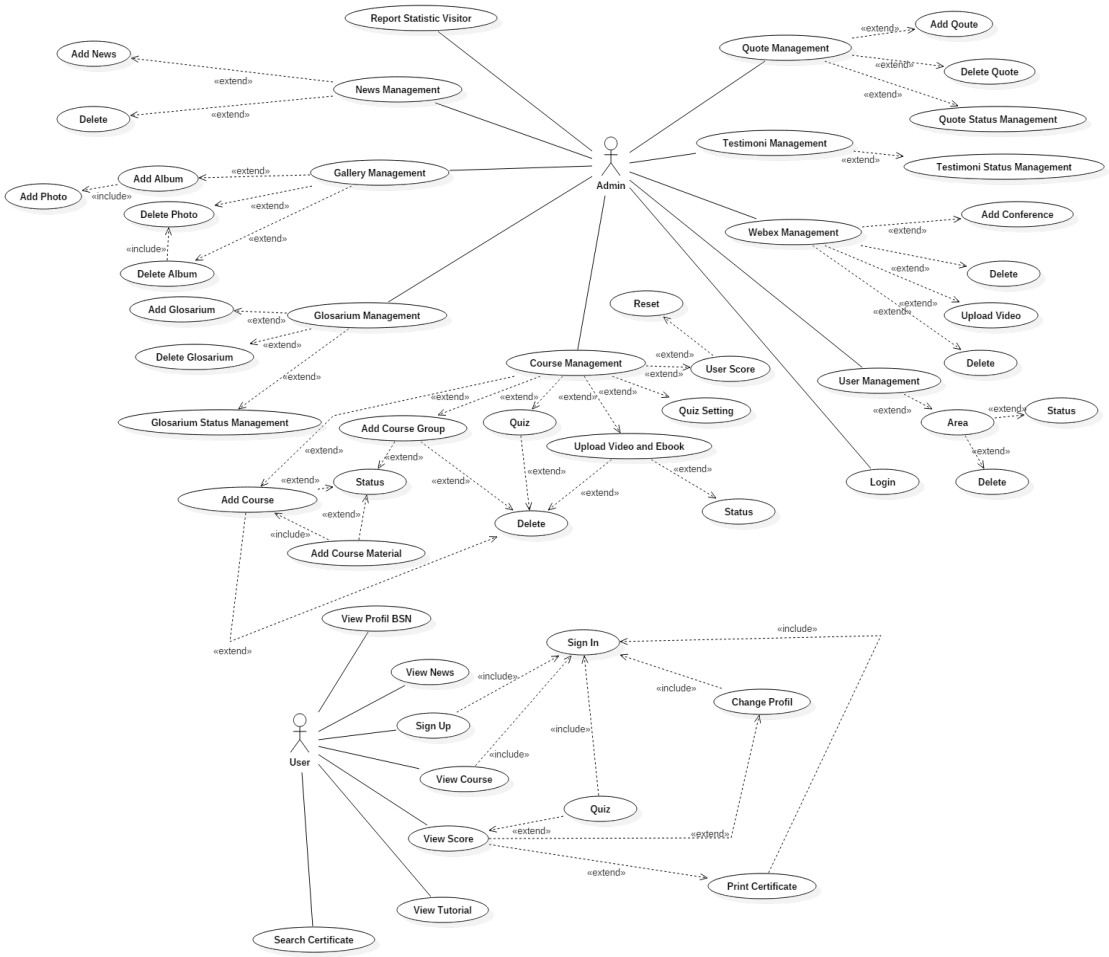
B.8 Use Case Diagram Internasional Study Website of Kemdikbud (MINITES Kemdikbud)



B.9 Use Case Diagram Biodiversity Mobile Aplli-cation (Aplikasi Biodiversity)



B.10 Use Case Diagram BSN E-learning Website (Website E-learning BSN)



Appendix C

Use Case Points (UCP)

C.1 UCP Evaluation Kemenpora Website (Website Evaluasi Kemenpora)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	2	6
Total				6

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	10	50
Average	4-7 Transaction	10	5	50
Complex	> 7 Transaction	15	2	30
Total				130

$$UUCP = UAW + UUCW$$
$$UUCP = 6 + 130$$
$$UUCP = 136$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10
T_2	Response time or through-put perfor-mance objectives	1,0	5	5
T_3	End-user online efficiency	1,0	4	4
T_4	Complex internal processing	1,0	5	5
T_5	Reusability of code	1,0	4	4
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	4	8
T_9	Ease of change	1,0	4	4
T_{10}	Concurrency	1,0	4	4
T_{11}	Special security objectives include	1,0	4	4

T_{12}	Direct access for thrid parties	1,0	4	4
T_{13}	Special user training required	1,0	5	5
Total TF				61
TCF				1,21

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	4	4
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	5	5
E_6	Requirements stability	2,0	4	8
E_7	Part time staff	-1,0	4	-4
E_8	Difficulty of programming language	-1,0	2	-2
Total EF				21
ECF				0,77

UCP = UUCP * TCF * ECF
UCP = 136,00 * 1,21 * 0,77
UCP = 126,72

C.2 UCP Financal System of Dikti (Sistem Keuangan Dikti)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	3	9
Total				9

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	11	55
Average	4-7 Transaction	10	7	70
Complex	> 7 Transaction	15	1	15
Total				140

$$\begin{aligned} \text{UUCP} &= \text{UAW} + \text{UUCW} \\ \text{UUCP} &= 9 + 140 \\ \text{UUCP} &= 149 \end{aligned}$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10

T_2	Response time or through-put perfor-mance objectives	1,0	4	4
T_3	End-user online efficiency	1,0	4	4
T_4	Complex internal processing	1,0	5	5
T_5	Reusability of code	1,0	4	4
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	4	8
T_9	Ease of change	1,0	4	4
T_{10}	Concurrency	1,0	4	4
T_{11}	Special security objectives include	1,0	4	4
T_{12}	Direct access for thrid parties	1,0	2	2
T_{13}	Special user training required	1,0	5	5
Total TF				58

TCF	1,18

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	4	4
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	5	5
E_6	Requirements stability	2,0	4	8
E_7	Part time staff	-1,0	4	-4
E_8	Difficulty of programming language	-1,0	2	-2
Total EF				21
ECF				0,77

UCP = UUCP * TCF * ECF
UCP = 149,00 * 1,18 * 0,77
UCP = 135,39

C.3 UCP SIMAYA System (Sistem SIMAYA)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	4	12
Total				12

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	12	60
Average	4-7 Transaction	10	5	50
Complex	> 7 Transaction	15	6	90
Total				200

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$
$$\text{UUCP} = 12 + 200$$
$$\text{UUCP} = 212$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10,00

T_2	Response time or through-put perfor-mance objectives	1,0	5	5,00
T_3	End-user online efficiency	1,0	4	4,00
T_4	Complex internal processing	1,0	4	4,00
T_5	Reusability of code	1,0	3	3,00
T_6	Easy to install	0,5	3	1,50
T_7	Ease of use	0,5	4	2,00
T_8	Portability	2,0	4	8,00
T_9	Ease of change	1,0	4	4,00
T_{10}	Concurrency	1,0	5	5,00
T_{11}	Special security objectives include	1,0	4	4,00
T_{12}	Direct access for thrid parties	1,0	3	3,00
T_{13}	Special user training required	1,0	5	5,00
Total TF				58,50

TCF	1,19

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	4	4
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	4	4
E_6	Requirements stability	2,0	5	10
E_7	Part time staff	-1,0	4	-4
E_8	Difficulty of programming language	-1,0	2	-2
Total EF				22
ECF				0,74

UCP = UUCP * TCF * ECF
UCP = 212,00 * 1,19 * 0,74
UCP = 186,69

C.4 UCP Inventory System of Pekalongan City (SIMSEDIA)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	2	6
Total				6

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	18	90
Average	4-7 Transaction	10	24	240
Complex	> 7 Transaction	15	3	45
Total				375

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$
$$\text{UUCP} = 6 + 375$$
$$\text{UUCP} = 381$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10

T_2	Response time or through-put perfor-mance objectives	1,0	4	4
T_3	End-user online efficiency	1,0	4	4
T_4	Complex internal processing	1,0	4	4
T_5	Reusability of code	1,0	3	3
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	4	8
T_9	Ease of change	1,0	3	3
T_{10}	Concurrency	1,0	4	4
T_{11}	Special security objectives include	1,0	3	3
T_{12}	Direct access for thrid parties	1,0	2	2
T_{13}	Special user training required	1,0	5	5
Total TF				54

TCF	1,14

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	4	4
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	5	5
E_6	Requirements stability	2,0	5	10
E_7	Part time staff	-1,0	5	-5
E_8	Difficulty of programming language	-1,0	1	-1
Total EF				23
ECF				0,71

UCP = UUCP * TCF * ECF
UCP = 381,00 * 1,14 * 0,71
UCP = 308,39

C.5 UCP Geographical Information System (GIS) Website of Kemenpora

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	2	6
Total				6

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	13	65
Average	4-7 Transaction	10	5	50
Complex	> 7 Transaction	15	0	0
Total				115

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$
$$\text{UUCP} = 6 + 115$$
$$\text{UUCP} = 121$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10,00

T_2	Response time or through-put perfor-mance objectives	1,0	5	5,00
T_3	End-user online efficiency	1,0	4	4,00
T_4	Complex internal processing	1,0	2	2,00
T_5	Reusability of code	1,0	4	4,00
T_6	Easy to install	0,5	5	2,50
T_7	Ease of use	0,5	5	2,50
T_8	Portability	2,0	5	10,00
T_9	Ease of change	1,0	5	5,00
T_{10}	Concurrency	1,0	5	5,00
T_{11}	Special security objectives include	1,0	4	4,00
T_{12}	Direct access for thrid parties	1,0	4	4,00
T_{13}	Special user training required	1,0	3	3,00
Total TF				61,00

TCF	1,21

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	5	7,50
E_2	Application experience	0,5	4	2,00
E_3	Object-oriented experience	1,0	5	5,00
E_4	Lead analyst capability	0,5	4	2,00
E_5	Motivation	1,0	4	4,00
E_6	Requirements stability	2,0	5	10,00
E_7	Part time staff	-1,0	5	-5,00
E_8	Difficulty of programming language	-1,0	1	-1,00
Total EF				24,50
ECF				0,67

UCP = UUCP * TCF * ECF
UCP = 121,00 * 1,21 * 0,67
UCP = 98,10

C.6 UCP Sportscience Website of Kemenpora (Website Sportscience Kemenpora)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	4	12
Total				12

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	25	125
Average	4-7 Transaction	10	8	80
Complex	> 7 Transaction	15	0	0
Total				205

$$UUCP = UAW + UUCW$$
$$UUCP = 12 + 205$$
$$UUCP = 217$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10

T_2	Response time or through-put perfor-mance objectives	1,0	4	4
T_3	End-user online efficiency	1,0	3	3
T_4	Complex internal processing	1,0	1	1
T_5	Reusability of code	1,0	4	4
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	5	10
T_9	Ease of change	1,0	4	4
T_{10}	Concurrency	1,0	5	5
T_{11}	Special security objectives include	1,0	4	4
T_{12}	Direct access for thrid parties	1,0	1	1
T_{13}	Special user training required	1,0	3	3
Total TF				53

TCF	1,13

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	3	3
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	4	4
E_6	Requirements stability	2,0	5	10
E_7	Part time staff	-1,0	4	-4
E_8	Difficulty of programming language	-1,0	2	-2
Total EF				21
ECF				0,77

UCP = UUCP * TCF * ECF
UCP = 217,00 * 1,13 * 0,77
UCP = 188,82

C.7 UCP Assesment Library Website of Kemdikbud (Website Perpustakaan Penilaian Kemdikbud)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	3	9
Total				9

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	5	25
Average	4-7 Transaction	10	2	20
Complex	> 7 Transaction	15	1	15
Total				60

$$UUCP = UAW + UUCW$$
$$UUCP = 9 + 60$$
$$UUCP = 69$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10,00

T_2	Response time or through-put perfor-mance objectives	1,0	3	3,00
T_3	End-user online efficiency	1,0	5	5,00
T_4	Complex internal processing	1,0	1	1,00
T_5	Reusability of code	1,0	5	5,00
T_6	Easy to install	0,5	5	2,50
T_7	Ease of use	0,5	5	2,50
T_8	Portability	2,0	5	10,00
T_9	Ease of change	1,0	5	5,00
T_{10}	Concurrency	1,0	5	5,00
T_{11}	Special security objectives include	1,0	4	4,00
T_{12}	Direct access for thrid parties	1,0	3	3,00
T_{13}	Special user training required	1,0	1	1,00
Total TF				57,00

TCF	1,17

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	5	7,50
E_2	Application experience	0,5	4	2,00
E_3	Object-oriented experience	1,0	5	5,00
E_4	Lead analyst capability	0,5	4	2,00
E_5	Motivation	1,0	4	4,00
E_6	Requirements stability	2,0	5	10,00
E_7	Part time staff	-1,0	5	-5,00
E_8	Difficulty of programming language	-1,0	1	-1,00
Total EF				24,50
ECF				0,67

UCP = UUCP * TCF * ECF
UCP = 69,00 * 1,17 * 0,67
UCP = 54,09

C.8 UCP Internasional Study Website of Kemdikbud (MINITES Kemdikbud)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	2	6
Total				6

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	8	40
Average	4-7 Transaction	10	3	30
Complex	> 7 Transaction	15	7	105
Total				175

$$UUCP = UAW + UUCW$$
$$UUCP = 6 + 175$$
$$UUCP = 181$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10

T_2	Response time or through-put perfor-mance objectives	1,0	5	5
T_3	End-user online efficiency	1,0	3	3
T_4	Complex internal processing	1,0	2	2
T_5	Reusability of code	1,0	4	4
T_6	Easy to install	0,5	2	1
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	5	10
T_9	Ease of change	1,0	4	4
T_{10}	Concurrency	1,0	4	4
T_{11}	Special security objectives include	1,0	3	3
T_{12}	Direct access for thrid parties	1,0	4	4
T_{13}	Special user training required	1,0	4	4
Total TF				56

TCF	1,16

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	5	7,50
E_2	Application experience	0,5	3	1,50
E_3	Object-oriented experience	1,0	4	4,00
E_4	Lead analyst capability	0,5	3	1,50
E_5	Motivation	1,0	3	3,00
E_6	Requirements stability	2,0	4	8,00
E_7	Part time staff	-1,0	4	-4,00
E_8	Difficulty of programming language	-1,0	3	-3,00
Total EF				18,50
ECF				0,84

UCP = UUCP * TCF * ECF
UCP = 181,00 * 1,16 * 0,84
UCP = 178,47

C.9 UCP Biodiversity Mobile Apllication (Aplikasi Biodiversity)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	1	3
Total				3

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	4	20
Average	4-7 Transaction	10	3	30
Complex	> 7 Transaction	15	0	0
Total				50

$$UUCP = UAW + UUCW$$
$$UUCP = 3 + 50$$
$$UUCP = 53$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	4	8,00

T_2	Response time or through-put perfor-mance objectives	1,0	4	4,00
T_3	End-user online efficiency	1,0	5	5,00
T_4	Complex internal processing	1,0	3	3,00
T_5	Reusability of code	1,0	4	4,00
T_6	Easy to install	0,5	4	2,00
T_7	Ease of use	0,5	5	2,50
T_8	Portability	2,0	2	4,00
T_9	Ease of change	1,0	5	5,00
T_{10}	Concurrency	1,0	5	5,00
T_{11}	Special security objectives include	1,0	3	3,00
T_{12}	Direct access for thrid parties	1,0	5	5,00
T_{13}	Special user training required	1,0	2	2,00
Total TF				52,50

TCF	1,13

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	5	7,50
E_2	Application experience	0,5	4	2,00
E_3	Object-oriented experience	1,0	4	4,00
E_4	Lead analyst capability	0,5	4	2,00
E_5	Motivation	1,0	4	4,00
E_6	Requirements stability	2,0	5	10,00
E_7	Part time staff	-1,0	4	-4,00
E_8	Difficulty of programming language	-1,0	2	-2,00
Total EF				23,50
ECF				0,70

UCP = UUCP * TCF * ECF
UCP = 53,00 * 1,13 * 0,77
UCP = 41,93

C.10 UCP BSN E-learning Website (Website E-learning BSN)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	2	6
Total				6

UUCW (Unadjusted Use Case Weight) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	1-3 Transaction	5	33	165
Average	4-7 Transaction	10	11	110
Complex	> 7 Transaction	15	8	120
Total				395

$$UUCP = UAW + UUCW$$
$$UUCP = 6 + 395$$
$$UUCP = 401$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	2	4

T_2	Response time or through-put perfor-mance objectives	1,0	2	2
T_3	End-user online efficiency	1,0	4	4
T_4	Complex internal processing	1,0	2	2
T_5	Reusability of code	1,0	3	3
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	4	8
T_9	Ease of change	1,0	4	4
T_{10}	Concurrency	1,0	4	4
T_{11}	Special security objectives include	1,0	3	3
T_{12}	Direct access for thrid parties	1,0	3	3
T_{13}	Special user training required	1,0	3	3
Total TF				44

TCF	1,04

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	3	4,50
E_2	Application experience	0,5	3	1,50
E_3	Object-oriented experience	1,0	3	3,00
E_4	Lead analyst capability	0,5	3	1,50
E_5	Motivation	1,0	3	3,00
E_6	Requirements stability	2,0	2	4,00
E_7	Part time staff	-1,0	2	-2,00
E_8	Difficulty of programming language	-1,0	2	-2,00
Total EF				13,50
ECF				1,00

UCP = UUCP * TCF * ECF
UCP = 401,00 * 1,04 * 1,00
UCP = 417,04

Appendix D

Fuzzy Use Case Points (FUCP)

D.1 FUCP Evaluation Kemenpora Website (Website Evaluasi Kemenpora)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	2	6
Total				6

FUUCW (Fuzzy Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Login	3	5,70	5,70
2	RKAKL Management	3	5,70	5,70
3	Add RKAKL	4	6,72	6,72

4	Delete RKAKL	1	4,79	4,79
5	RKAKL Revision	4	6,72	6,72
6	Activity Management	5	7,70	7,70
7	View Activity	3	5,70	5,70
8	Prosentase Management	3	5,70	5,70
9	Print Report	2	4,67	4,67
10	User Management	13	15,00	15,00
11	Delete User	1	4,79	4,79
12	Portal Management	4	6,72	6,72
13	Add Announcement	3	5,70	5,7 0
14	User Activity Management	4	6,72	6,72
15	Delete Activity	1	4,79	4,79
16	Add Activity	13	15,00	15,00
17	View Activity	3	5,70	5,70
FUUCW Value				117,82

$$\text{FUUCP} = \text{UAW} + \text{FUUCW}$$
$$\text{FUUCP} = 6{,}00 + 117{,}82$$
$$\text{FUUCP} = 123{,}82$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10
T_2	Response time or through-put perfor-mance objectives	1,0	5	5
T_3	End-user online efficiency	1,0	4	4

T_4	Complex internal processing	1,0	5	5
T_5	Reusability of code	1,0	4	4
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	4	8
T_9	Ease of change	1,0	4	4
T_{10}	Concurrency	1,0	4	4
T_{11}	Special security objectives include	1,0	4	4
T_{12}	Direct access for thrid parties	1,0	4	4
T_{13}	Special user training required	1,0	5	5
Total TF				61
TCF				1,21

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	4	4
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	5	5
E_6	Requirements stability	2,0	4	8
E_7	Part time staff	-1,0	4	-4
E_8	Difficulty of programming language	-1,0	2	-2
Total EF				21
ECF				0,77

FUCP = FUUCP * TCF * ECF
FUCP = 123,82 * 1,21 * 0,77
FUCP = 115,37

D.2 FUCP Financal System of Dikti (Sistem Keuangan Dikti)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	3	9
Total				9

FUUCW (Fuzzy Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Login	3	5,70	5,70
2	RKAKL Management	3	5,70	5,70
3	RKAKL Revision	4	6,72	6,72
4	Add RKAKL	4	6,72	6,72
5	Delete RKAKL	1	4,79	4,79
6	RAB Management	3	5,70	5,70
7	Print Detail Fund Report	1	4,79	4,79
8	Print UMK Proposal	1	4,79	4,79
9	RAB 51 Management	4	6,72	6,72
10	User Management	13	15,00	15,00
11	Non Active User	1	4,79	4,79
12	Active User	1	4,79	4,79
13	Print Report	4	6,72	6,72
14	View User Activity	2	4,67	4,67
15	Add User Activity	6	8,79	8,79
16	Add Account	4	6,72	6,72
17	Delete Account	1	4,79	4,79

18	Add Detail User	7	10,30	10,30
19	Delete User Activity	1	4,79	4,79
FUUCW Value				117,29

$$\text{FUUCP} = \text{UAW} + \text{FUUCW}$$
$$\text{FUUCP} = 9{,}00 + 117{,}29$$
$$\text{FUUCP} = 126{,}29$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10
T_2	Response time or through-put perfor-mance objectives	1,0	4	4
T_3	End-user online efficiency	1,0	4	4
T_4	Complex internal processing	1,0	5	5
T_5	Reusability of code	1,0	4	4
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	4	8
T_9	Ease of change	1,0	4	4

T_{10}	Concurrency	1,0	4	4
T_{11}	Special security objectives include	1,0	4	4
T_{12}	Direct access for thrid parties	1,0	2	2
T_{13}	Special user training required	1,0	5	5
Total TF				58
TCF				1,18

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	4	4
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	5	5
E_6	Requirements stability	2,0	4	8
E_7	Part time staff	-1,0	4	-4
E_8	Difficulty of programming language	-1,0	2	-2
Total EF				21
ECF				0,77

FUCP = FUUCP * TCF * ECF
FUCP = 126,29 * 1,18 * 0,77
FUCP = 114,75

D.3 FUCP SIMAYA System (Sistem SIMAYA)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	4	12
Total				12

FUUCW (Fuzzy Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Login	3	5,93	5,93
2	Add User	10	13,70	13,70
3	Edit User	5	7,70	7,70
4	Delete User	2	4,67	4,67
5	Authorization	4	6,72	6,72
6	Special Disposition	5	7,70	7,70
7	Incoming Letter Accsess	3	5,70	5,70
8	Outgoing Letter Accsess	3	5,70	5,70
9	Incoming Disposition	3	5,70	5,70
10	Outgoing Disposition	10	13,70	13,70
11	Make a Template	6	8,79	8,79
12	Make Incoming Agenda	10	13,70	13,70
13	Make Outgoing Agenda	12	15,00	15,00
14	Daily Eksekutor	3	5,70	5,70
15	Incoming Letter Accsess	3	5,70	5,70
16	Outgoing Letter Accsess	3	5,70	5,70
17	Incoming Disposition	3	5,70	5,70
18	Incoming Letter Accsess	3	5,70	5,70

19	Outgoing Letter Accsess	3	5,70	5,70
20	Incoming Disposition	3	5,70	5,70
21	Outgoing Disposition	10	13,70	13,70
22	Make Incoming Agenda	6	8,79	8,79
23	Make Outgoing Agenda	13,00	15,00	15,00
FUUCW Value				153,38

$$\text{FUUCP} = \text{UAW} + \text{FUUCW}$$
$$\text{FUUCP} = 12{,}00 + 153{,}38$$
$$\text{FUUCP} = 165{,}38$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10,00
T_2	Response time or through-put perfor-mance objectives	1,0	5	5,00
T_3	End-user online efficiency	1,0	4	4,00
T_4	Complex internal processing	1,0	4	4,00
T_5	Reusability of code	1,0	3	3,00
T_6	Easy to install	0,5	3	1,50
T_7	Ease of use	0,5	4	2,00

T_8	Portability	2,0	4	8,00
T_9	Ease of change	1,0	4	4,00
T_{10}	Concurrency	1,0	5	5,00
T_{11}	Special security objectives include	1,0	4	4,00
T_{12}	Direct access for thrid parties	1,0	3	3,00
T_{13}	Special user training required	1,0	5	5,00
Total TF				58,50
TCF				1,19

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	4	4
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	4	4
E_6	Requirements stability	2,0	5	10
E_7	Part time staff	-1,0	4	-4
E_8	Difficulty of programming language	-1,0	2	-2
Total EF				22
ECF				0,74

FUCP = FUUCP * TCF * ECF
FUCP = 165,38 * 1,19 * 0,74
FUCP = 145,64

D.4 FUCP Inventory System of Pekalongan City (SIMSEDIA)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	2	6
Total				6

FUUCW (Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Login	3	5,70	5,70
2	Signature	10	13,70	13,70
3	Closing	3	5,70	5,70
4	List Incoming Transaction	1	4,79	4,79
5	Add Incoming Transaction	5	7,70	7,70
6	Add Transaction Document	4	6,72	6,72
7	Delete	1	4,79	4,79
8	List Outgoing Transaction	1	4,79	4,79
9	Add Outgoing Transaction	5	7,70	7,70
10	Add Transaction Document	4	6,72	6,72
11	Delete	1	4,79	4,79
12	Add Opname Stock	6	8,79	8,79
13	List Outgoing Transaction	6	8,79	8,79
14	Delete	1	4,79	4,79
15	Inventory Mutation	6	8,79	8,79
16	Item Inventory Card	6	8,79	8,79
17	Income Expense	5	7,70	7,70

18	Inventory Book	6	8,79	8,79
19	Inventory Detail	5	7,70	7,70
20	Inventory Report	5	7,70	7,70
21	Book Item Consumable	5	7,70	7,70
22	Item Receipt Book	5	7,70	7,70
23	List Inventory Transaction	6	8,79	8,79
24	Inventory Position	5	7,70	7,70
25	Book Spending	5	7,70	7,70
26	Item Card	6	8,79	8,79
27	Inventory Report	6	8,79	8,79
28	Inventory Detail	9	12,50	12,50
29	Inventory Mutation	7	10,30	10,30
30	Type of Transaction Table	1	4,79	4,79
31	Inventory Position	6	8,79	8,79
32	Configuration SKPD	4	6,72	6,72
33	Export Configuration	4	6,72	6,72
34	Delete	2	4,67	4,67
35	User Management	8	11,6	11,6
36	User Table	1	4,79	4,79
37	Delete	1	4,79	4,79
38	Item Management	7	10,3 0	10,3 0
39	Delete	1	4,79	4,79
40	SKPD Data	1	4,79	4,79
41	UPB	2	4,67	4,67
42	Unit	2	4,67	4,67
43	Sub Unit	2	4,67	4,67
44	Field	2	4,67	4,67
45	Delete	1	4,79	4,79
FFUCW Value				321,13
				321,13

$$\text{FUUCP} = \text{UAW} + \text{FUUCW}$$
$$\text{FUUCP} = 6{,}00 + 321{,}13$$
$$\text{FUUCP} = 327{,}13$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10
T_2	Response time or through-put perfor-mance objectives	1,0	4	4
T_3	End-user online efficiency	1,0	4	4
T_4	Complex internal processing	1,0	4	4
T_5	Reusability of code	1,0	3	3
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	4	8
T_9	Ease of change	1,0	3	3
T_{10}	Concurrency	1,0	4	4
T_{11}	Special security objectives include	1,0	3	3
T_{12}	Direct access for thrid parties	1,0	2	2

T_{13}	Special user training required	1,0	5	5
Total TF				54
TCF				1,14

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	4	4
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	5	5
E_6	Requirements stability	2,0	5	10
E_7	Part time staff	-1,0	5	-5
E_8	Difficulty of programming language	-1,0	1	-1
Total EF				23
ECF				0,71

FUCP = FUUCP * TCF * ECF
FUCP = 327,13 * 1,14 * 0,71
FUCP = 264,78

D.5 FUCP Geographical Information System (GIS) Website of Kemenpora

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	13	65
Average	External System using a protocol based interface	2	5	50
Complex	Human	3	0	0
Total				115

FUUCW (Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Login	3	5,70	5,70
2	Athlete Management	1	4,79	4,79
3	Add Athlete	6	8,79	8,79
4	Delete Athlete	3	5,70	5,70
5	Coach Management	1	4,79	4,79
6	Add Coach	5	7,7 0	7,70
7	Delete Coach	3	5,70	5,70
8	Achivers Management	1	4,79	4,79
9	Add Achivers	5	7,70	7,70
10	Delete Achivers	3	5,70	5,70
11	KNPI Management	1	4,79	4,79
12	Add KNPI	7	10,30	10,30
13	Delete KNPI	3	5,70	5,70
FUUCW Value				82,15

FUUCP = UAW + FUUCW
FUUCP = 6,00 + 82,15
FUUCP = 88,15

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10,00
T_2	Response time or through-put perfor-mance objectives	1,0	5	5,00
T_3	End-user online efficiency	1,0	4	4,00
T_4	Complex internal processing	1,0	2	2,00
T_5	Reusability of code	1,0	4	4,00
T_6	Easy to install	0,5	5	2,50
T_7	Ease of use	0,5	5	2,50
T_8	Portability	2,0	5	10,00
T_9	Ease of change	1,0	5	5,00
T_{10}	Concurrency	1,0	5	5,00
T_{11}	Special security objectives include	1,0	4	4,00

T_{12}	Direct access for thrid parties	1,0	4	4,00
T_{13}	Special user training required	1,0	3	3,00
Total TF				61,00
TCF				1,21

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	5	7,5
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	5	5
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	4	4
E_6	Requirements stability	2,0	5	10
E_7	Part time staff	-1,0	5	-5
E_8	Difficulty of programming language	-1,0	1	-1
Total EF				24,50
ECF				0,67

FUCP = FUUCP * TCF * ECF
FUCP = 121,00 * 1,21 * 0,67
FUCP = 71,47

D.6 FUCP Sportscience Website of Kemenpora (Website Sportscience Kemenpora)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	4	12
Total				12

FUUCW (Fuzzy Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Login	3	5,70	5,70
2	User Management	6	8,79	8,79
3	Inactive User	1	4,79	4,79
4	Active User	1	4,79	4,79
5	Content Management PPITKON Organization	4	6,72	6,72
6	Content Management Knowledge Atlet	7	10,30	10,30
7	Publish	1	4,79	4,79
8	Unpublish	1	4,79	4,79
9	Delete	1	4,79	4,79
10	Content Management Hight Light SportScience	2	4,67	4,67
11	Content Management SportScience News	6	8,79	8,79

12	Publish	1	4,79	4,79
13	Unpublish	1	4,79	4,79
14	Delete	1	4,79	4,79
15	Content Management PPITKON News	6	8,79	8,79
16	Publish	1	4,79	4,79
17	Unpublish	1	4,79	4,79
18	Delete	1	4,79	4,79
19	Galery Management	4	6,72	6,72
20	Delete	1	4,79	4,79
21	Add Album	4	6,72	6,72
22	Delete Album	1	4,79	4,79
23	Video Galery Management	6	8,79	8,79
24	Delete	1	4,79	4,79
25	Profil Management	3	5,70	5,70
26	View Home	1	4,79	4,79
27	View SportScience Hight Light	1	4,79	4,79
28	View PPITKON Organization	1	4,79	4,79
29	View Knowledge Atlet	2	4,67	4,67
30	View SportScience News	2	4,67	4,67
31	View PPITKON News	2	4,67	4,67
32	View Announcement	2	4,67	4,67
33	FUUCW Value	2	4,67	4,67
FUUCW Value				180,77

$$FUUCP = UAW + FUUCW$$
$$FUUCP = 12{,}00 + 180{,}77$$
$$FUUCP = 192{,}77$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10
T_2	Response time or through-put perfor-mance objectives	1,0	4	4
T_3	End-user online efficiency	1,0	3	3
T_4	Complex internal processing	1,0	1	1
T_5	Reusability of code	1,0	4	4
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	5	10
T_9	Ease of change	1,0	4	4
T_{10}	Concurrency	1,0	5	5
T_{11}	Special security objectives include	1,0	4	4
T_{12}	Direct access for thrid parties	1,0	1	1

T_{13}	Special user training required	1,0	3	3
Total TF				53
TCF				1,13

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	4	6
E_2	Application experience	0,5	4	2
E_3	Object-oriented experience	1,0	3	3
E_4	Lead analyst capability	0,5	4	2
E_5	Motivation	1,0	4	4
E_6	Requirements stability	2,0	5	10
E_7	Part time staff	-1,0	4	-4
E_8	Difficulty of programming language	-1,0	2	-2
Total EF				21
ECF				0,77

FUCP = FUUCP * TCF * ECF
FUCP = 192,77 * 1,13 * 0,77
FUCP = 167,73

D.7 FUCP Assesment Library Website of Kemdikbud (Website Perpustakaan Penilaian Kemdikbud)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	3	9
Total				9

FUUCW (Fuzzy Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Login	3	5,70	5,70
2	Upload PDF	7	10,30	10,30
3	Upload Link	4	6,72	6,72
4	Sign Up	10	13,70	13,70
5	Sign In	3	5,70	5,70
6	Read PDF	1	4,79	4,79
7	Download PDF	1	4,79	4,79
8	Read Goverment Regulation	1	4,79	4,79
FUUCW Value				56,49

$$\text{FUUCP} = \text{UAW} + \text{FUUCW}$$
$$\text{FUUCP} = 9{,}00 + 56{,}49$$
$$\text{FUUCP} = 65{,}49$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10,00
T_2	Response time or through-put perfor-mance objectives	1,0	3	3,00
T_3	End-user online efficiency	1,0	5	5,00
T_4	Complex internal processing	1,0	1	1,00
T_5	Reusability of code	1,0	5	5,00
T_6	Easy to install	0,5	5	2,50
T_7	Ease of use	0,5	5	2,50
T_8	Portability	2,0	5	10,00
T_9	Ease of change	1,0	5	5,00
T_{10}	Concurrency	1,0	5	5,00
T_{11}	Special security objectives include	1,0	4	4,00
T_{12}	Direct access for thrid parties	1,0	3	3,00

T_{13}	Special user training required	1,0	1	1,00
Total TF				57,00
TCF				1,17

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	5	7,50
E_2	Application experience	0,5	4	2,00
E_3	Object-oriented experience	1,0	5	5,00
E_4	Lead analyst capability	0,5	4	2,00
E_5	Motivation	1,0	4	4,00
E_6	Requirements stability	2,0	5	10,00
E_7	Part time staff	-1,0	5	-5,00
E_8	Difficulty of programming language	-1,0	1	-1,00
Total EF				24,50
ECF				0,67

FUCP = FUUCP * TCF * ECF

FUCP = 65,49 * 1,17 * 0,67

FUCP = 51,34

D.8 FUCP Internasional Study Website of Kemdikbud (MINITES Kemdikbud)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	2	6
Total				6

FUUCW (Fuzzy Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Sign In	3	5,70	5,70
2	Sign Up	11	15 ,00	15,00
3	Edit Front Page	13	15,00	15,00
4	Add an Activity	12	15,00	15,00
5	Add User	3	5,70	5,70
6	Delete Group	2	4,67	4,67
7	Add Group	8	11,60	11,60
8	Delete User	2	4,67	4,67
9	View Report	7	10,30	10,30
10	Backup	11	15,00	15,00
11	Edit Profile	10	13,70	13,70
12	Send Message	4	6,72	6,72
13	Change Password	6	8,79	8,79
14	Restore	3	5,70	5,70
15	Show Question List	3	5,70	5,70
16	Create New Question	4	6,72	6,72
17	Doing Exam	3	5,70	5,70

18	Read Modul	3	5,70	5,70
FUUCW Value				155,67

FUUCP = UAW + FUUCW
FUUCP = 6,00 + 155,67
FUUCP = 161,67

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	5	10
T_2	Response time or through-put perfor-mance objectives	1,0	5	5
T_3	End-user online efficiency	1,0	3	3
T_4	Complex internal processing	1,0	2	2
T_5	Reusability of code	1,0	4	4
T_6	Easy to install	0,5	2	1
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	5	10
T_9	Ease of change	1,0	4	4
T_{10}	Concurrency	1,0	4	4

T_{11}	Special security objectives include	1,0	3	3
T_{12}	Direct access for thrid parties	1,0	4	4
T_{13}	Special user training required	1,0	4	4
Total TF				56
TCF				1,16

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	5	7,50
E_2	Application experience	0,5	3	1,50
E_3	Object-oriented experience	1,0	4	4,00
E_4	Lead analyst capability	0,5	3	1,50
E_5	Motivation	1,0	3	3,00
E_6	Requirements stability	2,0	4	8,00
E_7	Part time staff	-1,0	4	-4,00
E_8	Difficulty of programming language	-1,0	3	-3,00
Total EF				18,50
ECF				0,84

FUCP = FUUCP * TCF * ECF
FUCP = 161,67 * 1,16 * 0,85
FUCP = 159,41

D.9 FUCP Biodiversity Mobile Apllication (Aplikasi Biodiversity)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	1	3
Total				

FUUCW (Fuzzy Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	View Home	1	4,79	4,79
2	Sign Up	6	8,79	8,79
3	Sign In	3	5,70	5,70
4	Upload a Picture	1	4,79	4,79
5	Scan QR Code	5	7,70	7,70
6	View Information	1	4,79	4,79
FUUCW Value				36,56

$$\text{FUUCP} = \text{UAW} + \text{FUUCW}$$
$$\text{FUUCP} = 3{,}00 + 36{,}56$$
$$\text{FUUCP} = 39{,}56$$

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value

T_1	Distributed System	2,0	4	8,00
T_2	Response time or through-put perfor-mance objectives	1,0	4	4,00
T_3	End-user online efficiency	1,0	5	5,00
T_4	Complex internal processing	1,0	3	3,00
T_5	Reusability of code	1,0	4	4,00
T_6	Easy to install	0,5	4	2,00
T_7	Ease of use	0,5	5	2,50
T_8	Portability	2,0	2	4,00
T_9	Ease of change	1,0	5	5,00
T_{10}	Concurrency	1,0	5	5,00
T_{11}	Special security objectives include	1,0	3	3,00
T_{12}	Direct access for thrid parties	1,0	5	5,00

T_{13}	Special user training required	1,0	2	2,00
Total TF				52,50
TCF				1,13

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	5	7,50
E_2	Application experience	0,5	4	2,00
E_3	Object-oriented experience	1,0	4	4,00
E_4	Lead analyst capability	0,5	4	2,00
E_5	Motivation	1,0	4	4,00
E_6	Requirements stability	2,0	5	10,00
E_7	Part time staff	-1,0	4	-4,00
E_8	Difficulty of programming language	-1,0	2	-2,00
Total EF				23,50
ECF				0,70

FUCP = FUUCP * TCF * ECF
FUCP = 39,56 * 1,13 * 0,70
FUCP = 31,30

D.10 FUCP BSN E-learning Website (Website E-learning BSN)

UAW (Unadjusted Actor Weigth) Value

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	0	0
Average	External System using a protocol based interface	2	0	0
Complex	Human	3	2	6
Total				6

FUUCW (Fuzzy Unadjusted Use Case Weight) Value

No	Use Case	Type of Transactions	Weighting Factor	Value
1	Login	3	5,70	5,70
2	User Management	3	5,70	5,70
3	Area	5	7,70	7,70
4	Status	1	4,79	4,79
5	Delete	2	4,67	4,67
6	Webex Management	1	4,79	4,79
7	Add Conference	10	13,70	13,70
8	Delete	1	4,79	4,79
9	Upload Video	6	8,79	8,79
10	Delete	1	4,79	4,79
11	Testimoni Management	1	4,79	4,79
12	Testimoni Status Management	1	4,79	4,79
13	Quote Management	1	4,79	4,79
14	Add Quote	5	7,70	7,70
15	Delete Quote	2	4,67	4,67
16	Quote Status Management	1	4,79	4,79
17	Report Statistic Visitor	4	6,72	6,72

18	News Management	1	4,79	4,79
19	Add News	8	11,60	11,60
20	News Management	1	4,79	4,79
21	Gallery Management	1	4,79	4,79
22	Add Album	5	7,70	7,70
23	Add Photo	7	10,30	10,30
24	Delete Album	1	4,79	4,79
25	Delete Photo	3	5,70	5,70
26	Glosarium Management	1	4,79	4,79
27	Add Glosarium	5	7,70	7,70
28	Delete Glosarium	2	4,67	4,67
29	Glosarium Status Management	1	4,79	4,79
30	Course Management	1	4,79	4,79
31	Add Course	12	15,00	15,00
32	Add Course Group	4	6,72	6,72
33	Status	1	4,79	4,79
34	Add Course Material	7	10,30	10,30
35	Delete	2	4,67	4,67
36	Quiz	13	15,00	15,00
37	Upload Video adn Ebook	10	13,70	13,70
38	Status	1	4,79	4,79
39	Quiz Setting	8	11,60	11,60
40	User Score	1	4,79	4,79
41	Reset	2	4,67	4,67
42	View Tutorial	1	4,79	4,79
43	Search Certificate	4	6,72	6,72
44	View Profil BSN	1	4,79	4,79
45	View News	2	4,67	4,67
46	Sign Up	14	15,00	15,00
47	Sign In	4	6,72	6,72
48	View Course	3	5,70	5,70
49	Quiz	3	5,70	5,70
50	Change Profil	13	15,00	15,00
51	Print Certificate	2	4,67	4,67
52	View Score	1	4,79	4,79
FUUCW Value				234,6

FUUCP = UAW + FUUCW
FUUCP = 6,00 + 234,60
FUUCP = 240,60

TCF (Technical Complexity Factor)

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T_1	Distributed System	2,0	2	4
T_2	Response time or through-put perfor-mance objectives	1,0	2	2
T_3	End-user online efficiency	1,0	4	4
T_4	Complex internal processing	1,0	2	2
T_5	Reusability of code	1,0	3	3
T_6	Easy to install	0,5	4	2
T_7	Ease of use	0,5	4	2
T_8	Portability	2,0	4	8
T_9	Ease of change	1,0	4	4
T_{10}	Concurrency	1,0	4	4
T_{11}	Special security objectives include	1,0	3	3

T_{12}	Direct access for thrid parties	1,0	3	3
T_{13}	Special user training required	1,0	3	3
Total TF				44
TCF				1,04

ECF (Environment Complexity Factor)

No	Description	Weight	Assigned Value (0-5)	Weighted Value
E_1	Familiarity with system development process being used	1,5	3	4,50
E_2	Application experience	0,5	3	1,50
E_3	Object-oriented experience	1,0	3	3,00
E_4	Lead analyst capability	0,5	3	1,50
E_5	Motivation	1,0	3	3,00
E_6	Requirements stability	2,0	2	4,00
E_7	Part time staff	-1,0	2	-2,00
E_8	Difficulty of programming language	-1,0	2	-2,00
Total EF				13,50
ECF				1,00

FUCP = FUUCP * TCF * ECF
FUCP = 240,60 * 1,04 * 1,00
FUCP = 250,23

Appendix E

Question of TCF

1. T1. In the Assesment Library Website, whether the distributed system is important?

No Important	1	2	3	4	5	Important

2. T2. Response time or throughput performance objectives system is important?

No Important	1	2	3	4	5	Important

3. T3. In the Assesment Library Website end-user online efficiency?

Strongly disagree	Disagree	Ordinary	Agreee	Strongly agree

4. T4. In the Assesment Library Website complex internal processing is important?

No Important	1	2	3	4	5	Important

5. T5. Reusability of code in the Assesment Library Website?

Strongly disagree	Disagree	Ordinary	Agreee	Strongly agree

6. T6. Whether Assesment Library Website easy to install?

Strongly disagree	Disagree	Ordinary	Agreee	Strongly agree

7. T7. Whether Assesment Library Website ease of use?

Strongly disagree	Disagree	Ordinary	Agreee	Strongly agree

8. T8. Whether Assesment Library Website designed to be installed on variety platforms?

Strongly disagree	Disagree	Ordinary	Agreee	Strongly agree

9. T9. Whether Assesment Library Website designed to facilitate their changes (eg changes in the data or changing requirements, etc.)?

Strongly disagree	Disagree	Ordinary	Agreee	Strongly agree

10. T10. Concurrency in Assesment Library Website?

Strongly disagree	Disagree	Ordinary	Agreee	Strongly agree

11. T11. Whether Assesment Library Website designed to Special security objectives include?

Strongly disagree	Disagree	Ordinary	Agreee	Strongly agree

12. T12. Whether Assesment Library Website is made, the website will be very dependent on the system library or plugin or emmbed code or API?

Dependent	1	2	3	4	5	Independent

13. T13. In the Assesment Library Website Special user training required is important ?

No Important	1	2	3	4	5	Important

Appendix F

Question of ECF

1. E1 Familiarity with system development process being used?

No Familiar	1	2	3	4	5	Very Familiar

2. E2 Already familiar make on a project system or application system?

No Familiar	1	2	3	4	5	Very Familiar

3. E3 Already familiar with the system or applications that are object-oriented?

No Familiar	1	2	3	4	5	Very Familiar

4. E4 Having the ability to analyze very well?

Bad	1	2	3	4	5	Very good

5. E5 Working with a highly motivated team, optimistic, ambitious?

Strongly disagree	Disagree	Ordinary	Agreee	Strongly agree

6. E6 Already familiar adapt to many changes requirements of each project systems or applications?

No Familiar	1	2	3	4	5	Very Familiar

7. E7 Worked part-time for the project simple system or complex?

No Familiar	1	2	3	4	5	Very Familiar

8. E8 In working on the project, is not affected by any programming language from easy to difficult.

No affected	1	2	3	4	5	Very affected