A SOFTWARE COST ESTIMATION MODEL FOR GOVERNMENT PROJECTS

RIANTI ROZALINA

THESIS SUBMITTED IN FULFILMENT FOR THE DEGREE OF
MASTER OF INFORMATION SYSTEM

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2018

MODEL ANGGARAN KOS PERISIAN UNTUK PROJEK KERAJAAN

RIANTI ROZALINA

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEHI
IJAZAH SARJANA SISTEM MATLUMAT

FAKULTI TEKNOLOGI SAINS DAN MATLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2018

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

28 August 2018

RIANTI ROZALINA
P85964

ACKNOWLEDGMENT

Praise be to Almighty Allah for all his blessings - for giving me strength, patience, opportunity and good health to complete this study.

I would like to express the deepest gratitude to Dr. Zulkefli Bin Mansor, a lecturer at the National University of Malaysia, who is assigned to be my supervisor. During the completion of this research, my supervisor always guided, gave me advice, suggestions, motivation and support so that I can complete this research.

I would also like to express my sincere appreciation to all my respondents from government agencies: BAPPEDA, DISKOMINFO, BKD, and LPSE that are located in Padang city, Payakumbuh city, Solok city, Tanah Datar Regency, Sijunjung Regency, Bukittinggi city and Lima Puluh Kota Regency in West Sumatera Province. Thank you for collaboration and participation in the data collection.

I would like to thank all my colleagues at UKM for their assistance and friendship, and for having created a pleasant working environment throughout my years in UKM.

To my dearest father, Mr Darmawi and my beloved mother, Mrs Nuraini - thank you for being supportive. My deepest thanks go out to my best friend Wirman for the motivation, support and encouragement given during the course of this study.

Last but not least, I gratefully acknowledge the financial support provided by my sister, Leni Marlina and my brother in law, Zainal Abidin.

ABSTRACT

Software cost estimation is a substantial issue in software development, even though techniques and models of software cost estimation has been proposed for the last thirty years. However, software cost estimation model developed based on the local project data has higher accuracy compared to the existing model in as it reflects on the software development in a particular organization. Not all the parameters of the existing models are applicable with the local software development environment. Based on the new empirical evidence, the public sectors more often face estimation failure due the result inaccurate. As the impact, the project to overshoot the budget, get delayed, face termination or remain incomplete in terms of the scope and requirements of project. Hence, the purpose of this study is to determine the critical factors of software cost estimation in public sectors. Besides that, the objective is to identify software cost estimation practice in Indonesian regional government. Therefore, this study focus on the software cost estimation in regional government of Indonesia because Indonesia use an autonomy concept. The methodology used is a combination of quantitative and qualitative approach. The quantitative method used the questionnaire as the data collection instrument, while the qualitative method involved interviewing personnel in the software cost estimation in public sector. The quantitative data is analyzed by using RASCH model because it can use to identify the respondent competency, the level difficulties of the items and the relationship between the item and respondent. The result of this study reveals that the current technique of software cost estimation is not effective which has an impact on the estimation result accuracy. The most influential factor is found to be a lack of experience in previous related project and absence of proper tools to estimate software cost. Besides, the scope and requirements are not well defined during the project proposal phase, which causes the scope and requirements to change. Based on a quantitative approach, the study indicates that there are six critical factors that have significant impact on software cost estimation result: programmer capability, top management support, the understanding of the top management of the objectives of the project, risks management, knowledge, and competency of the project manager and top management involvement in the project. As the outcome of this study, a software cost estimation model is developed which uses to prepare conducting software cost estimation. The estimator is able to identify the factors influence of software cost estimation. Hence, the proposed model can assist the government, practitioners or software development communities in estimating the software cost so that it can enhance effectiveness, reliability and accuracy of the cost estimated. It can also use for other researchers as reference for doing a further research on software cost estimation.

ABSTRAK

Anggaran kos perisian merupakan perkara yang besar dalam pembangunan perisian, walaupun teknik dan model perisian telah dianggarkan selama tiga puluh tahun terakhir. Walau bagaimanapun, anggaran kos perisian yang dibangunkan berdasarkan data projek tempatan mempunyai ketepatan yang lebih tinggi berbanding model sedia ada kerana ia mencerminkan perkembangan perisian dalam organisasi tertentu. Tidak semua parameter model sedia ada terpakai dengan persekitaran pembangunan perisian tempatan. Berdasarkan bukti empirikal yang baru, sektor awam lebih sering menghadapi kegagalan anggaran akibat keputusannya tidak tepat. Sebagai kesannya, projek ini akan menembusi bajet, ditunda, menghadapi penamatan atau kekal tidak lengkap dari segi skop dan keperluan projek. Oleh itu, tujuan kajian ini adalah untuk menentukan faktor kritikal anggaran kos perisian dalam sektor awam. Di samping itu, tujuannya adalah untuk mengenal pasti kaedah penganggaran kos perisian dalam kerajaan serantau Indonesia. Oleh itu, kajian ini memberi tumpuan kepada pengiraan kos perisian di wilayah pemerintahan Indonesia kerana Indonesia menggunakan konsep otonomi. Metodologi yang digunakan adalah gabungan pendekatan kuantitatif dan kualitatif. Kaedah kuantitatif menggunakan soal selidik sebagai instrumen pengumpulan data, sementara kaedah kualitatif melibatkan menemubual kakitangan dalam anggaran kos perisian di sektor awam. Data kuantitatif dianalisis dengan menggunakan model RASCH kerana ia boleh digunakan untuk mengenal pasti kecekapan responden, tahap kesukaran dan hubungan antara subjek dan responden. Hasil kajian ini menunjukkan bahawa teknik penganggaran kos perisian semasa, tidak berkesan, dimana ia memberi kesan terhadap ketepatan keputusan anggaran. Faktor yang paling mempengaruhi adalah disebabkan kurang pengalaman dalam projek berkaitan sebelumnya dan tiada alat yang sesuai untuk menganggarkan kos perisian. Selain itu, skop dan keperluan yang tidak jelas dalam fasa cadangan projek, yang menyebabkan skop dan keperluan berubah. Berdasarkan pendekatan kuantitatif, kajian menunjukkan bahawa terdapat enam faktor kritikal yang mempunyai kesan yang signifikan terhadap hasil anggaran perisian: keupayaan pengaturcara, sokongan pengurusan atas, pemahaman tentang pengurusan objektif utama projek, pengurusan risiko, pengetahuan, dan kecekapan pengurusan projek serta penglibatan pengurusan atasan dalam projek. Hasil daripada kajian ini, satu model penganggaran perisian telah dibangunkan bagi menyediakan penganggaran kos perisian. Peramal dapat mengenal pasti faktor yag mempengaruhi penganggaran kos perisian. Oleh itu, model yang dicadangkan ini dapat membantu kerajaan, penguna atau komuniti pembangunan perisian dalam menganggarkan kos perisian supaya ia dapat meningkatkan keberkesanan, kebolehpercayaan dan ketepatan anggaran yang dianggarkan. Ia juga boleh digunakan oleh para penyelidik sebagai rujukan untuk melakukan penyelidikan selanjutnya mengenai anggaran kos perisian.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Meaning**
ACAP	Analyst Capability
AEXP	Application Experience
CO	Committed Officer
COCOMO	Constructive Cost Model
CPLX	Complexity of The Product
CUT	Code and Unit Test
DATA	Size of Application Database
DD	Detail Design
DoD	Department of Defense
ECF	Environment Complexity Factor
FLEX	Development Flexibility
FPA	Function Point Analysis
ICT	Information, Communication, and Technologies
IT	Information Technology
IT	Integration Test
LEXP	Programming Language Experience
MODP	Use of Modern Programming Practice
OEC	Owner Estimate Cost
PCA	Processing Complexity Adjustment
PCAP	Programmer Capability
PF	Productivity Factor

PMAT	Process Maturity
PREC	Precedentedness
RELY	Required Software Reliability
RESL	Risk Resolution/Architecture
RPD	Requirements Planning and Product Design
SCE	Software Cost Estimation
SCED	Required Development
SDLC	Software Development Life Cycle
SLIM	Software Life Cycle Model
SLOC	Source Line of Code
STOR	Memory Constraints
TCF	Technical Complexity Factor
TIME	Run Time Performance Constraints
TOOL	Use of Software Tool
TURN	Required Turnaround Time
UCP	Use Case Point
UUCP	Unadjusted Use Case Points
VEXP	Virtual Machine Experience
VIRT	Volatility of The Virtual Machine Environment

CHAPTER I

INTRODUCTION

1.1 RESEARCH BACKGROUND

Computer software has become essential in our daily lives; supporting education, business, entertainment, government operations, healthcare, military and transportation activities. Each of these sectors requires updating, maintaining or replacing software to enhance services. However, software development requires a good strategy and planning so that the process is effective and efficient (Ongere 2013; Mens 2008). The most essential part of the software project development involves predicting the project costs required to effectively complete the project. Software cost estimation is a complicated task in software management due to inaccuracies that can occur in estimation (Abdullah et al. 2012). It can result in either over or underestimation (Rajkumar & Alagarsamy 2013), both of which can lead to project failure (Potdar et al. 2014). Hence, cost estimation poses a substantial challenge in software project development although in current times, there is greater understanding of software project performance and software development method selection (Ramasubbu & Balan 2012; Rajkumar & Alagarsamy 2013; Ramesh & Reddy 2016; Shekar & Kumar 2016). Besides, the difficulty in estimating software cost is due to the intangible nature of software, making the cost estimation process even more complex (Kumari & Pushkar 2013).

Thus, accuracy in software cost estimation is important, as this determines the success of a project. According to Leena (2012), the accuracy of software cost estimation is vital due to the following reasons:

- It could be used to identify and manage the resources of the project wisely.

- The customer expectation of the actual cost should be in line with the estimation cost.

- It could be used to evaluate the effect of changes to be made and to guide re-planning of the project.

- Project control and management could be made easier by using resources wisely.

- It could assist the development project to meet the overall business plan.

Based on the Standish Group research in Chaos report (2014) it has been indicated that 52.7% projects would cost 189% than the original cost estimation. It shows that the cost estimate is inaccurate. As a result, the project was overrun and over-budgeted leading to additional costs and extra time to complete the project. Moreover, Haslindah, Azizah & Othman (2014) identified those cost estimation failures would influence the project sustainability. Due to ineffective cost estimate, the government ICT project failure in the Malaysian context was high. Consequently, 16% of projects would be canceled before they ever accomplished their objectives, 53% of projects would be over-budget twice than the original estimates, and less than 31% of projects successful (Haslindah, Azizah & Othman 2014).

As mentioned in Chaos report (2014), the American companies and government agencies spent $81 billion for canceling the project; while if they had wanted to complete the project; they would have had to incur an extra cost of $59 billion. As a result, the project would be overrun and over-budget. Hence, it showed that the error of software cost estimation results is not only faced by private sectors, but it is also faced by public sectors. According to Mensah (2003), the public and the private sectors are often forced to cancel software projects because the cost and the dateline far exceeded the initial schedule planning.

Due to ongoing-reports on the inaccuracy of software cost estimation and the importance of software cost estimation that has been stated by Leena (2012), the researcher conducted this study to suggest ways to improve the existing models by describing the system complexity from various factors to get a better result estimation. As stated by Shekar &Kumar (2016); Eberendu (2014); Patil, Badjate & Joshi (2014); Tailor, Saini & Rijwani (2014); Kumari and Pushkar 2013; Rajkumar &Alagarsamy (2013); Abdullah et al. (2012); Asiegbu & Ahaiwe (2011) that even though there are techniques or models of software cost estimation that have been proposed for the last thirty years, those models still need improvement because each model has strengths and weaknesses. Moreover, according to Kaur & Salaria (2012), all the existing techniques have different levels of accuracy hence one cannot determine if any one technique is better than others. It depends on the data set or parameter chosen to estimate the cost.

1.2 PROBLEM STATEMENT

Getting the exact result of software cost estimation continues to be problematic to government and the private sector. Nevertheless, based on the new empirical evidence, the public sectors are more often face the estimation failure which causes the project over budget, delay and undelivered all the project requirements (Ubani et al. 2015; Haslindah, Azizah & Othman 2014). Software cost estimation in government is extremely complicated as the estimation has to be conducted in the initial stage of the project. Hence, in the first stage of a project, the government requires to prepare detailed scope and requirements of the project; unfortunately, the project team not equipped in preparing it (Imam & Arry 2015). As stated by Haslindah, Azizah & Othman (2014); Phongpaibul & Aroonvatanaporn (2014); Chaos Report (2014) that the project manager and top management are lack of knowledge, skill, experience, and familiarity in ICT. Therefore, the scope and requirements can easily be highly over or underestimated and have an impact on modifying again. However, the scope and requirement are significant in estimating the cost because it encompasses entire activities of software development which is used to estimate the size of the project (Medsveska & Berzisa 2015).

Likewise, many models have been established by previous researchers. However, those existing models developed are based on a large number of historical projects from the USA. Hence, it necessary to adapt those models to the environment in which they are to be used. The software cost estimation model which is developed based on the local project data has higher accuracy compared to the existing model in as it reflects on the software development in a particular organization. Not all the parameters of the existing models are applicable with the local software development environment (Javed et al. 2013; Xuang et al. 2007; Suharjito & Budi 2006; Dillibabu & Krisnaiah 2005). Besides that, the existing models commonly use Lines of Code (LOC) and function point for measuring the software size. The primary problem is the lack of a universally accepted definition of line of code. As the impact, many variations of counting the Lines of Code (LOC) exist. Some models include the comment lines, while other models do not calculate it. LOC is language dependent; the calculations are also varied as it depends on the programming language used for the software development. While, the function point does not consider the tools, methodologies, programming languages, database management systems, processing hardware or any data processing technology in the estimation. It really depends on the functionalities of the system requirement (Matson et al 1994). It is very difficult to be implemented at the early stages of the project (Patil, Badjate and Joshi (2014). Therefore, it is difficult to implement those existing models in government due to the software cost estimation is performed at an early phase of the project whereby the scope and requirements of the project are no details yet.

As stated by (Ramesh & Reddy 2016; Shekar & Kumar 2016; Patil, Badjate & Joshi 2014, Rajkumar & Alagarsamy 2013; Khatibi & Jawawi 2012) that the common existing model used is COCOMO models which ignore other important parameters such as customers skills, cooperation, knowledge, hardware issues, personnel turnover levels and all documentation (Ramesh & Reddy 2016; Shekar & Kumar 2016; Patil, Badjate & Joshi 2014, Rajkumar & Alagarsamy 2013; Khatibi & Jawawi 2012). Basically, those factors are significant due to having a big impact on the estimation result. Consequently, there is a need to have a model which describe the software cost estimation complexity from various factors. As mentioned by Kaur & Salaria (2012) that the difficulties in estimating the software cost influenced by many factors such as lack of historical data, lack of project plan, poor management, project team ability, the pressure to lower estimation and project uncertainty.

In Indonesia, each government institution requires preparing the budget plan before implementing the project (Adi Nugroho et al. 2009). According to the Presidential Regulation no. 70 of 2012, only the software that can estimate owner cost is considered good (Subsection 66 number (5) item a). Before estimating the owner estimate cost, the Committing Officer (CO) requires identifying the requirements and specification. Unfortunately, the OEC is very difficult for the software cost estimation due to there are no standard techniques for reference and guidance (Sholiq et al. 2016). According to Sosa Star Web (2017) – one of the software providers in West Sumatera, there is no specification technique used to estimate the software cost. They perform owner estimate cost just based on *paguanggaran* [1]. Besides that, the government often changes the scope and requirements of a project which causes project delay.

Furthermore, the common cost estimation technique uses by the government is an expert judgment which depends on the experts. The weakness of this technique is the method still cannot be measured because it is hard to document the parameters by the experts. The experts may be biased, optimistic, and pessimistic (Holgeid &Thompson 2013; Rajkumar & Alagarsamy 2013; Whitfield 2007). When software features and cost are unquantifiable, it can lead the estimation to be potentially highly biased and unjustified due to it is difficult to verify and validate it (Phongpaibul & Aroonvatanaporn 2014).

In summary, the inaccuracy of software cost estimation result can cause the project over budget, project delay, project termination, and undelivered all the project requirements. Due to the aforementioned concerned, there is a need to have a model which describe the software cost estimation complexity from various factors. Thus, it

[1]*Pagu anggaran* is the maximum amount of budget allocated for each program that was produced by the Ministry of planning and the Ministry of finance of Indonesia.

can assist the government of Indonesia to effectively and quantitatively measure and analyze the factors that significantly impact software development in Indonesian government context.

1.3 RESEARCH QUESTIONS

The research questions of this research are as follows:

1. What are the software cost estimation practices used in Indonesian regional government?

2. What are the critical factors of estimating the software cost in Indonesian regional government??

3. How are the parameters used to estimate the software cost?

1.4 RESEARCH OBJECTIVES

The objectives of this research are as follows:

1. To identify the software cost estimation practice Indonesian regional government.

2. To determine the critical factors that significantly impact software cost estimation in developing software in Indonesian regional government context.

3. To formulate a software cost estimation model based on identified factors that influence the software cost estimation.

1.5 RESEARCH SCOPE

The research focused on software estimation in the government project because the previous researchers mostly had focused on software cost estimation in private sectors. Hence, the study about software cost estimation in government project is limited, while cost overrun in public sectors is higher than private sectors. The researcher focused on software cost estimation in the regional government project in Indonesia due to the Indonesian government system is used regional autonomy concept. According to Presidential Regulation no 22 of 1999 which is based on the 1945 Constitution that the local government has the freedom to organize the regional autonomy. While the regional authority in all areas of government except in the field of foreign policy, defense of security, judicial, monetary and fiscal, religious, and other fields of authority such as national planning and development control, national macro, financial balance fund, administration system, state and state economic institutions, guidance and resource empowerment human power, utilization of natural resources and high technology strategic, conservation, and national standardization. It has been stated in chapter 5 of article 7 of Presidential Regulation no 22 of 1999. Thus, the software cost estimation in regional government is managed by the regional government itself.

The outcome of this research is the software cost estimation model for a government project. The model is useful for the government agency employees who are involved in estimating the software cost. Besides that, the model is important in preparing the cost estimation due to based on the model the estimators will be able to determine the factors that influence of the software cost estimation. As a result, the estimators will be able to identify the factors that might occur during the software development. Hence, it can reduce the risk of the project from overrunning and over budget. However, the model has a limitation which may not include all the factors that influence of the software cost estimation in government project yet. Therefore, there is a need to do a further research on other factors that might influence the software cost estimation so that it can be added to the model.

1.6 SIGNIFICANCE OF THE RESEARCH

The findings of this research are essential for the government, project managers, academicians, researchers and practitioners who are involved in estimating the cost of software development in any organizations. Hence, the consequences of this research are:

1. Identifying the software cost estimation practice in the regional government of Indonesia.

2. Analyzing and understand the software cost estimation process of Indonesian regional government so that the regional government gets the budget from the central government.

3. Identifying the software cost estimation factors that influence the software cost estimation in regional government.

4. The outcome of this research is a new model that describes the complexities of software cost estimation, specifically for public sectors and generally for private sectors.

1.7 EXPECTED RESULTS

The expected result of this research is to provide a clearer understanding of the current practice of software cost estimation for development in the government project in Indonesia. Besides that, it describes in detail the complex factors that influence the software cost estimation so that the public sectors and the private sectors can become aware of those factors that might cause ineffectiveness of the software cost estimation. Based on that, the researcher would produce a new model that describe the complex factors of software cost estimation so that the government considers those factors in estimating cost for software development project. As a result, the software cost estimation process is more effective and efficient whereby the estimation cost is in line with the actual cost.

1.8 RESEARCH METHODOLOGY

This study consists of four phases which are conceptual study, empirical study, development of model and model validation. In the first phase, the researcher explores previous related research and literature. As an outcome, the researcher is able to identify the research problems and produce the background of the research, problem statement, research questions, research objectives, research scope, and literature review. Likewise, in the second phase is the empirical study, the researcher conducts the pilot test and the actual study. The result of the conceptual study and the empirical study used to develop the model is arrived at. Lastly, the model is validated by the experts.

1.9 THESIS ORGANIZATION

The organization of this study consists of six chapters which are: Chapter I (Introduction), Chapter II (Literature Review), Chapter III (Research Methodology), Chapter IV (Discussion), Chapter V (Model Development), and Chapter VI (Conclusion and Recommendation).

The first part of the chapter includes the background of the research, problem statement, research questions, research objectives, the scope of the research, the expected result and the resulting approach. These aspects are the basis of this study. The second chapter contains literature review which explains the related topics on software cost estimation such as software development, software cost estimation history, cost estimating phases, the current practice of software cost estimation in public sector, software cost estimation metrics, software cost estimation techniques and software cost estimation model. This literature review content is based on the previous research that has been done by other researchers.

Chapter three describes research methodology that comprises research design and research phases. Research phases explain the steps of the study which are conceptual study, empirical study, model development and model validation. Chapter four discusses the findings of the quantitative and qualitative approaches. Chapter five is the output of the study which is the model of software cost estimation. It consists of the summary of the research and recommendations for improvement to this study.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

Accurate software cost estimation is essential for the software development process because the estimation result uses to plan, monitor and control the project during the project development. Thus, overestimation and underestimation affect the process of a project (Arnuphaptrairong 2016). Based on Standish Group Report on 25,000 software projects data in 2011-2015 reveal that 27-31% of the project is successful (project completed within time and budget estimate) and 17-22% of the project is failed (project canceled or terminated). There is also 49%-56% that falls into a challenging project which means that the project is completed with overrun and over budget. These occurred due to lack of cross-checking of the judgment given by experts in the input or output estimation. Besides that, the organization might not use or misuse the existent models or techniques (Boehm 2017). Therefore, the software cost estimation is still a challenge in software development due to the estimation result is inaccurate. Either overestimation or underestimation is a problem due to both overestimations might cause wasting the resources, while underestimation might cause the project to cancel or add more budget so that the project completed. Hence, the accuracy of the software cost estimation is very significant towards the successful a project.

Likewise, Heemstra (1992) stated that the software cost estimation issues could occur in many ways such as organizational, sociological, psychological, and technical. The organizational factors include staff responsibilities, decision making, and monitor and control the software development process. While the examples of sociological and psychological factors are leadership, commitment and the solidarity of the project team. Then, technical factors are significant too which consist of equipment required for the software development process such as tools, software, hardware and the programming task. According to Patil, Badjate & Joshi (2014), the failures in software cost estimation usually occurred during planning and estimation phase because incomplete of requirements, insufficient planning and the estimation that conducted in the early phase of the project. Therefore, the academic researchers and practitioners have been researched on software cost estimation so that the software cost estimation issue should be solved (Arnuphaptrairong 2016; Holgeid & Thompson 2013). So, there are many factors influence of software cost estimation that gives impact on the estimation result and the effectiveness of a project.

2.2 THE SUCCESS FACTORS OF SOFTWARE DEVELOPMENT IN PUBLIC SECTORS

Moreover, the software cost estimation process is integrated with the software development process because both processes are used to make an adequate plan so that the project completed in particular time, budget and resources (Rajkumar & Alagarsamy 2013). According to Agarwal, Tayal & Gupta (2010), the software development process is a set of activities that are involved in developing a software. There are four fundamental process activities that are common to all software process which are:

1. Software specification.
   The first process in software development is defined as the software functionality and constraints on its operation.
2. Software development
   The second process is producing the software based on the software specification given.
3. Software validation
   Software validation is to validate the software that has been developed to fulfill the requirements given by the customers.
4. Software evolution
   The software must evolve to meet changing customer needs.

Likewise, Mensah (2003) stated that software development failures could happen in any organization regardless of their organizational size, geographic region, industry, and market group. Thus, it could occur in private sectors, public sectors or any government institutions. Haslindah, Azizah & Othman (2014) said that many national, state and local government put the information technology (IT) transformation into the agenda to enhance government activities and services. As the impact, there are many types of government software development project such as ICT (information and communication technology) research, ICT strategic plan, application system development, hardware and software procurement, ICT system enhancement, expansion of ICT system as well as ICT compliance and fortification.

Furthermore, the UK Public Accounts Spending Committee (PASC) has been identified the factors that influence public IT project failures. First, the government does not have enough information on its IT needs, as the impact, they are not able to manage it. Then, lack of skills and competence in ICT procurement, thus, the government rely on the outsourcing to manage the project. Besides that, the security requirements are over-specified, and the project is multifaceted. The failure in integrating IT into the wider policy that has been implemented in government (PASC 2011). While according to Holgeid & Thompson (2013), the public project failures influence by contextual factors, management factors, and technical factors. The contextual factors refer to the size, volatility and overoptimistic towards the project. The size of projects includes effort (person-months), time, team size and budget. The bigger the size of a project is, thus the higher risk of the project failure. While for the volatility, it can be affected by the changes made during the project development such as project manager, sponsor, scheduled, budget and scope. Furthermore, the technical factors can be because of inappropriate requirements, technical design, development tools, and user documentation. It is also because of poor test planning and technical support. While for management factors can be because of poor estimation method, lack of leadership skill, poor of communication, poor risk management and lack of management support.

Likewise, Medvedska & Berzisa (2015) said that the successful government software development project was influenced by environmental factors such as government institution hierarchical structure and functional matrix project organizational structure. It is also because the lack of qualified personnel due to the salary in private sectors are more than in the government institutions. During the implementation process, it depends on competence leader, operational experience and personality entirely who have significant roles in the whole project which might cause the conflict possessions issue.

Also, Raffo, Pfahl & Wang (2007) stated that there is no framework for the internal rules and regulations for software development project activities. Additionally, the historical information and lesson learned are unavailable due to the knowledge base is not sustained. The government project could run through outsourcing. Unfortunately, lack of contract pricing standard from government contractors which is used as the reference to propose. As a result, the government contract pricing is mostly by subjective judgment. It would affect the actual cost is not congruent with the cost estimation. Most of the government project is used three approaches which are insourcing, outsourcing and co-sourcing. In-sourcing means the project is handled by internal personnel and expert, while external services perform the outsourcing. Lastly, co-sourcing is the collaboration between the internal expert and the external services. There is six failure of root causes of IT project in the government which is: project management, top management, technology, organizational, complexity and also process factors (Haslindah, Azizah & Othman 2014).

On the other hand, according to Medvedska & Berzisa (2015), software development project in the public sector is quite different from the process in the private sectors. In the private sectors, the process is more flexible regarding resources limitation, documentation, the system development approach and so forth. Besides that, the private sector environments in software development are task culture, democratic management style, projected matrix organizational structure and less of inferior employee's skills level. While in the government sectors, it has its characteristics that would give negative impact the process of development. Likewise, the private sector is less interdependence with the organizational boundaries. Therefore, the process is easier and flexible. The project manager more focuses on the internal coordination while the public sector is more concerned about the linkages external to the organization (Furumo et al. 2006). Thus, the IT project failure in the public sector causes by many factors whereby the process is more complicated compared to the private sectors. According to Medvedska & Berzisa (2015), the software development in the public sector requires to meet the following criteria:

- The level of software system interdependency is high.
- The legislation is identified as the government institution business process that needs reinforce the software system.
- High-quality project documentation because of the outsourcer developer responsible for software system requirements analysis, design and development. The government only perform the software system acceptance testing, implementation and maintenance.
- Government software system process controlled the information in order to ensure security, confidentiality and integration of information.

So, there are many factors influence the software development process which is summarized in Table 2.1.

Table 2.1 Factors influence of software development

No	Factors	References
•	Skills & competence of the employees in ICT	PASC (2011); Holgeid & Thompson (2013); Medvedska & Berzisa (2015)
•	Contextual Factors: • The size of the project (Effort (person-months), time, team size, and budget). • The volatility of the project. • Over-optimistic towards the project.	Holgeid & Thompson (2013)
•	Management Factors: • Estimation method • Communication • Risk Management • Top management support.	Holgeid & Thompson (2013)
•	Technical factors: • Requirements specification. • Technical design. • Development tools. • User Documentation	Holgeid & Thompson (2013)

to be continued…

…a continuation	
• Environmental Factors: • Government institution hierarchical structure. • Functional matrix project organizational structure.	Medvedska & Berzisa (2015)
• The information on IT needs	PASC (2011)
• Security requirements are over specified.	PASC (2011)
• The complexity of the project.	PASC (2011)
• The historical data about the previous projects	Raffo, Pfahl & Wang (2007)
• The framework for SD activities	Raffo, Pfahl & Wang (2007)

2.3 SOFTWARE COST ESTIMATION

Software cost estimation is a process of estimating the cost, effort, and productivity required to develop a software project (Shekhar & Kumar 2016; Rajkumar & Alagarsamy 2013; Eberendu 2014; Zaid et al. 2008; Lee, Titchkosky & Bowen 2002). The processes of the software cost estimation are determining software size, estimating the needed effort, derived the schedule and calculating the software cost (Shekar & Kumar 2016; Kumari & Pushkar 2013). The essential aspects of the software project estimation are to balance the "magic" triangle which comprises the effort, schedule, and quality (Rajkumar &Alagarsamy 2013). Furthermore, software cost estimation is a significant phase in software development. However, it has been a tough activity since the beginning of the computer era in the 1940s to produce the accurate result (Ramesh & Reddy 2016). The estimation accuracy is significant because it can be used for proposal, contract negotiations, scheduling, monitoring and controlling of the project. (Borade & Khalkar 2013; Ramesh & Reddy 2016).

According to Khatibi & Jawawi (2010), many estimation methods have been proposed in the last decades such as expert judgment, analogy, neural network, COCOMO and so forth. However, none of the technique is the best which can give the accurate and reliable results for various situations of the project (Khatibi & Jawawi 2010). Zaid et al. (2008) stated that software cost estimation is different with other cost estimation in another field due to it is an intangible product. There is no standard rules and regulations that need to follow for estimating the cost. Therefore, it is more complicated and difficult to estimate which cause the inaccurate estimation. Borade & Khalkar (2013) & Ramesh & Reddy (2016) mentioned that the inaccuracy cost estimation is divided into two categories which are overestimation and underestimation that will influence the project. The impact of overestimation might be wasting resources, while underestimation can cause additional cost or project cancellation. Thus, both inaccuracies cause the organization to lose.

As stated by Potdar et al. (2014), accurate cost estimation is significant to ensure the project complete within a specific period and budget. However, there are various factors influence the software cost estimation which determines the accuracy. Therefore, the estimators have to consider all the factors influence the estimation because inaccuracy result of estimation may lead to the project overrun and too optimistic on the software development. There are five significant critical measurements of the software cost estimation which are effort hours, time, resource requirements and risk occurred. While Ramesh & Reddy (2016) stated that software cost estimation consists of one or more determinations such as effort (usually in person-months), project duration (in calendar time) and cost (in dollars). Moreover, Sommorville (2011) stated that there are determinations required in estimating the software cost such as:

- Hardware cost, software cost, and maintenance.
- Travel and training costs.
- Efforts costs to pay the software engineers.

According to Abdullah et al. (2012), in order to get an appropriate cost estimation, the following core parts of software project development should be included in estimating the software cost:

- Project domain
- Technical resources
- Financial issues
- Well inspected method for finance distribution

On the other hand, software cost estimation will never be produced the exact result like science because there are many aspects affect the estimation calculation such as human, technical, environmental and political. Likewise, cost estimation for software development is difficult to predict (Lee, Titchkosky & Bowen 2002).

So, there are elements that determine the software cost estimation. The element is not only about the cost, but also the followings element which are:

- Person-months (effort).
- Schedule.
- Quality.

In order to make the project successful, the project should balance the triple constraints which are shown in Figure 2.1 that includes time, cost and scope. Hence, the project completed within the estimation schedule and cost. The project also fulfils the scope and requirements of project that that have been identified in the early phase of project development (Schwalbe 2013).

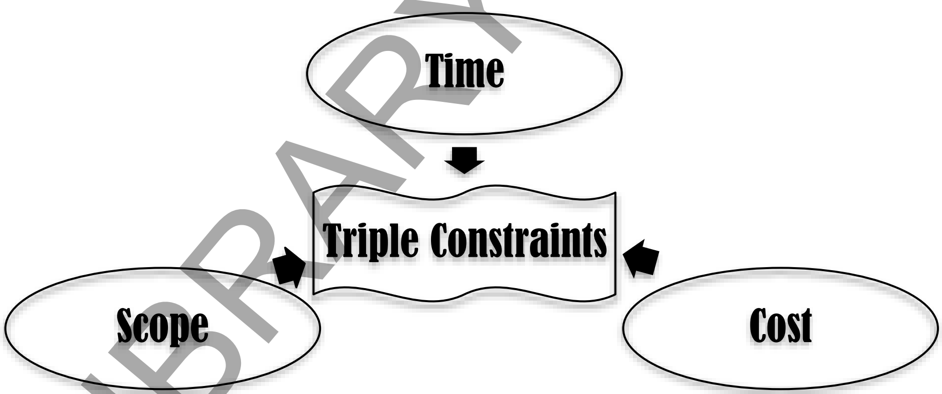

Figure 2.1 Triple Constraints

Source: Schwalbe (2013)

2.3.1 Previous Study of Software Cost Estimation

The software cost estimation techniques have been developed from time to time to control and monitor the estimation process. There are techniques or models of software cost estimation has been proposed for the last thirty years. However, each of it has its strengths and weaknesses. Hence, those models still need improvement due to its still inaccurate (Shekar & Kumar 2016; Eberendu 2014; Patil, Badjate & Joshi

2014; Tailor, Saini & Rijwani 2014; Kumari & Pushkar 2013; Rajkumar & Alagarsamy 2013; Abdullah et al. 2012; Asiegbu & Ahaiwe 2011). Hence, in order to get more accurate results, those models still need to be improved.

According to Boehm & Chulani (2000), in software engineering, it is required to used cost and schedule models and estimation techniques for the following purposes:

- Cost estimation: To ensure the level of accuracy of the estimation so that it can achieve the expectation and requirements.
- Tradeoff and Risk Analysis: To clarify the quality decision of cost and schedules for software project such as scope, project team, tools, and reuse.
- Project Planning and Control: To monitor and control the project based on cost and schedule breakdowns which include component, phase, and action taken for the project.
- Software improvement investment analysis: To predict the cost and also advantages of the development strategy concerning tools, reuse, and process maturity.

According to Suri & Ranjan (2012), many software cost estimation techniques and software cost estimation models are available. However, no technique is able to apply for all situation. While the software size and complexity are increased, therefore it is difficult to get an accurate estimation. As stated by Singh (2014), the research on software cost estimation has been started in 1965. Therefore, the partial model of software cost estimation has been used since the late 1960s and early 1970s. While in 1970, the researchers were introduced several models such as SLIM (Software Life Cycle Model) that was introduced in 1992 by Putnam & Myers. Then, Jones also introduced Check Point in 1997. Park came up with PRICES in 1988 and SEER was developed by Jensen in 1983. Then, COCOMO was introduced Boehm in 1981. Even though there are techniques available, the software cost estimation still difficult phase in the software development due to the size and the complexity is increased too.

Suri & Rajan (2012) & Zaid et al. (2008) said that in 1960, Frank Freman had been developed parametric estimation model concept. While in 1970, there were not specific estimation techniques yet, Albercht found the way for estimating stages through thumb rules and some simple algorithms which are done manually for measuring the size, efforts, and cost. In fact, it has high reliability up to now. The previous researchers continued to analyze what are the aspects that influence cost for software development through correlation and regression techniques. There were four techniques introduced which are COCOMO (Constructive Cost Model) that introduced by Barry W. Boehm and C. Abts. Furthermore, Barry Boehm developed an automated software estimation in 1970 due to facing the difficulties in estimating the large system. The estimation software was written in his book "Software Engineering Economics" that was released in 1989. He was developed COCOMO (Constructive Cost Model). Then, Allan Albercht and John Graffney were formulated function point analysis (FPA) model in 1975. Function Point Analysis (FPA) was used to estimate the size and effort that is estimated based on the input, outputs, inquiries, logical files and interfaces. Then, Frank Freiman and Robert Park have been developed PRICE model development in 1977 for hardware in enhancing the parametric estimation techniques.

Halasted also introduced new cost estimation software that could not be used for a long time which is based on the number of operand and operators. Likewise, Lawrence H. Putnam came up with SLIM (Software Life Cycle Model) in 1979 that was established in 1979. Furthermore, in 1980, the U.S. Department of Defense (DoD) introduced Ada programming language, and it is used to estimate the software cost which is name as Ada-COCOMO. These techniques were enhanced in 1980 into ADA COCOMO due to it modified by using ADA programming language. Additionally, Capres Jones added the functionality of FPA with a complex algorithm impact on the FPA calculation. Then, Charles Symons established Mark II Function Points to overcome the problems that existed in FPA. Furthermore, Barry Boehm et al. made improvement ADA COCOMO into COCOMO II that includes Called Application Composition, Early Design and Post architecture model as main sub-models (Suri & Rajan 2012). Other techniques could be used in object-oriented environments, real-time system, agile projects and so forth (Zaid et al. 2008).

2.3.2 Estimating Cost Phases

There are four phases of cost estimation which are the estimation preparation, creation, management and estimation process improvements. All this process is interconnected to each other. Estimation preparation is a process to gather the information, creating the work breakdown structure and deciding the estimation approach according to the historical data. Then, the team creates the actual estimation, inform the clients and get approval from them. After that, managing estimation is for the entire project whereby check the progress of the whole process so that it is on track as well as refining the estimation if there is a change to be made (Pinto 2010). As said by Zulkefli et al. (2016) it challenges to manage the cost of the project. Thus, the poor cost management would give impact on the software project.

According to Schwalbe (2013), cost management includes four processes which reflect on the organization, strategic planning, goal, and mission. The processes include resource planning, cost estimation, cost budget and cost control. A Foremost thing, the cost management usually is one of the weaknesses of information technology project. Lastly, estimation process improvement is to evaluate the estimation made in the project so that the team could determine the weaknesses. As a result, they would be able to enhance the next estimation in the future (Pinto 2010). Furthermore, each phase has different input, activities, and output which is shown in Table 2.2.

Table 2.2 Cost Estimating Phases

Phase	Estimation Preparation	Estimation Creation	Estimation Management	Improve Estimation Process
Input	Project documents Experts Estimating techniques Constraints and assumptions Additional influences Historical project information	Project estimating approach Estimating information Estimators Enterprise environmental factors Organizational process assets	Baseline estimates Approved changes Resource plan Work performance information Organizational process assets Project estimating approach	Baseline estimates Project estimating approach Historical project information Baselined estimates Updated estimates/forecasts Change log Actuals versus estimate data Stakeholder feedback Organizational process assets

to be continued…

…a continuation

Activities	Create project estimating approach	Use of analogs techniques Use of parametric techniques Use of bottom-up techniques	Manage estimates Use of tools and techniques	Gather estimating process Assess the estimating process Determined root causes and lesson learned Develop an action plan for changes Implement action plan and share lesson learned
Output	Estimating approach Estimating information	Completed estimates Basis of estimates	Updated estimates Updated forecast Updated change request log Reporting and communication	Updated tools and techniques Updated organizational process assets

Source*:* Pinto (2010) & Schwalbe (2013)

2.3.3 Software Cost Estimation in Public Sector

Many factors influence software cost estimation which are data availability, data quantity, unrealistic assumptions, fewer details design specifications, project complexity, product size, available time and level of technology. Other factors are lack of cost estimators experience, historical data quality, lack of user involvement, insufficient requirements, lack of executive support, developer incompetence and software development method (Zulkefli et al. 2016; Kumari & Pushkar 2013; Ministry of Transportation and Infrastructure 2013; Leena 2012; GAO 2009; ISBG 2004). Those factors are affecting the estimation accuracy result which causes the project can be canceled, exceeded the budget and time, or underdeveloped all functionalities (Leena 2012; Ramesh & Reddy 2016). Therefore, cost estimation is significant in software development since it influences the success or failure of a project (Abdullah et al. 2012).

Ubani et al. (2015) found that the possibility of public sectors project to be overrun was higher than the private sectors project because many factors influencing the software cost estimation process such as project complexity and inexperience of cost estimator. Besides that, purposely understating project cost to ensure acceptance of the project and to gain funding commitments. According to Flyvberjerg et al. (2002) that was cited by Ubani (2015), the noble lie principle has stolen into project management especially in the public sector, and people are using the noble principle as a foundation when it comes to miscalculating expenses of public projects. The latter could be a situation where a certain project would be in the interest of the people. On the other side, when it comes to the total cost to be invested into the project, it always tends to freak out the public due to the huge cost of money to be invested. If there are any cost changes, it will give impact to the government, senior management, political leaders and auditing agencies. The most important thing it would give the impact on public trust with the government (Ministry of Transportation and Infrastructure (2013). Therefore, it needs attention especially based on the previous research show that the successful record of software cost estimation project is "poor." GAO (2009) stated that there are many factors influence the project overrun which are detailed documentation availability, risk analysis conducted, historical data of the previous projects, well trained and experience analyst, adequate budget, adequate cost reserve, unrealistic assumptions and over-optimism.

As stated by Phongpaibul & Aroonvatanaporn (2014) the cost estimation of software development projects in government tends to be biased, inaccurate and unjustified exceedingly. Hence, it could happen the corruption which would give the great impact on the country especially the economic growth. The inaccuracy of the software cost estimation can due to a number of reasons which are project uncertainty, software sizing errors and inexperienced project manager. Furthermore, according to Rajkumar & Alagarsamy (2013), many uncertainty factors influence software cost estimation too which describe in Figure 2.2. Those factors can identify through the software estimation models so that it can assist the managers in predicting the cost of the project.

Singh and Dwivedi (2014) also stated that the success of a software project depends on various factors that are interrelated to each other. The success of a project is identified by fulfilling all the project requirements within the schedule and cost that has been estimated before. There is no warranty that technology or standard procedure for the development process that would prevent software project from failures, cost overrun, late delivery, logical errors and incorrect design. The most significant factor influences the success of a project is management ability and people who are involved in the project.

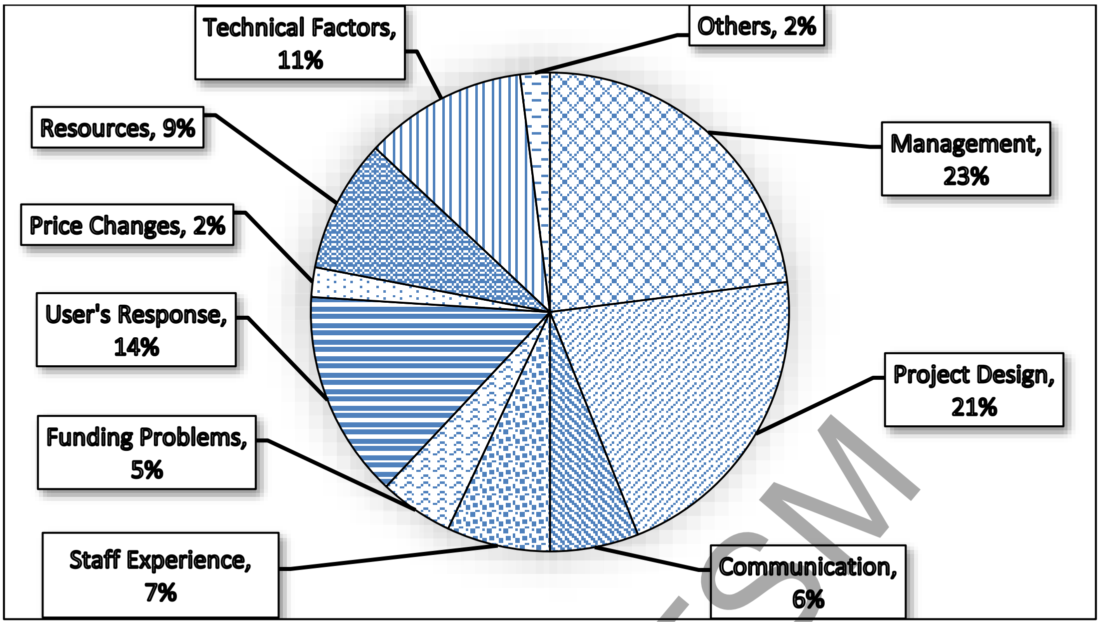

Figure 2.2 The major uncertainty factors affecting software cost estimation

Source: Rajkumar & Alagarsamy (2013)

Moreover, Haslinda, Azizah & Othman (2014) investigated the government ICT project failure in the Malaysian context. The failure was because ineffective cost estimation and the project was not fit the project requirements. The main cause was the organizational factors due to the estimation cost was not done correctly by the agencies or the project champions. It was influenced by the reduction of project cost by the government because of the economic downturn. As the impact, 16 of projects would be canceled before they ever accomplished, 53% of projects would be overbudget twice than the original estimates, and less than 31% of projects successful. Besides that, the estimation is performed at early phase of the project. According to Gumaei, Almaslukh & Tagoug (2015), the organization that estimates the software cost at early phases of a project tends to produce inaccurate results. However, it is necessary to estimate the software cost at early phases due to preparing the project proposal in order to get the budget. The estimation at early phase has high risk due to many uncertainties and risks on the scope and project requirements. As stated by Kumari & Pushkar (2013), there is no detailed information on the scope and requirements during the early phase of the software development. Zulkefli et al. (2011) also explain that the requirements of the project are essential so that the result accurate. Therefore, it is difficult to estimate the software cost at an early phase of the project because changes made on requirements. The estimation could be done in a later phase. The later phases refer to design, implementation and testing stage. Attarzadeh & Hock (2011) stated that the incomplete information on the software causes the difficulties in software cost estimation which lead to producing an inaccurate result.

Furthermore, lack of the historical government project records, as the impact the data availability is not effective and efficient regarding quantity and quality of data. It is caused by the data is not centralized, collected and maintained consistently. As the impact, the historical data is not effective to use estimating the cost (Raffo, Pfahl & Wang 2007). Historical data quality would affect estimation too. Thus, the information determines the success of the project due to incomplete and incorrect information would lead to the estimation failure (Ministry of Transportation and Infrastructure 2013; GAO 2009). So, the better quality the data are, the better-quality estimation would be.

Then, people who are lack experience in doing estimating the cost are often involved in the estimating process. For instance, United States Government Accountability Office (GAO) found the issue that happened for NASA which is lack of cost analysts' skills. The person who performed the estimation was a budget specialist who has a responsibility to manage funds. Cost analysts are the one who supposed to make the cost prediction because they are responsible for facilitating the financial services to control the project so that the project would be on track (GAO 2009). Cost evaluation is often calculated by people who are inexperienced using the estimating methods and tools. Although they are good in financial and accounting methods, it does not mean they are good in software estimating because it is required specific skills and training (Jensen 2003). According to GAO (2009), the cost estimator has to be multi-talented in analyzing high-quality data. Cost estimation is a difficult task, but yet it is very important. It takes time to develop which cannot be done in a hurry. Hence, the cost analyst needs to be well-trained and have experience because they are not only estimating the cost but also to predict the risk that might occur. As a cost analyst, it requires having knowledge of different disciplines such as accounting, budgeting, computer sciences, economics, engineering, and mathematics which describes in Table 2.3. However, the individual who does not have skills as a cost analyst are often involved in estimating the cost.

Table 2.3 Disciplines and concept in Cost Analysis

Cost Analysis	
Disciplines	**Concept**
Economics	Break-even analysis
	Foreign exchange rates
	Industrial base analysis
	Inflation
	Labor agreements
	Present value analysis
	Budgeting
Budgeting	Budget appropriations
	Internal company (industry)
	Program specific
Engineering	Design
	Materials
	Performance parameters
	Production engineering
	Program development test
	Scheduling
	System integration
Computer science / Mathematics	Analysis of commercial models
	Analysis of proposals
	Development of cost estimating relationship
	Model development
	Programming
Statistics	Forecasting
	Learning curve applications
	Regression analysis
	Risk/uncertainty analysis
	Sensitivity analysis
Accounting	Cost data analysis
	Financial analysis
	Overhead analysis
	Proposal analysis
Interpersonal skills	Approach
	Estimate
	Knowledge
Public and government affairs	Appropriations process
	Auditors
	Legislative issues
	Outside factors

Source: GAO 2009

Also, in Finland, the ministry of justice was developed prison information system. The project was huge overrun in terms of schedule and budget. They have to pay four times more than the "original fix price." The project was unsuccessful due to the users does not participate in the development process (ISBG 2004).

Project pricing does not involve the management only, but also the customer who will involve in project development (Rajkumar & Alagarsamy (2013).

Likewise, good cost estimation would also represent the project manager ability (Zulkefli et al. 2016). Besides that, top management commitment is very important for the sustainability of the project because the management is the one who responsible to manage the whole project (Rajkumar & Alagarsamy (2013). As stated by, Renny et al. (2015), number of failures in IT project is high caused by several factors which include less of support by top management, lack of user involvement, the objectives of project unclear and the maturity of organizational.

The requirements quality is able to influence the quantity of effort required to accomplish the project. Thus, it also affects the expenditures required to finish the project. For instance, the requirement of security system level is changed. As a result, the project requires more cost, effort, and resources (Boehm 2017). Therefore, gathering data and information is important to determine the resources of the project. The project design is associated with the project planning and project variables which are significant to identify the requirements and outcome of the project (Rajkumar & Alagarsamy (2013).

Furthermore, selecting the tool and technique to perform software cost estimation is substantial for cost estimation accuracy. Therefore, it is important to decide the right proper tool that uses for estimating the software cost. However, the common traditional tool used are Microsoft Excel and Microsoft project which has challenges in terms of the accuracy (Rajkumar & Alagarsamy (2013). Zulkefli et al. (2016) stated that most of the project managers are used the manual method to perform the calculation due to unavailability of computerized tools. The tools that usually selected for performing the calculation are EVM, cash flow statement, WBS statement burndown charts, and Gantt charts by using the Microsoft Excel. They also used HP quality center which is an automation method. Unfortunately, it does not have all the functionality that supports cost estimation. The availability of computerized tool is essential for a project which can increase the effectiveness and the efficiency in managing the project cost.

Other difficulties in software cost estimation are novel application software, changing technology and lack of homogeneity of project experience. The novel application of software means each software has its own uniqueness which causes the uncertainty in estimation. The software can be implemented in different environments, but at the same time, it needs to line with technological development. As a result, it will have different project estimates. Lastly, lack of homogeneity of project experience means the software cost estimation should be done based on the previous information of a project. in this case, it requires experiences on the similar previous projects (Suharjito & Budi 2013).

Likewise, lack of communications also affects the software cost estimation because cause misunderstanding and conflict in the project. As the impact, the project may delay or fail (Rajkumar & Alagarsamy (2013). Furthermore, the experience, knowledge, skills, and commitment of staff also influence the software cost estimation since they have an important role also in a development process. Other factors are the financial issues and user responses. The financial issues contribute to the project completion because the budget constraints can make the project delay. The user responses mean how the users react toward a new system because most of the end users do not have the training to use the new system and it is difficult for them to adapt themselves to using that system (Rajkumar & Alagarsamy (2013).

Hence, there is many factors influence software cost estimation in public sectors. The factors are listed in Table 2.4.

Table 2.4 Factors influence software cost estimation

Factors	**References**
Data availability	Zulkefli et. al (2016)
Data quantity	Kumari & Pushkar (2013)
Unrealistic assumptions	Ministry of Transportation and infrastructure (2013)
Fewer details design specification	
Project complexity	Leena (2012)
Product size	GAO (2009)
Available time	ISBG (2004)
Level of technology	
Lack of cost estimator experience	
Historical data quality	
Lack of user involvement	
Insufficient requirements	

to be continued…

…a continuation

Lack of executive support Developer incompetence Software development method Detailed documentation availability Risk analysis conducted Historical data Well trained and experience analyst Adequate budget Adequate cost reserve Unrealistic assumptions Over-optimism	GAO (2009)
Top management commitment Project complexity The inexperience of cost estimator	Ubani et. al (2015)
Project uncertainty Software sizing errors Inexperience project manager	Phongpaibul & Arronvanaporn (2014)
Management Project design Communication Staff experience Funding problems User's response Price changes Resources Technical factors	Rajkumar & Alagarsamy (2013)
Project management ability Project team	Singh & Dwivedi (2014)
Organizational factors	Haslindah, Azizah & Othman (2014)
Less of support by top management Lack of user involvement The objectives of the project unclear The maturity organizational	Renny et.al (2015)
Lack of detail information on the scope and requirements	Kumari & Pushkar (2013), Zulkefli (2011), Attarzadeh & Hock (2011)
Historical data	Raffo, Pfahl & Wang (2007)
Inexperienced cost analyst	Jensen (2003)
Lack of user involvement	ISBG (2004)
Project manager ability Tools to perform the calculation	Zulkefli et. al (2016)
Requirements quality	Boehm (2017)
Lack of homogeneity of project experience	Suharjito & Budi (2013)

Based on the table, the factors of the software cost estimation have an important role in determined the accuracy result of the estimation. Therefore, it is important to evaluate and analyze the factors as part of preparation before performing the estimation. The purpose is to detect the influence factors that might cause the actual cost does not achieve the estimation cost because no matter whatever the method or technique used to estimate the cost, those factors still have a significant impact towards the project sustainability. Moreover, the cost estimation consists of three steps which are estimation preparation, estimation creation and estimation management (see Table 2.1). The important aspect is preparing the estimation is a project document, historical project information and also constraints and assumptions. Thus, there is a need to analyze the factors so that it can create a better assumption towards the project and estimation. Then, the second phase is estimation creation which influences by the environmental factors and organizational process assets. Lastly, management of estimation is effect by the estimation tools organizational process and communication. Hence, the factors in Table give a great impact on the estimation process.

2.3.4 Software Cost Estimation in Indonesia

Indonesia spending on information technology has been increased by 15% in 2015 which is around 176.3 trillion IDR. One of the expenditures is for a software development project (BMI 2015). Before implementing a project, each government institution requires preparing the budget plan (*Penyusunan Rencana Anggaran Biaya / RAB*[2]). Hence, the first stage needs to be done estimating the project cost that requires project development (Adi Nugroho et al. 2009). According to the Presidential Regulation of the Republic Indonesia number 70 of 2012 on procurement good or services, the government agencies have to make its cost estimation *(Harga Perkiraan Sendiri*[3]*)* (Subsection 66 number (5) item a). The purpose of the cost estimation is to ensure the project cost within reasonable cost (Ziyad et al. 2014).

Based on the Presidential Regulation No. 4 of 2015, every procurement process of good and services that conduct by public sectors and private sectors of Indonesia must make the owner estimate cost (OEC) or *Harga Perkiraan Sendiri* (HPS) (INKINDO 2017). According to the Presidential Regulation no. 70 of 2012, the software considers as a good which require owner estimate cost in the planning stage. Before estimating the owner estimate cost, the Committing Officer (CO) requires

---

[2]*Rencana Anggara Biaya (RAB)* is budget estimate plan on the activities of the institutions in Indonesia

[3]*Harga Perkiraan Sendiri (HPS)* is the estimation of price goods and services of a project which use as the highest bidding limit offered by consultants or software provider.

identifying the requirements and specification. Unfortunately, the OEC is very difficult for the software cost estimation due to there are no standard techniques for reference and guidance (Sholiq et al. 2016). The OEC is used to estimate the cost and state the technical specification of the project. However, most of the private sectors and the public sectors are difficult to make a request or to offer the software development project due to facing difficulties in performing the own price estimate (OEC). If the estimation is too high for the fair price, the companies or the government will be potential losses. While if the estimation lower than the fair price, then it will have the potential for the software procurement failures due to the software providers will not be interested in doing the project (Imam & Arry 2015).

2.3.5 Software Cost Estimation Metrics

Software metric is a software measurement that can use for planning and control the software development project specifically for the software project cost estimation (Malathi et al. 2012). There are many types of cost estimation metrics such as size metrics, function point metrics, object point and use case point (Borade & Khalkar. 2013). However, the common size metrics frequently use are lines of code and function point analysis. Estimating the software size is not an easy task because it requires specific knowledge on the system functionalities such as scope, complexity and interactions (Matson et al 1994).

a. Function Point Oriented Metrics

The function point was published in 1979 by IBM which was introduced by Allan Albrecht. It can use for software cost estimation and also for controlling the productivity level by managers (Ramesh & Reddy 2016; Shekar & Kumar 2016; Patil, Badjate & Joshi 2014). Patil, Badjate & Joshi (2014) identified the factors of the estimation which is shown in Figure 2.4. As stated by Matson et al (1994), function point measurement is estimating the software size based on the functionalities of the system. The characteristics of the function point metric are as follows:

1. The tools, methodologies, programming languages, database management systems, processing hardware or any data processing technology do not take into account.
2. It is estimated based on the scope and requirements of the project.
3. The non-technical people are able to understand the function point measurement due to it relates to the external view of the system.

The formula of the function point is (see Figure 2.3):

$$FP = FC \quad (PCA)$$

$$PCA = 0.65 + 0.01 \sum_{i=1}^{14} c_i,$$

Figure 2.3 Function Point Formula

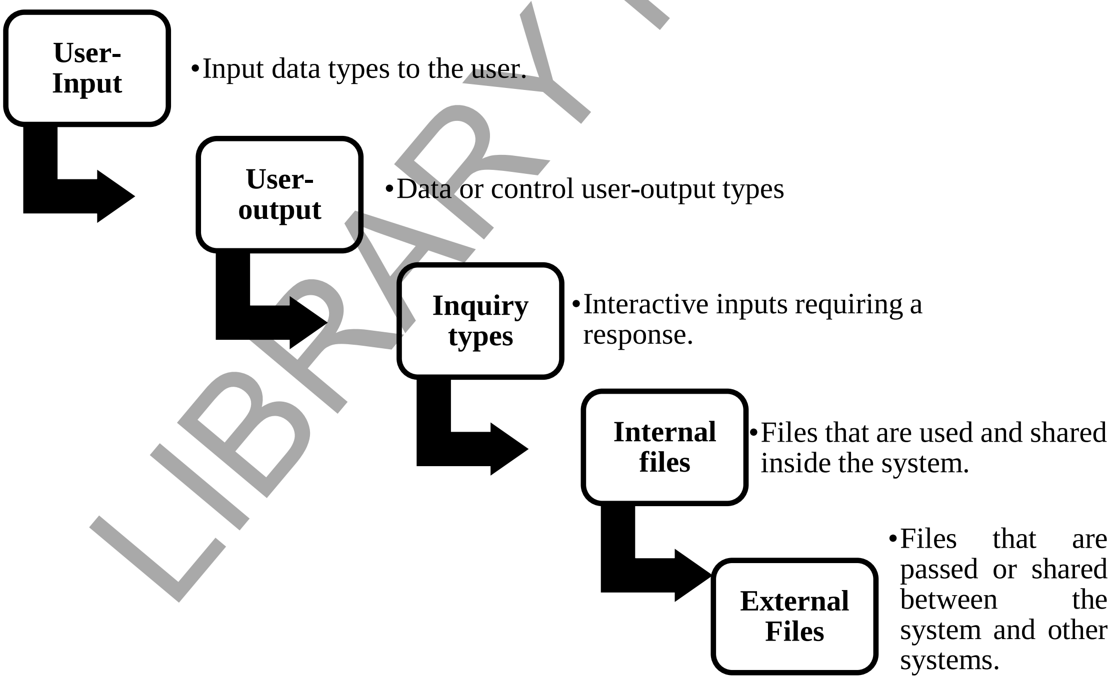

Figure 2.4 The factors of cost estimation by using Function point technique

Source: Patil, Badjate & Joshi (2014)

Kmerer & Zmud (1987) & Borade & Khalkar (2013) identified that FC refers to previously computed function counts and FP is function points). PCA stands for processing complexity adjustment whereby it is $0.65 \leq PCA \leq 1.35$, while c_i is complexity factors ($0 \leq c_i \leq 5$) of functional complexity (see Table 2.5) which be rated on the following scale:

- 0 = no influence
- 1 = incidental
- 2 = moderate
- 3 = average
- 4 = significant
- 5 = essential

Hence, implementing the function metric for measurement should consider its advantages and disadvantages which is shown in Table 2.6.

Table 2.5 Weight Value by Functional Complexity

Sr.No	General System Characteristics (GSC)	Degree of Influence (0-5)
1	Data Communications	-
2	Distributed Data Processing	-
3	Performance	-
4	Heavily used Configuration	-
5	Transaction Rate	-
6	Online Data Entry	-
7	End-User Efficiency	-
8	Online Update	-
9	Complex Processing	-
10	Reusability	-
11	Installation Ease	-
12	Operational Ease	-
13	Multiple Sites	-
14	Facilitate Change	-
	Total Degree of Influence (TDI)	-----

Source: Borade & Khalkar (2013)

Table 2.6 Advantages and Disadvantages Function Point Metric

Function Point Technique	
Advantages	**Disadvantages**
Its results are better than SLOC	Mechanization is hard to do.
Language free	The output quality does not consider.
They are not dependent on the language, tools, and methods of implementation.	It needs subjective evaluations with a lot of judgment involved.
Development costs can be estimated in the early phases of software development.	Many efforts and cost models are based on LOC, so function points need to be converted.
It can be estimated from requirements specifications.	Fewer research data is available on function points as compared to LOC.
	It is performed after the creation of design specifications.
	It requires manual work which is more time-consuming.
	Difficult for a new developer to estimate the size of the software as function point usage requires experience.

Source: Ramesh & Reddy (2016); Shekar & Kumar (2016); Patil, Badjate & Joshi (2014); Rajkumar & Alagarsamy (2013); Khatibi & Jawawi (2010)

b. Size Oriented Metrics (Lines of Code/LOC).

Size oriented metrics is a very important factor that influences successful a project because it uses to determine the cost, schedule and effort estimation. Therefore, software sizing error can lead to failure of the project such as cost overruns, schedule slip and the quality problem (Malathi et al. 2012). The example of size metric is SLOC which stands for source line of code. Source line of code (SLOC) consists of commands and data definition. Moreover, SLOC is easy to compute line of code because it has a scope for automation of counting the lines of code. However, SLOC Measuring seems very difficult in the early stages of the project because of the lack of information about requirements (see Table 2.7) (Patil, Badjate & Joshi 2014). It measures the size of software through the number of lines in the source code program which excludes the blank lines, comment lines, and library (Borade & Khalkar 2013).

The formula of the line of codes is shown in Figure 2.5.

$E = A + B \times (KLOC)^{C.}$

E is the estimated effort in man-months;

A. B. and C are constants.

KLOC is the estimated number of line of code

Figure 2.5 Line of Code (LOC) Formula

Source: Matson et.al (1994)

According to Matson et.al (2004), many issues of using LOC in measuring the software size. First, many definitions in defining the lines of code, as the impact, many variations of counting the lines of code (LOC) such as some models include the comment lines, while other models do not calculate it. Second, it is language dependence which means that it depends on the programming language used for the software development. For instance, the time required for the software development by using high-level programming is more than a low-level programming language. Then, it is difficult to estimate the lines of codes based on the information in the project requirements. Besides that, the historical data of the project is significant for LOC measurement. Lastly, the coding is only 10% to 15% of the total effort and it is very questionable whether the total effort in line with the numbers of lines of code (LOC).

Table 2.7 Advantages and disadvantages SLOC metric

Advantages	**Disadvantages**
SLOC is easy to compute line of code because it has a scope for automation of counting the lines of code. The KDLOC (1000 Lines of code) can be used to estimate a complex project.	SLOC Measuring seems very difficult in the early stages of the project because of the lack of information about requirements.

Source: Patil, Badjate & Joshi (2014); Malathi et al (2012)

c. Use Case Points (UCP) Oriented Metric

Use case points was introduced by Karner in 1993 which used to calculate the software size based on a use case diagram (Nassif et al. 2013). The use case points use to estimate the size and effort of the project from its use cases (Eberendu 2014). The formula of use case (see Figure 2.6).

UCP = TCF * ECF * UUCP * PF

TCF = Technical Complexity Factor

ECF = Environment Complexity Factor

UUCP = Unadjusted Use Case Points

PF = Productivity Factor

Figure 2.6 Use Case Formula

d. Object Points Oriented Metric

Malathi et al. (2012) define the object point as the measurement of the software size based the quantity of complexity of the following object:

- The number of separate screens that are displayed
- The number of reports that are produced by the system
- The number of program modules that must be developed

2.3.6 Software Cost Estimation Techniques

There are many techniques of cost estimation for software development. These techniques are divided into two categories which are: non-algorithmic technique and algorithmic technique (Shekar & Kumar (2016); Patil, Badjate & Joshi (2014); Rajkumar & Alagarsamy (2013); Heemstra (1992)). Each cost estimation model consists of the cost-drivers' parameters such as personnel competencies, experience using the techniques, domain and technology and project team cohesion levels (Boehm 2017).

a. Non-Algorithmic Methods

Non-algorithmic techniques are done based on the analysis of the previous project through analogy and assumption (Shekar & Kumar 2016). Ramesh & Reddy (2016) determined the estimating constraints which are relied on the past project such as:

- If the records of the previous project do not document it well, so when the estimator wants to predict the cost, it will be difficult to estimate it because of lack of historical data.

- If the proposed project is a project that is never done before, this technique cannot be used to perform the calculation since it depends on the data of the previous project.

i. Estimation by Analogy

This technique relies on the historical data of previous projects. After that, the project of historical data is compared with the project that is being proposed to make the analogy for the estimation (Ramesh & Reddy 2016; Shekar & Kumar 2016; Eberendu 2014;Patil, Badjate & Joshi (2014); Heemstra 1992). Moreover, the technique of the estimation is straightforward. However, it needs to extrapolate the actual data from the previous project to predict the cost. This method is also known as a systematic form of expert judgment since they will make decision-based on the analogous condition (Ramesh & Reddy 2016).

Eberendu (2014) identified that this technique is required of expert analysts to find out about the previous project cost drivers such as size, team experience, technology in use and application domain. However, Eberendu (2014); Ramesh & Reddy (2016); Lee, Titchkosky & Bowen (2002) stated that the issue of this technique is the previous project represent of the proposed project or not. Similarities of the project would be affected by the result of estimation. If the previous project is not similar to the project being proposed, it will lead to inaccurate estimation (see Table 2.7). The steps of estimation by using analogy technique refers to Figure 2.7.

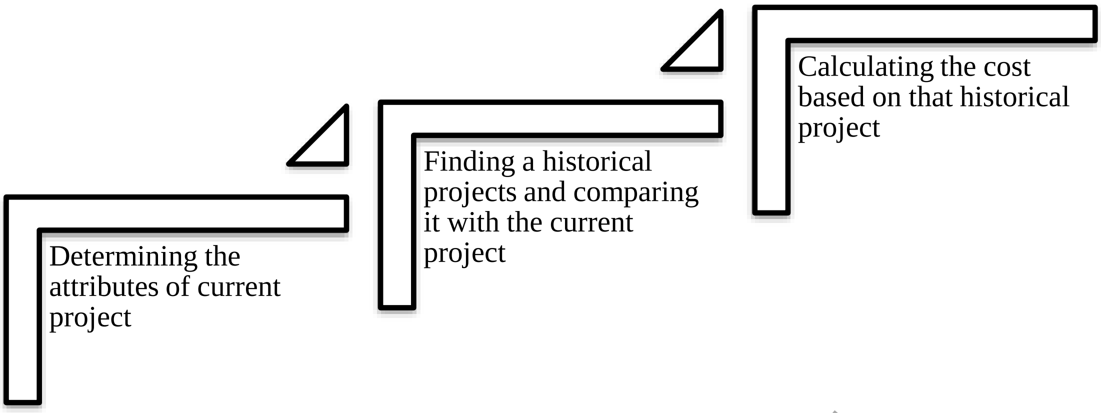

Figure 2.7 The process of estimation by using analogy technique

Source: Shekar & Kumar (2016); Patil, Badjate & Joshi (2014)

Table 2.8 Advantages and disadvantages of estimation by analogy technique

Estimation by Analogy	
Advantages	**Disadvantages**
1. Based on actual project data and experience.	1. Similar projects may not exist.
2. Having especial expert is not important.	2. Historical data may not be accurate.
3. Estimators experience can be used which helps in arriving at a better cost estimate.	3. A lot of information about past projects is required.
4. Easy to find a distinction between the previously completed projects and our current projects and this in a way also helps in knowing their impacts.	4. Must determine how best to describe projects.
	5. The choice of variables must be restricted to information that is available at the point that the prediction required.
	6. Have to determine the similarity and how much confidence can place in the analogies.
	7. Not applicable to every project

Source: Rajkumar & Alagarsamy (2013); Khatibi & Jawawi (2010); Ramesh & Reddy (2016); Shekar & Kumar (2016); Patil, Badjate & Joshi (2014)

ii. Expert Judgment

Expert judgment is the technique based on an expert's perception based on knowledge and experience from a similar project in estimating the cost (Shekar & Kumar (2016); Ramesh & Reddy (2016); Eberendu (2014); Rajkumar & Alagarsamy (2013). According to Lee, Titchkosky & Bowen (2002), The estimation is mostly qualitative; as the impact, it is hard to document the factors that are used by the experts. The outcome of the estimation is just as good as the expert opinions. It is difficult to sustain the knowledge and experience that is done by experts (Heemstra 1992). This method is good to use when lack of gathering and finding the data. This can be done through work breakdown structure (Shekar & Kumar 2016).

There are advantages and disadvantages to this technique which is described in Table 2.9.

Table 2.9 Advantages and disadvantages of expert judgment technique

Expert Judgment	
Advantages	**Disadvantages**
1. Expert with the relevant experience can provide a good estimation 2. Fast estimation 3. Adapt to special projects **4.** The impacts caused due to new technologies, architecture and languages can be predicted by the experts.	1. Dependent on the expert. 2. Usually, is done incompletely. 3. It's hard to document the factors used by the experts or experts group. 4. An expert might be some biased, optimistic and pessimistic. 5. This method cannot be quantified. 6. It always compliments to other cost estimating the model

Source: Ramesh & Reddy (2016); Shekar & Kumar (2016); Patil, Badjate & Joshi (2014); Rajkumar & Alagarsamy (2013); Khatibi & Jawawi (2010)

iii. Top-Down

The top-down technique is derived from the global view product into many kinds of partition in low-level components (Shekar & Kumar 2016; Ramesh & Reddy 2016; Lee, Titchkosky & Bowen 2002). The top-down technique also is known as "Macro Model. This method was suitable for the early stage of software development due to the information detailed was unavailable. The features of the model also exist in Putnam's model which is the cost-time trade (Ramesh & Reddy 2016; Shekar & Kumar 2016). Likewise, there are advantages and disadvantages of Top-Down technique which is shown in Table 2.10.

Table 2.10 Advantages and disadvantages of the Top-Down technique

Top-Down Technique	
Advantages	**Disadvantages**
1. Faster and easier than bottom-up method 2. It requires very fewer details about the project. **3.** This technique focuses on activities like integration, management, etc.	1. Less accurate than other methods 2. It often does not identify difficult low-level problems that are likely to escalate costs and sometimes tends to overlook low-level components which can increase the cost. 3. It provides no detailed basis for justifying decisions or estimates.

Source: Ramesh & Reddy (2016); Shekar & Kumar (2016); Rajkumar & Alagarsamy (2013); Khatibi & Jawawi (2010)

iv. Bottom-Up

"Bottom-Up," the estimation is done by calculating each software component cost. Then, sum up them to get the overall cost estimation (Shekar & Kumar 2016; Ramesh & Reddy 2016; Lee, Titchkosky & Bowen 2002). Therefore, the project team required splitting the work into small components, so that the process estimation would be easier. Unfortunately, the accuracy of the estimation result was 20%, it is either overrun 20% or underrun 20% (Eberendu,2014). Hence, it is done based on the detailed analysis. On the other hand, many disadvantages should take into consideration which describes in Table 2.11.

Table 2.11Advantages and Disadvantages Bottom-Up Technique

Bottom-Up Technique	
Advantages	**Disadvantages**
Based on detailed analysis This technique is more stable. It takes more time	Less accurate than other methods because the necessary information may not be available in the early phase. It tends to be more time-consuming. It may not be feasible when either time or personnel are limited. It may overlook many of the system-level costs (integration, configuration management, quality assurance, etc.) associated with software development

Source: Ramesh & Reddy (2016); Shekar & Kumar (2016); Rajkumar & Alagarsamy (2013); Khatibi & Jawawi (2010)

v. Artificial Neural Networks

The artificial neural network is an estimation technique which is trained by using historical data (Shekar & Kumar 2016). Patil, Badjate & Joshi (2014) defined that as a technique which uses node that is interrelated to each other like biological neural networks. The factors of the neural network are:

- Interconnection nodes
- The learning process for updating the weights of nodes
- Activation function which converts the input and output.

Furthermore, the model of neural networks has some functionality like a human brain. The good estimate does not only consider function point, but also various aspects that are associated with estimation such as development environment. While the COCOMO model might difficult to produce an accurate result due to it depends on the restricted data (Kaur & Salaria 2012). So, there are advantages and disadvantages of using the artificial neural network for estimation as shown in Table 2.12.

Table 2.12 Advantages and Disadvantages Neural Networks Technique

Neural Networks Technique	
Advantages	**Disadvantages**
1. Consistent with unlike databases 2. Power of reasoning **3.** Can give accurate results due to the training of the network process.	1. There is no guideline for designing 2. The performance depends on a large amount of training data 3. Complexity increases due to learning of parameters.

Source: Shekar & Kumar (2016); Patil, Badjate & Joshi (2014); Rajkumar & Alagarsamy (2013); Khatibi & Jawawi (2010)

vi. Delphi

Delphi technique gathers all the experts in the meeting to find correct information from the discussions. This technique works by using expert judgment technique. The steps of this cost estimation technique are referred to Figure 2.8. According to Rajkumar & Alagarsamy (2013) & Khatibi & Jawawi (2010), the advantages of estimating by using the Delphi technique is able to share information. However, it is more time consuming than the nominal group process.

Steps of estimation by using Delphi technique:	The project manager gives specification and estimation form of project to the experts.
	The coordinator calls the experts to attend the meeting to discuss about estimation issues.
	The project manager prepares and distributes estimation summary in the forms.
	Coordinator again calls meeting with experts discuss points where their estimates varied widely.
	Experts fill the form and above steps iterate for many times as accurate estimate occurs.

Figure 2.8 Cost estimation steps of Delphi technique

Source: Patil, Badjate & Joshi (2014); Tailor, Saini & Rijwani. (2014); Kumari & Pushkar (2013);

vii. Price to Win

Price to win the estimation technique is the price which is essential to win the project. The focused of this technique is the budget of the customer, which has been stated in the proposal, rather than the software functionalities (see Table 2.13) (Shekar & Kumar 2016; Lee, Titchkosky & Bowen 2002). The advantages of this technique are time and money run out before the project completed. Heemstra (1992) identified that "Price to Win" technique is hard to say as software cost estimation technique due to it is more influenced by the commercial motive to win the project with a certain price. Therefore, it takes the software developer into a risk because they may suffer because they are not able to gain much profit.

Table 2.13 Advantages and Disadvantages Price to Win Technique

Price to Win Technique	
Advantages	**Disadvantages**
Cost is estimated based on the customer budget.	This method may lead to a delay in delivery of the software project, due to which software developers may suffer loss.

Source: Shekar & Kumar (2016)

viii. Fuzzy Logic

Fuzzy logic is also known as soft computing techniques which can solve the estimation problem when mathematical models cannot be formed (Shekar & Kumar 2016). The steps of the fuzzy technique are described in Figure 2.9.

Fuzzification	• It converts a crisp input into a fuzzy set.
Fuzzy rule bases	• Fuzzy logic system use fuzzy IF-THEN rules.
Fuzzy Inference Engine	• Access fuzzy rule base to generate the output.
Defuzzifier	• It converts the fuzzy output into crisp output

Figure 2.9 Steps of Fuzzy Logic Technique

Source: Patil, Badjate & Joshi (2014)

Moreover, the fuzzy logic technique is flexible, but it is hard to use due to the process is more complex. The advantages and disadvantages of the fuzzy technique are shown in Table 2.14.

Table 2.14 Advantages and Disadvantages Fuzzy Technique

Fuzzy Technique	
Advantages	**Disadvantages**
1. Training is not required 2. Flexibility **3.** Provide reliable estimates	1. Hard to use. 2. Maintaining the degree of meaningfulness is difficult. 3. Cost estimation of complex features is tedious. 4. Complexity increases due to the complex process of fuzzy logic.

Source: Shekar & Kumar (2016); Rajkumar & Alagarsamy (2013); Khatibi & Jawawi (2010)

b. Algorithmic Cost Modelling

Algorithmic cost modelling is done by using the mathematical equations which has the following inputs such as SLOC, function points and other cost drivers (language, design methodology, skill-levels, risk assessments, etc.) (Shekar & Kumar 2016;

Ramesh and Reddy 2016; Patil, Badjate & Joshi (2014); Lee, Titchkosky & Bowen 2002). Algorithmic model is also known as the parametric model. Examples of this techniques are COCOMO (Constructive Cost Model), COCOMO II and Putnam's software life-cycle model (SLIM) (Lee, Titchkosky & Bowen 2002). The algorithmic cost modeling also consider the cost factors which describe in Figure 2.11. Patil, Badjate & Joshi (2014) stated that each algorithmic model uses a formula to calculate the estimation (see Figure 2.10).

$$\textbf{Effort} = f\,(x1, x2, x3, \ldots, x_n)$$

$$(x1, x2, x3, \ldots, x_n)\text{: The cost factors}$$

Figure 2.10 The Algorithmic Model Formula

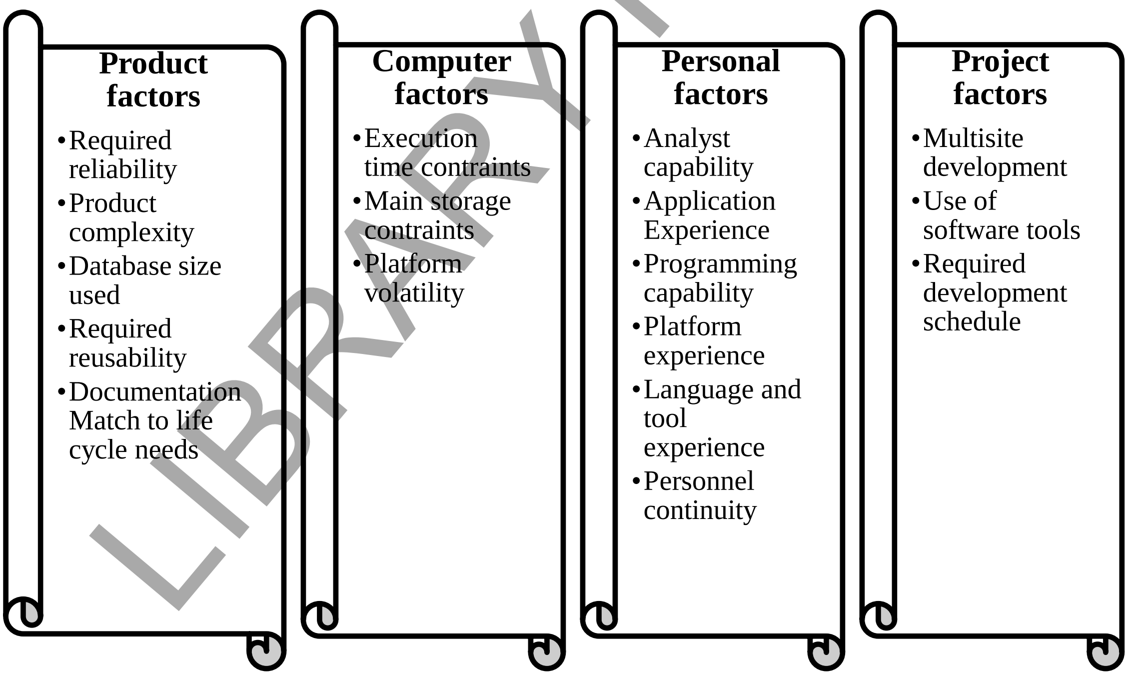

Figure 2.11 The cost factors of Algorithmic Method

Source: Patil, Badjate & Joshi (2014)

There are advantages and disadvantages of the algorithmic method which are described in Table 2.15.

Table 2.15 Advantages and Disadvantages of Algorithmic Method

Algorithmic Models	
Advantages	**Disadvantages**
1. Repeatable estimations are possible. **2.** Easy to modify input data, refine and customize formulas	1. It is unable to deal with exceptional conditions, such as exceptional personnel in any software cost estimating exercises, exceptional teamwork, and an exceptional match between skill-levels and tasks. 2. Poor sizing inputs and inaccurate cost driver rating will result in inaccurate estimation. 3. Some experience and factors cannot be easily quantified.

Source: Ramesh & Reddy (2016); Patil, Badjate & Joshi (2014)

c. Software Cost Estimation Model

In the last decades, there are many models have been developed such as COCOMO I, COCOMO II, Putnam, and Estimacs. However, a software manager faces the problem in selecting the model due to each model has different inputs which will give impact to the output of software estimation.

i. Putnam's Software Life-cycle Model (SLIM)

Software life-cycle management model was introduced by Lawrence H. Putnam in 1978. This model is a quantitative software effort estimation model that states the time and effort needed for the specific size of the project (Suri & Ranjan 2012). This technique is based on the manpower distribution which is also known as an empirical software effort estimation model (Shekar & Kumar 2016). The formula of the model is shown in Figure 2.12.

Technical Constant C = size*$B^{1/3}$*$T^{4/3}$

Total Person Months B=1/T^4*(size/C)^3

T = Required Development Time in years

Size = Estimated in LOC

Figure 2.12 Putnam's SLIM Formula

Source: Ramesh & Reddy (2016); Shekar & Kumar (2016)

B is the total effort that is required to complete the project, while T is the time needed to complete the project. Size is shown the progress of programmer which can calculate by using the historical data (Suri & Ranjan 2012). Moreover, based on Pasha & Atique (2010) SLIM is an empirical software cost estimation model that depends on SLOC in the estimation process. SLIM model also like COCOMO in terms of their performance. As described by other previous researchers, there is a need to calibrate the parameters before using it. Otherwise, the performance of the model is not good due to the parameter influenced by the software development environment. According to Ramesh & Reddy (2016), based on the Kemerer's research, the error of the result estimation by using Putnam SLIM is 772.87%. It is because the great uncertainty in the software size. Hence, the strengths and weaknesses of the Putnam model described in Table 2.16.

Table 2.16 Advantages and Disadvantages of Putnam Model

Putnam Model	
Advantages	**Disadvantages**
1. This model is based on two variables which are time and size.	1. There is often great uncertainty in the software size. 2. Putnam's Model does not take into account other aspects of the software development lifecycle.

Source: Ramesh & Reddy (2016)

ii. Estimacs

Estimacs model was developed by H. Rubin which required the users to answer 25 questions to find the complexity and size of the project. In fact, it consists of nine modules such as a function point module, a risk module and an effort module

(Tejaswini et al. 2015; Heemstra 1992; Pasha &Atique n.d). Estimaca model is a high-level model even though it does not provide the accurate results (Shandu & Salaria 2014). The critical estimation dimensions of ESTIMACS are described in Figure 2.13.

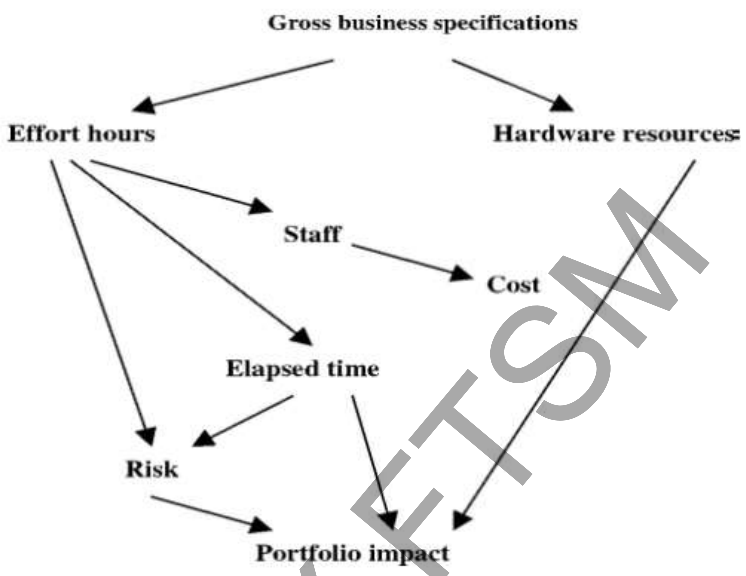

Figure 2.13 Rubin's map relationship of estimation dimensions

Source: Singhs (2014)

According to Pasha & Atique (2010), Howard Rubin has introduced the ESTIMACS model that use Function Point for measurement. Based on Rubin's map, it describes the importance of the growth business specification which use to estimate the efforts and resources of a project. all those dimensions influence the portfolio impact of the project.

The estimation dimension is related to the project factors that are shown in Table 2.17.

Table 2.17 The estimation dimension relation with the project factors

Estimation Dimension	Project Factors
Effort hours	• Customer complexity • Customer Geography • Developer familiarity • Business function size • Target system sophistication • Target system complexity

to be continued…

… a continuation

Staff/cost	• Effort hours • Staff productivity • Skill level development • Rate at each skill level
Hardware	• System category • Generic system type • Operating window • Transaction volume
Risk	• System size • Project structure • Target technology
Portfolio	• Resource needs of concurrent projects • Relative project risks

Source: Singhs (2014)

iii. COCOMO

COCOMO (Constructive Cost Model) is an algorithmic model that used to calculate the cost and time that is needed to accomplish the project. It was introduced by Barry Boehm in 1981. The parameters and equation depend on the previous project (Shekar & Kumar 2016, Patil, Badjate & Joshi (2014). The COCOMO model is a model that developed based on the historical data. This model takes the top management support and teamwork of project into account because the estimation result produced based on the man-months required to complete the project (Rosmala & Akbar 2010). Ramesh & Reddy (2016) defined that COCOMO is a regression model which influence by many factors that described in Figure 2.14.

On the other hand, the COCOMO, SLIM and Jensen model must be calibrated because the model is developed based on the historical project data which usually done in the USA. The software development environment from one place to another place is not the same. As the impact, the value used in the model is not the same since it is derived based on the historical projects in a particular environment (Heemstra et al. 1987).

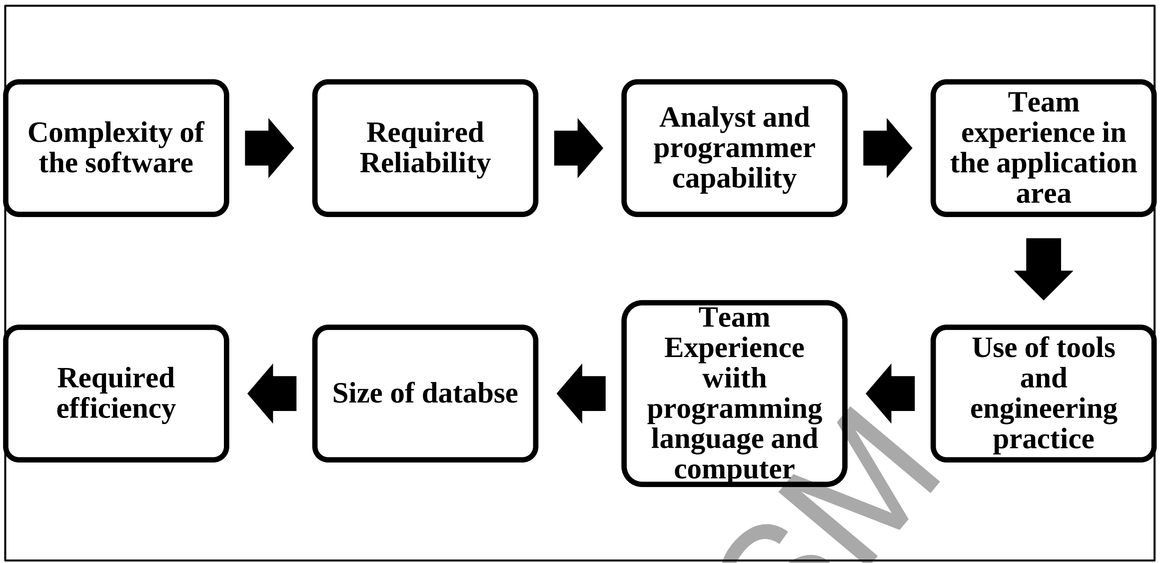

Figure 2.14 The factors that influence COCOMO model

Source: Ramesh & Reddy (2016)

According to Rosmala &Akbar (2010), Sheta & Aljahdali (2013) & Sadiq et al. (2013), there are three models of COCOMO that have been proposed by Boehm which are a basic model, intermediate model, and detailed COCOMO model. Suri & Ranjan (2012) stated that the basic COCOMO is used to estimate the software cost that needs to be done quickly. However, the accuracy of the model is limited because few factors take into account. Intermediate COCOMO includes the cost drivers in estimating the software cost, while the detailed COCOMO is estimated a project by considers the influence of individual project phases. The summary of the variety of COCOMO illustrates in Figure 2.16.

The general COCOMO model formula is described in the equation (see Figure 2.15) (Sheta & Aljahdali 2013 & Sadiq et al. 2013).

$$E = \lambda \times Size^{\mu} \times EAF$$

E	: Effort (man-months)
λ	: Calibrated constant
μ	: Size scale factor
Size	: Measured by KLOC (Thousand lines of code)
EAF	: The effort adjustment factor from cost multipliers

Figure 2.15 General COCOMO Equation

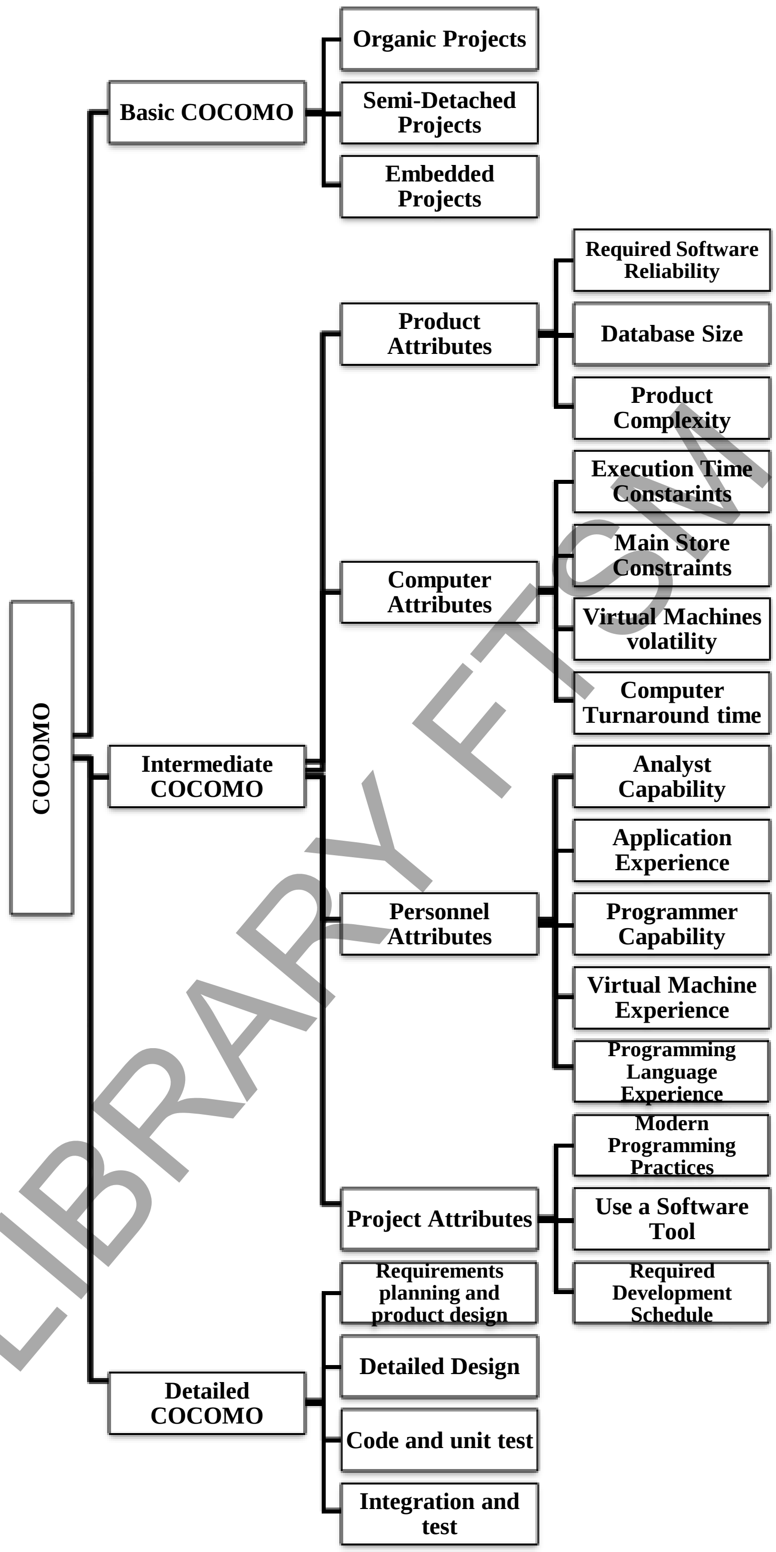

Figure 2.16 Types of COCOMO Summary

- **Basic model**

The basic model of COCOMO is divided into three types of project which are organic, semi-detached and embedded (see Table 2.18). The organic project is a project that is performed by small teams who have a good experience and familiar with the development environment. Besides that, the project size falls into a small size projects category because it has few project requirements. On the other hand, the embedded project is large projects that consist of many project requirements. The project has to be completed within the tight schedule, and the project size is over than 300 KLOC. Lastly, semi-detached is in between the organic and embedded of project type. The project size is from 50 to 300 KLOC (Suri & Ranjan 2012). According to Merlo et al. (2002), the development mode is divided based on the project characteristics such as size, innovation, deadline/constraints and development environment which shown in Table 2.19.

Table 2.18 Three types of the basic model COCOMO

Model Types	Effort (E)	Time (T)
Organic Model	$E=2.4(KLOC)^{1.05}$	$T=2.5(E)^{0.38}$
Semi-Detached Model	$E=3.0(KLOC)^{1.12}$	$T=2.5(E)^{0.35}$
Embedded Model	$E=3.6(KLOC)^{1.20}$	$T=2.5(E)^{0.32}$

Source: Suri & Ranjan (2012)

Table 2.19 Project characteristics of development modes

Development Mode	Project Characteristics			
	Size	**Innovation**	**Deadline/ Constraints**	**Dev Environment**
Organic	Size	Little	Not tight	Stable
Semi-detached	Medium	Medium	Medium	Medium
Embedded	Large	Greater	Tight	Complex hardware/customer interfaces

Source: Merlo et al. (2002)

- **Intermediate Model**

Suri & Ranjan (2012) explain that intermediate COCOMO is estimated based on the function of program size and the cost drivers which consists of 15 cost drivers. The cost drivers of the intermediate model are divided into to four categories which illustrate in Table 2.20 and the multipliers values for estimating the effort calculation are shown in Table 2.21. Albakri & Qureshi (2012) describes that the software cost estimation by using the intermediate model is predicted based on the 15 project cost drivers that have the multiplier value for each factor with a rating from very low to extra high.

Table 2.20 15Characteristics of Intermediate COCOMO Model

No	Product Characteristic	Hardware Characteristics	Personnel Characteristics	Project Attributes
1	Required Software Reliability	Runtime performance constraints	Analyst capability	Use of the software tools
2	Size of application database	Memory constraints	Software engineer capability	Application of software engineering methods
3	Complexity of the product	Volatility of the virtual machine environment	Application Experience	Required development schedule each of characteristic
4		Required turnaround time	Virtual machine experience	
5			Programming language experience	

Source: Kumari & Pushkar (2013)

Table 2.21 Cost Drivers Multipliers Value

Cost Drivers	Rating					
	Very Low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	-
DATA	-	0.94	1.00	1.08	1.16	-
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	-	-	1.00	1.11	1.30	1.66
STOR	-	-	1.00	1.06	1.21	1.56
VIRT	-	0.87	1.00	1.15	1.30	-
TURN	-	0.87	1.00	1.07	1.15	-
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	-
AEXP	1.29	1.13	1.00	0.91	0.82	-
PCAP	1.42	1.17	1.00	0.86	0.70	-
VEXP	1.21	1.10	1.00	0.90	-	-
LEXP	1.14	1.07	1.00	0.95	-	-
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	-
TOOL	1.24	1.10	1.00	0.91	0.83	-
SCED	1.24	1.08	1.00	1.04	1.10	-

Source: Albakri & Qureshi (2012)

- **Detailed COCOMO Model**

Pasha &Atique (2010) describes that the detailed or advanced COCOMO model is combination the basic and intermediate COCOMO model characteristics. It consists of four stages which are:

- Requirements planning and product design (RPD)
- Detailed design (DD)
- Code and unit test (CUT)
- Integration test (IT)

Those stages are part of the analyst capability cost driver. Each step has a multiplier value too which describe in Table 2.22.

Table 2.22 Multiplier value of analyst capability

Cost Driver	Rating	RPD	DD	CUT	IT
ACAP	Very Low	1.80	1.35	1.35	1.50
	Low	0.85	0.85	0.85	1.20
	Nominal	1.00	1.00	1.00	1.00
	High	0.75	0.90	0.90	0.85
	Very high	0.55	0.75	0.75	0.70

Source: Pasha & Atique (2010) & Merlo et al (2002)

- **COCOMO II**

COCOMO II was established in 1990 and required large of data. For the input, it requires source line of code (SLOC), function point and object points (Shekar & Kumar 2016). Kumari & Pushkar (2013) stated that the COCOMO II model consists of application composition model, early design model and post architecture mode. The general equation of the effort is (see Figure 2.17):

$$\text{Effort} = 2.9\ (\text{KLOC})^{1.10}$$

Figure 2.17 Effort COCOMO II Equation

COCOMO II model consists of three steps. The first step is to develop a prototype of the project. The purpose is to solve the high risk that might occurs due to the user interface, software and system interaction, performance, and technical maturity. This stage is estimated by using object points. Moreover, the second step is estimating the project size by using function point based on the project requirements. The last stage is started to develop the real of software development. The size is calculated according to the lines of code (LOC) (Dillibabu & Krishnaiah 2005). Zulkefli et al. (2010) explain that COCOMO II contains several factors, for instance, non-sequential and rapid development process models. The COCOMO II consists of Application composition model, early design and post architectures model. COCOMO II model can be used to estimate small or large software project, while Putnam SLIM model is suitable for the large project only. On the other hand, there is also advantages and disadvantages of using COCOMO II technique which is shown in Table 2.23.

Table 2.23 Advantages and Disadvantages COCOMO II technique

COCOMO II Technique	
Advantages	**Disadvantages**
1. COCOMO II proves to be an industry standard model. 2. It has a clear and effective calibration process.	1. Calculation of duration for small projects is unreasonable.

Source: Shekar & Kumar (2016)

Each model has its strengths and weaknesses. The Table 2.24 describes the advantages and disadvantages of using COCOMO model in estimating the software cost.

Table 2.24 Advantages and Disadvantages COCOMO Model

COCOMO Technique	
Advantages	**Disadvantages**
1. Clear results 2. Very common 3. Simple to estimate cost	1. Much data is required. 2. It is not suitable for any project in future. 3. COCOMO model ignores requirements and all documentation. 4. It ignores customer skills, cooperation, knowledge and other parameters. 5. It oversimplifies the impact of safety/security aspects. 6. Estimation at the early stage of software development leads to estimation failure. 7. It ignores hardware issues 8. It ignores personnel turnover levels 9. It is dependent on the amount of time spent in each phase.

Source: Shekar & Kumar (2016); Ramesh & Reddy (2016); Patil, Badjate & Joshi (2014); Rajkumar & Alagarsamy (2013); Khatibi & Jawawi (2010)

2.4 THE CONCEPTUAL MODEL

The purposed of the propose model is to assist the estimator in preparing software cost estimation in public sectors by determine the critical factors that significantly impact the software cost estimation in public sector. Therefore, the conceptual model is created based on the conceptual study and also the weaknesses of the existing model (see Table 2.16, Table 2.23 & Table2.24). According to the previous research on the software cost estimation in public sector, it can conclude that there are many factors influence of software cost estimation result in the public sector (see Table 2.4). Those factors are essential to estimate the software cost because the factors determine the result accuracy and the successful project.

Besides that, the software cost estimation is related with the four fundamentals of the software development process, especially the first phase of the software development process which is software specification. Hence, the software cost estimation could be done after the details design software specification is completed which means that the software functionality and constraints are identified based on the software development public sector criteria as mentioned by Medvedska & Berzisa (2015). It is important to highlight that the software development in public sectors are quite different with the software development process in private sectors. Furthermore, there are also many factors of influence the software development (see Table 2.1) which are almost similar with the factors influence of the software cost estimation. Thus, those factors are related to each other to ensure the sustainability of the project.

Moreover, as stated in the problem statement, the software cost estimation in government complicated due to the estimation conducted early in the first stage of the project. However, the scope and requirements are not identified clearly. As a result, it becomes issues in estimating due to the scope and requirements are fundamental in software cost estimation as it can be seen in the ESTIMACS model that was introduced by H. Rubin. It is also because of the lack of knowledge, skill, and experience of the project team which includes the project manager, top management, cost analyst, and developer. Furthermore, COCOMO is a common model chosen to estimate the software cost. On the other hand, it ignores user’s skill, documentation as well as cooperation and knowledge of the project team. Besides that, COCOMO and SLIM model use lines of codes (LOC) to measure the software size. It is difficult to implement it in the software project in the public sector due to the estimation perform in the early stage of the project.

The conceptual model consists of four main factors which are technology, people, process and organizational factors which is created based on the factors of algorithmic method such as product factors, computer factors, personal factors and project factors (see Figure 2.11). Hence, the main factors in the conceptual model expands the previous research on the algorithmic method factors of the software cost estimation. First, the technology factor is developed based on the project factors (cost algorithmic method factors) by adopting the used of software tool in estimation. Second, the people factor is derived from the personal factor (cost algorithmic method factors). Third, the people factor derives the personal factor (cost algorithmic method factors). Then, process factors are extended the product factors in the cost of the algorithmic method. Lastly, in the proposed model is added the organizational factors because based on the conceptual study the organizational factor also influences the software cost estimation. Furthermore, the sub-factors in the conceptual model is also some of the sub-factors in algorithmic method.

The sub-factors in the proposed model also derives from the conceptual study on the success factors of software development in public sector, software cost estimation in public sector, estimating cost phases, software cost estimation in Indonesia and the weaknesses of the existing model (SLIM, ESTIMAC and COCOMO). The factors of the conceptual model are described in Figure 2.18 and the details choosing factors influence of the software cost estimation in public sector explain in Table 2.25. So, the proposed model is important for estimation preparation to identify the factors that might influence the software cost estimation so that the project competed within the scope, quality, time and cost that has been estimated before the project begin.

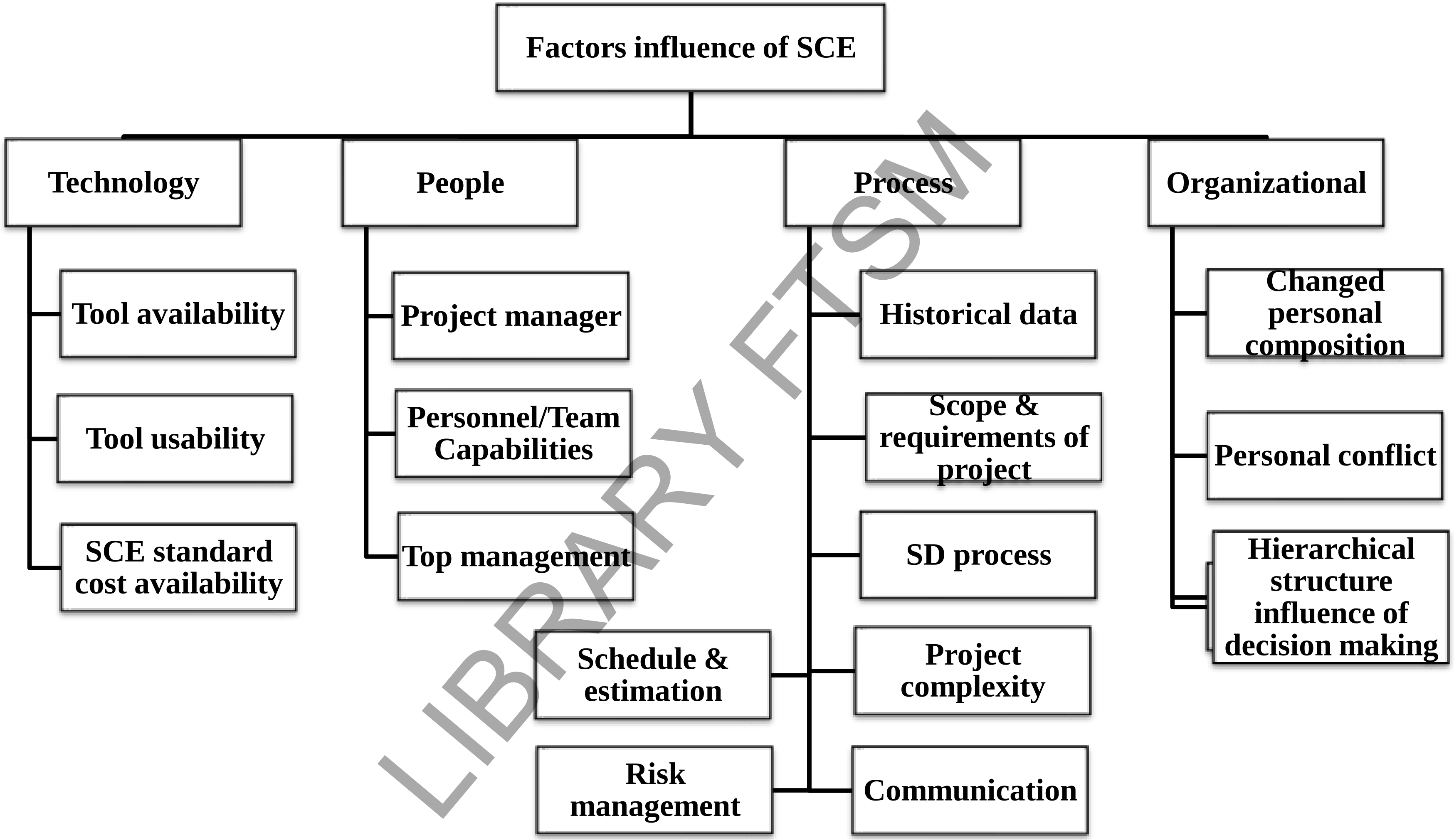

Figure 2.18 Conceptual Model

Table 2.25 Conceptual Model Details

No.	Sub-Factors	Description
	I. Technology	**Developed based on project factors (the cost factors of algorithmic method) which focus of the software cost estimation technique. This is because as stated in problem statement that there is no specification technique used to estimate the software cost in Indonesian public sectors. Thus, the government usually just rely on the experts or it is called as expert judgment technique.** **Hence, the sub-factors expand "the use of software tool) in project factors (Patil, Badjate, & Joshi 2014). In fact, in intermediate COCOMO model, one of the characteristics is "project attributes" that content "use of the software tools" as one of project cost drivers (see Table 2.17).**
No.	**Sub-Factors**	**Description**
1.	There is a standard used in estimating the cost.	The factor chosen is because based on the literature review on the software cost estimation in Indonesian public sector has to perform owner estimate cost (OEC) before the software cost estimation is conducted (The President Regulation of the Republic Indonesia number 70 of 2012). So, this factor is an additional for that rules to have a standard in estimating the cost so that it can use as guidance for perform the software cost estimation. Extend of the sub-factors of the "use of software tools" which include in the cost factors of algorithmic method and project cost drivers of intermediate COCOMO model (Patil, Badjate & Joshi (2014);Kumari & Pushkar (2013).
2.	Software cost estimation is done by using a proper tool.	
3.	There is tool can be used to perform the software cost estimation.	
	II. People	**Developed based on the personnel characteristics in intermediate COCOMO model and personal factors (cost algorithmic method factors). People refers to the one who involve in estimating cost and manage the project which have different roles in managing the project which are project manager, personnel/ team and top management.**
a.	**Project Manager**	The sub-factors are expanding the factors from existing model which are: a) ESTIMACS model. One of critical dimension in ESTIMACS is staff /cost that consists of staff productivity/skill development which include in project factors (Singhs 2014).
4.	The project manager is knowledgeable and competence in ICT	
5.	Project manager understood the scope and requirements of the project clearly.	
6.	Project manager is proficient in controlling the software development project	
c.	**Personnel / Team Capabilities**	
7.	The project team knows about the status of the project.	

to be continued…

…a continuation

8. The team size gives impact on accomplishing the project. 9. The man-power allocation is sufficient for the size of the project. 10. The experience of cost estimator. 11. Programmer capability significance towards the successful project 12. Lack of training for staff in software development project.	b) SLIM model. In order to estimate the software cost by using SLIM model, it requires to calculate the effort, while the size is calculated based on the programmer progress. Hence, it depends on the programmer capability (Ramesh & Reddy 2016; Shekar & Kumar 2016). c) COCOMO.
d. Top Management 13. Top management support is essential for the project to be successful. 14. Top management is understood the objectives of the project. 15. Top management is involved and committed to the project.	Analyst capability, software engineer capability, programming language experience are included in project cost drivers of intermediate COCOMO model. Those people involve as the personnel or project team (Kumari & Pushkar 2016). The first software development process is software specification which is vital for the SCE. The software specification depends on the people involve in the project such as the skills and competence of the employees in ICT, person month, team size and top management support (see Table). Based on the factors influence SCE in public sectors (see Table), the people involve in the project has a big impact on the SCE such as top management support and commitment, developer competency, experience cost analyst, project manager capability and project team. Furthermore, in the conceptual model, the sub-factors on the people factors are extended the factors that are stated in previous research and the existing models (SLIM, ESTIMAC, and COCOMO) which are more specific details.
III. Process	**Developed based on the factors influence of the SCE and also the factors include in the existing models. So, the sub-factors is more details than the product factors in cost factors of algorithmic method.**
a. Historical Data 16. The previous data project is important to estimate the software cost a new project. 17. The data of the previous projects are well documented.	In Table, the historical data influence of the software cost estimation, while in the conceptual model it is extended which focus on the data documentation and the important of the historical data. It can use as the reference to estimate the new project.
b. Scope and Requirements of Project 18. Scope and requirements defined clearly 19. Scope and requirements of project understood by all the project team. 20. All the requirements of the project are achieved. 21. The project objectives are achieved.	Scope and requirements include as the factor because the project comprise the scope and requirements of project which is important for the first software development process which is software specification details.

to be continued…

…a continuation

	Based on the table, inaccurate result of the software cost estimation is because fewer details design specification and insufficient requirement of project. Therefore, the scope and requirements still included in the conceptual model by extending the factors become four sub-factors.
c. Risk Management	
22. Potential risk identified from the project start. 23. Risk management has been included in project planning. 24. Risks that occurs during the software development project manage well.	Risk has been included in the estimation dimension in ESTIMACS model. Besides that, according to the Table, risk also influence the software cost estimation. Thus, it still considers as factor in conceptual model that concern on managing the risk that might cause impact the project as well as the cost require for a project.
d. Project Complexity	
25. The programmer is familiar with the programming language. 26. The project requirements completed when the project about to start. 27. The government system is a high degree of interdependence. 28. The software design planning is efficient. 29. The project team has experience with the new technology that will be implemented. 30. Estimating the efforts is the difficult task in software cost estimation.	According to the previous research (see Table), project complexity has been identified as the factors influence of the SCE. In conceptual model, project complexity expands the factors by including and selecting the factors that has significant impact on the software cost estimation process.
e. Software development process	
31. The government relies on the outside agency (consultant) in managing the project. 32. The execution and management process is based on the hierarchical structure. 33. Poor planning in the first stage of project influences the whole stages of the project. 34. Project planning is effective and efficient. 35. The project well managed and organized. 36. The project progress is on track with the estimation schedule. 37. The software development methodology applied is suitable for the project.	As mentioned by Rajkumar & Alagarsamy (2013), software cost estimation cannot separate with software development process due to it is integrated to each other. The software cost estimation can perform after the first development process complete. Therefore, software development process includes as the factor in conceptual model. The sub-factors focus on the first and second software development process which are software specification and software development.
f. Estimation and Schedule	
38. The cost estimation is estimated based on owner estimate cost (OEC). 39. The actual cost is often unreliable with the estimation cost. 40. The project completed within schedule. 41. The resource of the project is sufficient to accomplish the project.	Schedule is one of essential aspect in the software cost estimation due to the project is successful if it is completed within the time and cost estimated (Rajkumar & Alagarsamy 2013).

to be continued…

…a continuation

42. Inaccuracy result of software cost estimation causes the project force to be canceled/revised. 43. The developer involves in estimating the cost of some software project phases.	Ramesh & Reddy (2016) stated that project duration, cost, efforts and resources are important measurement in software cost estimation. Thus, the estimation and schedule include as then actor in conceptual model since it is vital for estimating the software cost.
g. Communication	
44. Miscommunication between the project team could lead to project failure. 45. The progress of the project is updated effectively by the project team. 46. Any changes made has been agreed by all project team. 47. Project manager and top management communicate effectively with the project team. 48. Miscommunication between the project team, project manager, and top management could produce a wrong decision.	Rajkumar & Alagarsamy (2013) identified the communication as the factor influence of the software cost estimation. In the conceptual model, the communication factor is expanded it to five sub-factors because smooth communication between project manager, top management and project team has a big impact towards the software cost estimation accuracy.
IV. Organizational Factor	According to Medveska & Berzisa (2015), the environment of public sector is quite different with private sectors. The private sector is more flexible than public sectors, especially the public sector has its own criteria in developing a software. By looking the differences, thus the organizational factor is added into a factor that influence of software cost estimation.
49. The hierarchical structure influences the decision-making process. 50. The different perspective on the software cost estimation technique influences the process of decision making. 51. Changed the personnel composition of the project team. 52. The personal conflict between the project team influences the decision making.	The software cost estimation in Indonesian public sector involves many departments which might has different point of view on the software development project. This might impact of the software cost estimation as well.

2.5 CONCLUSION

Software cost estimation is significant for any software development project because it has a great impact on the project's success. If the cost estimation is not effective, the project could be overrun and over budget. As discussed in previous sections, inadequate cost estimation is caused by many factors that are faced by private and public organizations. The cost estimators should consider all issues and challenges in estimating the cost because if they are not aware of it, there is a high possibility of the actual cost turning out to be unreliable and not as expected in initial cost estimation. Furthermore, the research in cost estimation in government project is few conducted. Therefore, it is important to study cost estimation in a government project. While it is observed that there are software cost estimation techniques and models introduced by previous researchers, none of them claim that a particular model will produce accurate results. Each model has its own strengths and weaknesses. Therefore, the research on software cost estimation is conducted in order to enhance the models that have been developed so that it can enhance the software cost estimation accuracy. The accuracy in software cost estimation means the difference actual cost and estimation cost is not too high. It is difficult to get an exact result as it is intangible. Therefore, the software cost estimation process is complicated.

CHAPTER III

RESEARCH METHODOLOGY

3.1 INTRODUCTION

This research employs quantitative and qualitative approaches in order to obtain varied information from the respondents. The survey questionnaire is used as a tool for the quantitative method to obtain the respondent's perceptions while the qualitative approach is conducted through interviews of the government employees involved in the software cost estimation. These approaches are expected to garner information about the current software cost estimation techniques used in regional government bodies in Indonesia. Besides, the researcher is able to identify the software cost estimation problems in developing a software cost estimation model to be used in government project projects.

The research methodology includes the research design, population, sample, data collection method, data instruments and data analysis. This research consists of four phases, i.e. the conceptual study, empirical study, developing the model and model validation. For the conceptual study, information is collected from the works of previous researchers. In the empirical study, the data is obtained via survey questionnaires, and, interviews with the target respondents. Hence, the study will obtain information from people who are directly involved in estimating the cost of the software development process. Based on the conceptual and empirical study, the researcher will produce a model that describes the software cost estimation complexities from various aspects. The last stage involves model validation.

3.2 RESEARCH DESIGN

According to Roller (2015), research design is important to get the answer to the research questions. It is a framework or strategy that has been planned to achieve the research objectives by conducting certain research processes. In this study, the research findings are based on primary and secondary data. For the primary data, the result would be delivered based on findings from the empirical study based on questionnaires and interview results analysis from the respondents. While the secondary data is based on the conceptual study by exploring the previous research that has been done by previous researchers to identify the research problems and research gaps such as journals, articles, books, papers or website that are accessible. The research design of this study is shown in Figure 3.1.

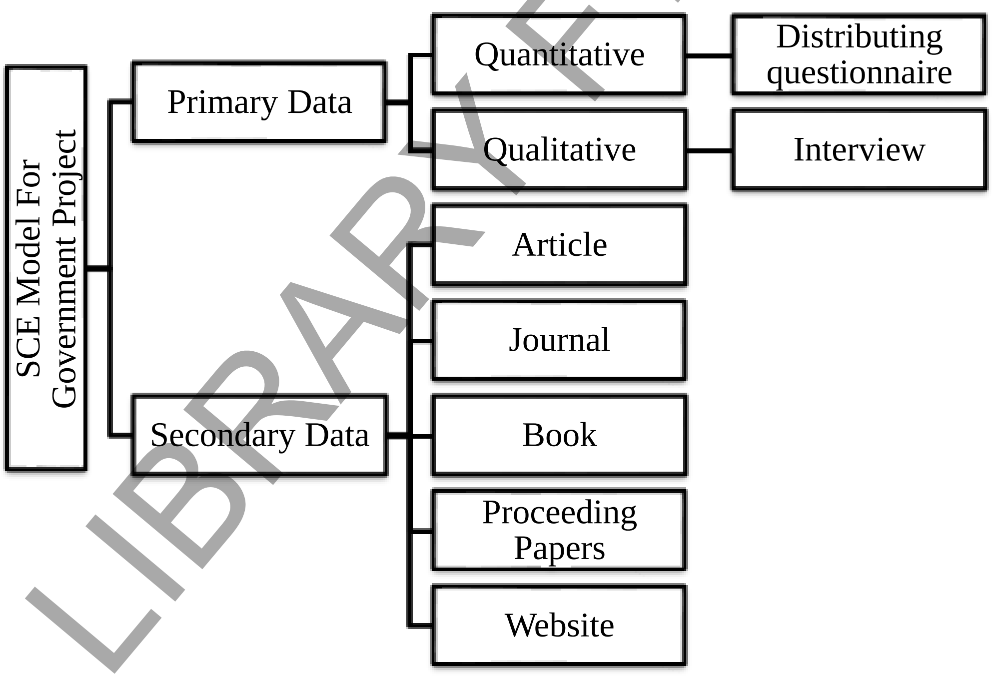

Figure 3.1 Research Design

3.3 RESEARCH PHASES

This study consists of four phases which are conceptual study, empirical study, model development and model validation. The phases of this study are elaborated as in table 3.1.

Table 3.1 Research Phases

Research Phase	Input	Activities	Output
Phase 1 **Conceptual study**	Literature Review	• Explore previous related research and literature. • Identify the research problems. • Determine the research gaps.	• Background of research • Problem Statement of research • Research questions • Research objectives • Research scope • Literature Review
Phase 2 **Empirical Study**	• Conceptual study	• Develop questionnaires. • Validate the questionnaires with the experts. • Pilot questionnaires test. • Pilot questionnaires revision. • Data Collection • Design interview question. • Validate the interview questionnaire with the experts. • Conduct the interview • Data Analysis	• Questionnaires validated • Gathered the questionnaires • Findings on current practice of software cost estimation in government project • Findings on the critical factors of SCE
Phase 3 **Model Development**	• Empirical Study Result	• Develop model based on the findings	• Final Model
Phase 4 **Model Validation**	• The Proposed Model	• Validate the model.	• Model Validated

3.3.1 Conceptual Study

A conceptual study is the first stage of this research. According to Kothari (2009), the conceptual study is related to some ideas or theory which is used to develop a new concept based on the previous ones. In this phase, the researcher gathers the information about software cost estimation especially in government project from many sources. The sources are books, articles, journal, website and proceeding papers from computer science database like ACM Digital Library, ScienceDirect, IEEE, Springer Link and also ResearchGates.

From those information sources, the researcher analyzes and reviews the literature about the software development in the government project, software cost estimation, cost estimation history, cost estimation phase and the software cost estimation techniques. As the outputs of this research stage are research background, problems statement, research questions, research objectives and literature review. The summary of this phase is shown in Table 3.2.

Table 3.2 Conceptual study

Sources	Conceptual study	Outputs
Books. Articles. Journal. Website Proceeding papers	Software development in the public sectors Software cost estimation (SCE) History of SCE Cost estimation phases SCE in public sectors SCE in Indonesia SCE Metrics Cost estimation techniques SCE Model	**Research background** The content of the research background comprises the important of the software cost estimation accuracy due to high number of failures software development project due to inaccurate the estimation result. **Problems statement** Problem statement describe issues of software cost estimation in public sectors. It also includes the weaknesses of the existing models. Therefore, there is a need to develop the software cost estimation model. **Research questions & Research objectives** Focus study of the research are identify the SCE practice Indonesian regional government and determine the critical factors of SCE. While the output of this research is to develop a SCE model. **Research Scope** Research scope consists of the research focus, model scope and limitation and people **Significance of the Research** The content of significant of research is the essential of the research about the software cost estimation for government, project managers, academicians, researchers and practitioners. **Expected Results** The expected result of the SCE is the SCE process more effective and efficient so that the result of the estimation in line with estimating cost.

to be continued…

…a continuation

	Research Methodology Research methodology comprise the research process from the conceptual study until model validation. **Literature Review** Literature review comprise previous research that is related with software cost estimation which is significant for proposed model development. **Conceptual model** The conceptual model constructed by elaborating the references in chapter 1 and Literature review.

3.3.2 Empirical Study

An empirical study is the second stage in this research. The empirical study is gathering the evidence through experiment or observation to get the data (facts) (Kothari 2009). As stated in Long (2014) the empirical study is a process to collect and analyze the data from the primary sources through direct observation or experiences. The data can gather through the qualitative and quantitative approaches. In this study, the empirical study is conducted by using the aforementioned approaches. The empirical study consists of five processes which are designing the questionnaire, expert validation, pilot testing, data collection and data analysis. The summary of this phase is shown in Figure 3.3.

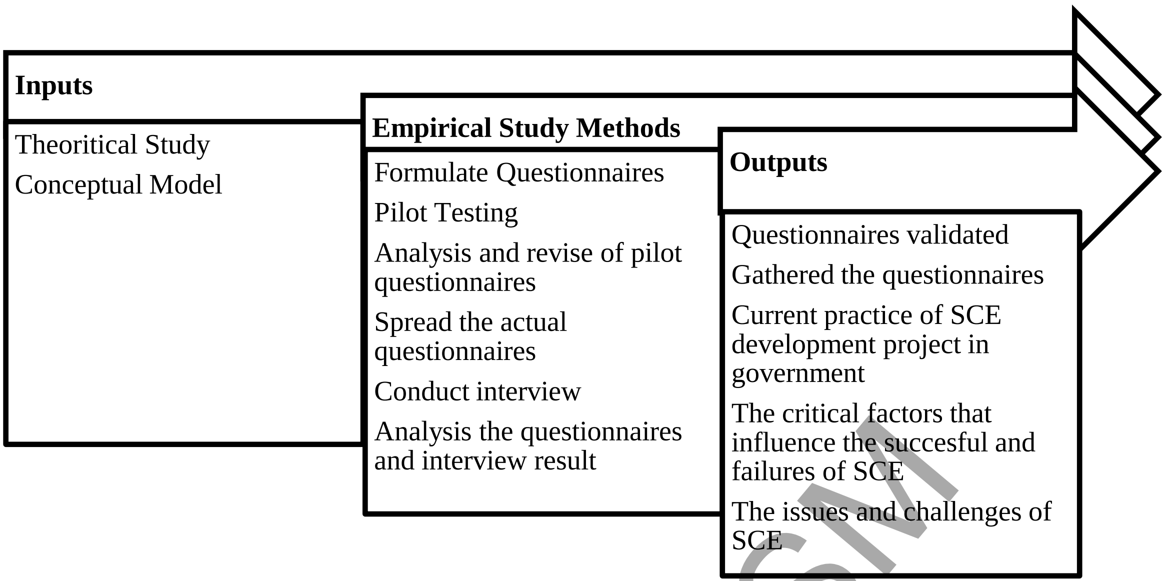

Figure 3.2 Empirical Study

a. Questionnaire Design

The questionnaire design develops based on the conceptual model (see Figure 2.18 and Table 2.25) and the previous research in literature review. The questionnaire is divided into two because the methods or this research use quantitative and qualitative approach.

1. Questionnaire for Quantitative Approach

The questionnaire for quantitative approach (see Appendix B) consists of two characteristics and three sections which are:

- The structure of question is close-ended questions.

- The scale of measurement of data would be in rating scale which requires the respondents to choose either strongly disagree, disagree, neutral, agree or strongly agree. The purpose is to calculate the common choice of answer from the respondents.

Furthermore, the questionnaire design process describes in Table 3.3 which consists of three sections.

Table 3.3 Questionnaires Structure

Items	Questions	Description
Section A Personal Data	Gender Age Highest educational level Current Position in the organization	In Section A, it contains the personal data of the respondents such as gender, age, experience, education level and position in the organization. These questions used to identify the impacts of this factors on software cost estimation in government.
Section B The current practice of software cost estimation in a regional government agency.	Who does estimate the software cost?	According to previous research, one of the factors influence of SCE factor is incompetence of cost estimator (see Table 2.4). Thus, this question is to analyse the people who has authority to perform the software cost estimation. As software cost estimator, the persons should have multidiscipline knowledge as shown in Table 2.3.
	When do organization usually do the cost estimation?	Another factor of inaccurate result of software cost estimation (see Table 2.4) is because lack of details in scope and requirements due to the estimation is done in early phase of the project. This question is to identify time for performing software cost estimation by the government of Indonesia.
	What is the most influential factor of the inaccuracy of the software cost estimation?	This question is to find out the significant factor influence of the SCE in public sector of Indonesia that cause the inaccurate result of the SCE.
	Did the scope and requirements of the project often modify? How little the scope and requirements are changed in software project?	This question is to identify the frequent of change and amount of change scope and requirements of project. The reason is because changing of scope and requirements also give the impacts on the SCE.
Section C Factors influence of software cost estimation in software development government projects.	Refers to Table 2.25 and Figure 2.18.	The respondents require selecting the answer based on the Liker-Scale rate. The purpose is to identify and analyze how those factors influence the software cost estimation process in public sector.

2. Questionnaire for Qualitative Approach

The question is qualitative approach consists of two parts (see APPENDIX C). This first part is about the personal data of the participants, while the second part focus on the understanding of the participants on the software cost estimation concept. Then, the questions also include the software cost estimation technique and the input use in estimating the software cost. Besides that, the question also about the effectiveness of the current SCE technique. Furthermore, since one of the factor inaccurate result in SCE is inexperience of cost estimator. Thus, it is important to investigate the people involve in software cost estimation. Lastly, the question is to determine the participants knowledge about the existing models and the need to develop a new software cost estimation model.

b. Content Validation

The content validation uses to validate the questionnaire content, language, and statistic. The experts' perspective is important to enhance the question so that it is understandable and match with the research questions and research objectives. The content validation involves three experts in order to check the content, statistical analysis and language.

Based on the comments and suggestions given, there are questions that have been changed due to inappropriate language as shown in Figure 3.4. The negative question might offense the respondents, hence the sentence change into an affirmative sentence.

7.	The project manager is lack of knowledge and skill in ICT					

Figure 3.3 Example of inappropriate language in the questionnaire

Besides that, there are some questions that have been removed due to its repetitive nature as also similarity with other questions. There is also a question that has been deleted due to difficulty in answering the same. As shown in the Figure 3.5, the respondent will find it difficult to answer since it requires an activity to measure the percentage of the inaccuracy of estimation.

4. How is the estimation accuracy in practice?

- [ ] Less than 60%
- [ ] 61-70%
- [ ] 71-80%
- [ ] 81-90%
- [ ] 90-100%

Figure 3.4 Ineffective Questionnaire

Furthermore, the experts have also given advice on the scale that is used in the questionnaire so that it is relevant to the statistical software analysis. In this study, the statistical software analysis used is RASCH model software. The scale used is the LIKERT scale such as very disagrees, disagree, neutral, agree and very agree. Thus, the expert validity is important to evaluate the questionnaire language, content and statistical measurement so that it is easy to understand and relevant to the respondents.

3.3.3 Pilot Testing

The pilot testing has been conducted to get the feedback from respondents with aims to improve the questionnaire based on the respondent perspective. The pilot testing is conducted in six agencies in West Sumatera Province. The agencies are located in Padang city, Tanah Datar Regency and Payakumbuh city. The respondents for pilot testing consist of 20 respondents who are involved in estimating the software cost in government. The response of the respondents is analyzed by using RASCH model so that the questionnaire is improved for the actual data collection.. Furthermore, the questionnaire of the pilot testing attaches in APPENDIX A.

a. Pilot Testing Result Analysis

The pilot testing is used to check the item reliability and item validity. Hence, based on the result, it can be used to determine the items that have low validity. As the impact, those items should be deleted. The following is the result of the pilot study.

b. Result of software cost estimation survey

Based on the pilot testing questionnaire (APPENDIX A), part B of the questions focus on the people in charge of SCE, the SCE schedule, the most significant factor influence SCE in public sector and the change made in scope and requirements of project. So, the result of part B questionnaire describe in Table 3.4.

Most of the software cost estimation in public sectors is estimated by an IT officer and project manager. The software cost estimation is conducted during the project proposal phase whereby the software requirements are still not detailed. Commonly, in project proposal phase, the general scope and requirements are only identified. The software cost estimation is conducted manually due as there is no proper tool to estimate the software cost. The most influential factor of the inaccuracy of software cost estimation is the lack of experience in previous related projects. The scope and requirements sometimes change with less number of change.

Table 3.4 Pilot Test Result

No	Questions	Frequency	Percentage (%)
1.	**People who have authority to estimate the software cost:**		
	IT officer	9	42.85%
	Accountant	1	4.76%
	Project Manager	6	28.57%
	Cost Analyst	3	14.28%
	Others	2	9.52%
2.	**The software cost estimation is conducted during:**		
	Project proposal phase	18	85.71%
	Feasibility study	1	4.76%
	Scope and Requirement Analysis	2	9.52%
	Later Phases	-	-
3.	**The most influential factor of the inaccuracy of software cost estimation.**		
	Scope and requirements are not clear	5	23.80%
	There is no tool to estimate the software cost	7	33.33%
	Not using an available software cost estimation	1	4.76%
	Lack of experiences in previous related projects	8	38.09%
	Lack of top management support	-	-
4.	**The frequency of changing the scope and requirements of the project.**		
	Never	4	19.04%
	Seldom	4	19.04%
	Sometimes	10	47.61%

to be continued...

	…a continuation		
	Often	3	14.28%
5.	**The number of changed scope and requirements of a software project.**		
	No	3	14.28%
	A little	12	57.14%
	Moderately	5	23.80%
	Substantially	1	4.76%

c. Pilot Test Result of Factors influence of software cost estimation

In pilot testing questionnaire (APPENDIX A), it also includes part B which content the factors influence of the software cost estimation in public sectors. The analysis used Rasch model which can check the reliability and validity based on the result of summary statistic, misfit item and item dimensionality.

i. Summary Statistic

The summary statistic (Figure 3.5) reveals that the Cronbach alpha value is which is 0.92. It indicates that the interaction between the person and the item is very good. Furthermore, the item reliability is good which means that the items quality in the instrument is good too. The person reliability value is high too which is 0.91 which indicates that the respondent consistency in answering is high. So, the respondents are competent in responding the instrument of study. Furthermore, the person mean is higher than item mean, which means that the whole test meet expectation. The mean square of the person and item reveals that both are productive for measurement because the infit and outfit MNSQ falls between 0.50 and 1.50. While the standardized fit statistic also has the reasonable data to predict due to the value of infit and outfit are between -1.90 and 1.90. In addition, the mean person measures us 1.60 which shows that overall the respondents agree that those factors influence of software cost estimation. The respondent separation value also high which is 3.14. It indicates the group separation between person and items. The greater the separation value is, so, in general, the instrument quality is good in terms of the person and the items.

```
TABLE 3.1 C:\Users\User\Desktop\Pilot test resul ZOU024WS.TXT  Dec  5 13:36 2017
INPUT: 21 Person  52 Item  REPORTED: 21 Person  52 Item  5 CATS  WINSTEPS 3.72.3
--------------------------------------------------------------------------------

     SUMMARY OF 21 MEASURED Person
-------------------------------------------------------------------------------
|          TOTAL                         MODEL         INFIT        OUTFIT    |
|          SCORE     COUNT     MEASURE   ERROR      MNSQ   ZSTD   MNSQ   ZSTD |
|-----------------------------------------------------------------------------|
| MEAN      209.6      52.0       1.60     .22      1.05    -.3   1.02    -.4 |
| S.D.       16.2        .2        .82     .03       .83    2.6    .80    2.6 |
| MAX.      244.0      52.0       3.65     .30      4.42    8.6   4.29    8.7 |
| MIN.      177.0      51.0        .36     .18       .30   -4.3    .32   -4.4 |
|-----------------------------------------------------------------------------|
| REAL RMSE    .25 TRUE SD    .78  SEPARATION  3.14  Person RELIABILITY  .91 |
|MODEL RMSE    .22 TRUE SD    .79  SEPARATION  3.53  Person RELIABILITY  .93 |
| S.E. OF Person MEAN = .18                                                   |
-------------------------------------------------------------------------------
Person RAW SCORE-TO-MEASURE CORRELATION = .99
CRONBACH ALPHA (KR-20) Person RAW SCORE "TEST" RELIABILITY = .92

     SUMMARY OF 52 MEASURED Item
-------------------------------------------------------------------------------
|          TOTAL                         MODEL         INFIT        OUTFIT    |
|          SCORE     COUNT     MEASURE   ERROR      MNSQ   ZSTD   MNSQ   ZSTD |
|-----------------------------------------------------------------------------|
| MEAN       84.6      21.0        .00     .35      1.02    -.1   1.02    -.1 |
| S.D.        6.7        .1        .75     .04       .59    1.5    .59    1.5 |
| MAX.       97.0      21.0       1.85     .44      2.81    3.6   2.72    3.6 |
| MIN.       64.0      20.0      -1.75     .25       .25   -2.9    .27   -2.9 |
|-----------------------------------------------------------------------------|
| REAL RMSE    .39 TRUE SD    .65  SEPARATION  1.68  Item   RELIABILITY  .74 |
|MODEL RMSE    .35 TRUE SD    .67  SEPARATION  1.90  Item   RELIABILITY  .78 |
| S.E. OF Item MEAN = .11                                                     |
-------------------------------------------------------------------------------
UMEAN=.0000 USCALE=1.0000
Item RAW SCORE-TO-MEASURE CORRELATION = -.98
1091 DATA POINTS. LOG-LIKELIHOOD CHI-SQUARE: 2006.90 with 1016 d.f. p=.0000
Global Root-Mean-Square Residual (excluding extreme scores): .6511
```

Figure 3.5 Summary Statistic Result of Pilot Study

ii. Misfit Item

According to Bambang & Wahyu (2014), misfit is inappropriate items based on the answer of the respondent which is inconsistence. The misfit item is identified by the following criteria:

- The outfit mean square value is 0.16<MNSQ<1.84
- The outfit Z-Standard (ZSTD) value is -2.0<zstd<+2.0
- The Point Measure correlation (PT Mean Corr) is 0.4<Pt Measure Corr<0.85

Based on the result in Figure 3.6, it reveals that there are six misfit items which are an item with entry number 1, 4, 6, 12, 28, and 31 because the misfit item result is sorted from the appropriate list. Therefore, those items fall into difficult group category. However, overall the respondents are competent in answering the questionnaire. The first item uses the standard to estimate the software cost. The fourth item is about the competency and knowledge of the project manager. The six-item is about the proficiency of the project manager in controlling the software project. The entry number twelve is lack of training for staff in software development project. On the other hand, the researcher decided not to remove the items from the actual survey as this task is still questionable. First, those items are really difficult, or they do not reflect the respondent's ability due to the small number of respondents. Hence, these items still remain used for the actual survey.

The researcher decided to remove the entry number 28 and 31. The entry number 28 is the software design planning complicated. It may be a difficult task because lack of respondent capability in planning the design of software. Furthermore, the 31^{st} item is the government relied on the agency (consultant) in managing the project. This item is deleted because this question is also part of the qualitative approach question.

Based on the result in Figure 3.6, it can be concluded that all the software cost estimation above influences the software cost estimation accuracy result. Therefore, the model is valid and reliable for measuring this study.

TABLE 10.1 C:\Users\User\Desktop\Pilot test resu ZOU862WS.TXT Dec 5 14:35 2017
INPUT: 21 Person 52 Item REPORTED: 21 Person 52 Item 5 CATS WINSTEPS 3.72.3

Person: REAL SEP.: 3.14 REL.: .91 ... Item: REAL SEP.: 1.68 REL.: .74

Item STATISTICS: MISFIT ORDER

ENTRY NUMBER	TOTAL SCORE	TOTAL COUNT	MEASURE	MODEL S.E.	INFIT MNSQ	INFIT ZSTD	OUTFIT MNSQ	OUTFIT ZSTD	PT-MEASURE CORR.	PT-MEASURE EXP.	EXACT MATCH OBS%	EXACT MATCH EXP%	Item
6	87	21	-.20	.36	2.81	3.6	2.72	3.6	A .24	.42	42.9	62.6	I0006
28	86	21	-.07	.36	2.62	3.3	2.61	3.4	B .50	.42	38.1	62.1	I0028
12	83	21	.29	.34	2.50	3.0	2.51	3.2	C .38	.44	33.3	60.3	I0012
4	88	21	-.33	.37	2.50	3.2	2.39	3.2	D .38	.41	52.4	63.1	I0004
31	71	21	1.38	.27	2.16	2.8	2.27	3.0	E .17	.49	38.1	45.3	I0031
1	86	21	-.07	.36	1.90	2.1	1.90	2.2	F .42	.42	61.9	62.1	I0001
3	81	21	.51	.32	1.63	1.6	1.75	1.9	G .28	.44	52.4	57.9	I0003
39	70	21	1.45	.27	1.45	1.4	1.50	1.5	H .31	.49	42.9	45.6	I0039
30	78	21	.81	.31	1.15	.5	1.39	1.1	I .18	.45	66.7	55.2	I0030
33	88	21	-.33	.37	1.26	.8	1.24	.8	J .38	.41	57.1	63.1	I0033
8	83	21	.29	.34	1.18	.6	1.25	.8	K .11	.44	66.7	60.3	I0008
17	96	21	-1.57	.43	1.24	.9	1.05	.3	L .52	.33	61.9	63.4	I0017
49	88	21	-.33	.37	1.14	.5	1.23	.8	M .02	.41	42.9	63.1	I0049
23	82	21	.40	.33	1.04	.2	1.21	.7	N .45	.44	61.9	58.4	I0023
11	97	21	-1.75	.44	1.17	.7	1.02	.2	O .37	.32	66.7	66.3	I0011
2	64	21	1.85	.25	1.10	.4	1.17	.6	P .23	.52	38.1	43.8	I0002
38	79	21	.71	.31	1.05	.3	1.15	.5	Q .43	.45	52.4	55.9	I0038
35	87	21	-.20	.36	1.08	.3	1.06	.3	R .63	.42	42.9	62.6	I0035
42	73	21	1.23	.28	1.08	.3	1.08	.4	S .35	.48	71.4	47.2	I0042
5	88	21	-.33	.37	1.06	.3	1.04	.2	T .41	.41	76.2	63.1	I0005
29	91	21	-.75	.38	1.06	.3	1.01	.1	U .42	.39	57.1	61.6	I0029
36	85	21	.05	.35	1.01	.2	1.03	.2	V .65	.43	47.6	61.9	I0036
52	80	21	.61	.32	.95	.0	1.00	.1	W .21	.45	61.9	56.4	I0052
7	88	21	-.33	.37	.96	.0	.87	-.3	X .66	.41	76.2	63.1	I0007
25	92	21	-.90	.39	.96	.0	.91	-.2	Y .55	.38	52.4	60.3	I0025
44	85	21	.05	.35	.91	-.1	.96	.0	Z .37	.43	57.1	61.9	I0044
20	88	21	-.33	.37	.90	-.2	.89	-.2	z .53	.41	61.9	63.1	I0020
34	85	21	.05	.35	.86	-.3	.87	-.3	y .62	.43	57.1	61.9	I0034
32	75	21	1.07	.29	.81	-.5	.81	-.5	x .41	.47	57.1	50.1	I0032
51	71	21	1.38	.27	.76	-.7	.70	-.9	w .31	.49	47.6	45.3	I0051
43	77	21	.90	.30	.74	-.7	.68	-.9	v .33	.46	71.4	54.0	I0043
21	92	21	-.90	.39	.73	-.8	.71	-.9	u .56	.38	61.9	60.3	I0021
19	90	21	-.61	.38	.67	-1.0	.63	-1.2	t .71	.40	71.4	62.4	I0019
22	85	21	.05	.35	.67	-.9	.62	-1.2	s .65	.43	66.7	61.9	I0022
26	90	21	-.61	.38	.66	-1.1	.62	-1.2	r .72	.40	71.4	62.4	I0026
13	96	21	-1.57	.43	.65	-1.3	.60	-1.3	q .56	.33	71.4	63.4	I0013
46	87	21	-.20	.36	.64	-1.0	.65	-1.1	p .51	.42	71.4	62.6	I0046
16	88	21	-.33	.37	.64	-1.1	.61	-1.2	o .60	.41	71.4	63.1	I0016
10	87	21	-.20	.36	.63	-1.1	.62	-1.2	n .71	.42	71.4	62.6	I0010
47	85	21	.05	.35	.61	-1.1	.62	-1.1	m .53	.43	76.2	61.9	I0047
50	82	21	.40	.33	.56	-1.3	.61	-1.2	l .13	.44	66.7	58.4	I0050
27	81	21	.51	.32	.59	-1.2	.60	-1.2	k .20	.44	61.9	57.9	I0027
45	86	21	-.07	.36	.53	-1.5	.60	-1.3	j .27	.42	81.0	62.1	I0045
9	85	21	.05	.35	.58	-1.3	.58	-1.3	i .56	.43	76.2	61.9	I0009
40	85	21	.05	.35	.53	-1.5	.56	-1.4	h .42	.43	76.2	61.9	I0040
18	89	21	-.47	.37	.52	-1.6	.53	-1.6	g .61	.40	76.2	62.9	I0018
14	92	21	-.90	.39	.52	-1.8	.50	-1.8	f .58	.38	81.0	60.3	I0014
48	91	21	-.75	.38	.50	-1.8	.52	-1.8	e .54	.39	76.2	61.6	I0048
41	80	20	.24	.35	.49	-1.5	.51	-1.5	d .56	.42	70.0	61.6	I0041
15	91	21	-.75	.38	.46	-2.0	.48	-2.0	c .58	.39	76.2	61.6	I0015
24	84	21	.17	.34	.42	-1.9	.42	-2.0	b .80	.43	76.2	60.9	I0024
37	83	21	.29	.34	.25	-2.9	.27	-2.9	a .72	.44	81.0	60.3	I0037
MEAN	84.6	21.0	.00	.35	1.02	-.1	1.02	-.1			62.4	59.5	
S.D.	6.7	.1	.75	.04	.59	1.5	.59	1.5			12.9	5.3	

Figure 3.6 Misfit Items of Pilot Results

iii. Item Dimensionality

The item dimensionality (see Figure 3.7) reveals that to evaluate the variety of instruments are able to measure that supposed to measure. Based on the result, the raw variance explained by measures value is 30.1% which means it fulfills the item dimensionality requirement. The requirement is the value should be greater than 20%. Hence, the variety instrument can measure the factors of software cost estimation. There is also an instrument variance that cannot be explained. The ideal result is not more than 15%. Thus, the result there is two variance that is more than 10% which are unexplained variance first and second contrast with value 12.2% and 11.0%. The rest have a value less than 10% (Bambang & Wahyu 2014).

```
TABLE 23.0 C:\Users\User\Desktop\Pilot test resu ZOU024WS.TXT  Dec  5 13:36 2017
INPUT: 21 Person  52 Item  REPORTED: 21 Person  52 Item  5 CATS  WINSTEPS 3.72.3
--------------------------------------------------------------------------------

     Table of STANDARDIZED RESIDUAL variance (in Eigenvalue units)
                                                 -- Empirical --    Modeled
Total raw variance in observations     =       74.4 100.0%          100.0%
  Raw variance explained by measures   =       22.4  30.1%           32.8%
    Raw variance explained by persons  =        6.7   9.1%            9.9%
    Raw Variance explained by items    =       15.7  21.1%           23.0%
  Raw unexplained variance (total)     =       52.0  69.9% 100.0%    67.2%
    Unexplned variance in 1st contrast =        9.1  12.2%  17.4%
    Unexplned variance in 2nd contrast =        8.2  11.0%  15.7%
    Unexplned variance in 3rd contrast =        4.7   6.3%   9.0%
    Unexplned variance in 4th contrast =        4.3   5.8%   8.4%
    Unexplned variance in 5th contrast =        3.6   4.9%   7.0%
```

Figure 3.7 Item Dimensionality of Pilot Study

d. Actual Data Collection

Lastly, the distribution of the actual questionnaires to the respondents. Questionnaires are one of the types of quantitative research analysis tools to collect the information about the user's point of view regarding software cost estimation in the government project. Given the time factor, questionnaires are considered to be an efficient method of collecting data. The researchers will approach the respondents directly and also distribute it through emails. The target of respondents is government employees and practitioners who know software cost estimation in the government project. In this phase, the activity is investigating the current practice of software cost estimation and the software cost estimation factors that influence the software cost estimation process in a government project. The respondents will get the assurance that all the data are used for the research, and the identities of the defendants will be confidential.

i. Research Population

Research population is a collection of objects or elements which have similar characteristics to address the issues of the research questions (Kothari 2004). The research population of this study is comprised all the regional government institutions in West Sumatera Province in Indonesia. In this study, the participants require having experience working in public sectors in Indonesia, specifically have experience on software cost estimation so that the researcher can get the information based on the research scope. However, number of total research population is unknown.

ii. Research Sampling

Choosing the right samples are very significant in representing the population. The researcher would select the sample by using purposive sampling method. Purposive sampling is non-probability sampling which is also known as judgment sampling. Purposive sampling would be used because questionnaires and interview respondents are selected based on the subjective judgment by the researcher (Kothari 2004).

The research sampling of this study is:

- Government employees who involve in software cost estimation project in West Sumatera Province. It is because not every government employees understand and participate in cost estimation for software development. Hence, the government employees chosen are from THREE agencies which are *DISKOMINFO*[4], *BAPPEDA*[5], *and LPSE*[6].

---

[4] *DISKOMINFO (Dinas Komunikasi danInformatika*) is communications and informatics agency

[5] *BAPPEDA (Badan Perencanaan Pembangunan Daerah)* is regional agency for planning and development

[6] *LPSE (Lembaga Pengadaan Secara Elektronik)* is a provider of electronic procurement systems for government goods / services which operates by using electronic procurement system.

- The researcher will choose the samples from four cities and three regencies in West Sumatera Province. The cities are Padang, Payakumbuh, Solok, and Bukittinggi, while the regencies are Sijunjung, Tanah Datar, and Lima Puluh Kota.

The quantity of sampling is calculated by using Lemeshow approach. The Lemeshow approach can use to calculate the sample when the number of population is unknown. The elements of the estimation are absolute precision pr limit of error (e), and level of confidence value (Z_a) value is 1.96 (95%). Based on the result , a total of the sample used is 96 respondent (see Figure 3.9) (Lwanga & Lemeshow 1991).

$$N = ((Z_a/2)/e)^2$$
$$= (1.96/0.2)^2$$
$$= 96$$

Figure 3.8 Lemeshow Formula

iii. Data Analysis

The pilot test and actual data are analyzed using Rasch Model because it can explain the specific person capability and the item difficulties. Based on the result, it can be used to identify the respondent competency and the level difficulties of the items. Rasch model was introduced by Georg Rasch in 1960. The model is very popular because it is one of the item response theory (IRT) which describe the relation between persons and test items. Furthermore, the Rasch model was used for analyzing the dichotomous data; then it is evolved by Andrich to analyze the rating scale data. Masters make an improvement on the Rasch model too so that it can use to evaluate the partial model. Lastly, Linacre has introduced the facets model. Moreover, Rasch model can analyze the data from science and social science fields such as education, psychology, marketing, communication and so forth (Bambang & Wahyu 2014). According to Engelhard & Stefanie (2013), the Rasch model uses to measure the items, respondents and the relationship between the item and respondent.

The analysis of the data is done by using Winstep. In this study, the Rasch model uses to measure the pilot testing data which includes summary statistics, item measure, item fit order, person measure, and dimensionality (Bambang & Wahyu 2014).

a) Summary Statistics

Summary statistics used to describe the respondents and instruments quality in general. It also describes the connection between person and item. The outcome of the summary statistics is person measure, Cronbach alpha, person reliability, item reliability, infit, and outfit MNSQ and ZSTD.

b) Person measure

If the average is greater than 0.0, it shows that the respondents tend to agree with the statement in items.

c) Cronbach Alpha

Cronbach alpha uses to measure the interaction reliability between person and items. The following are the description of the result (see Table 3.5).

Table 3.5 Cronbach Alpha Value

Result	Explanation
<0.5	Very Bad
0.5-0.6	Bad
0.6-0.7	Enough
0.7-0.8	Good
>0.8	Very Good

Source: Bambang & Wahyu (2014)

d) Person and item Reliability

Based on Table 3.6, if the person reliability is less than 0.67 and item reliability is greater than 0.80, so meaning that the items quality in the instrument is good, but the consistency of respondent's answer is weak.

Table 3.6 Person and Item Reliability Value

Value	Description
<0.67	Bad
0.67-0.80	Enough
0.81-0.90	Good
0.91-0.94	Very Good
>0.94	Excellent

e) Infit and Outfit MNSQ

The mean square fit statistic shows the randomness of measurement which determines the total of the distortion in the measurement. The expected value is 0.5<y<1.5 because if less than that value, it indicates the data overfit the model. While if the value greater than that value, it means that the data underfit the model. The result of the mean square is divided into four categories which described in Table 3.7.

Table 3.7 Infit and Outfit MNSQ value

Value	Description
>0.2	Distorts or degrades the measurement system
1.5-2.0	Unproductive for construction of measurement, but not degrading
0.5-1.5	Productive for measurement
<0.5	Less productive for measurement, but not degrading. May produce misleadingly good reliabilities and separations

f) Infit and Outfit ZSTD

The standardized fit statistic is usually to measure the data fit the model or not. The expected value of the infit and outfit are between -1.9 and 1.9 which shows that the data have reasonable predictability. The value of the infit and outfit standardized are described in Table 3.8.

Table 3.8 Infit and outfit ZSTD value

Value	Description
≥3.0	Bad
2.0-2.9	Enough
-1.9-1.9	Good
≤0.2	Very Good

g) Scalogram

Scalogram uses to identify the misfit person. Scalogram is known as Gutmann matrix which can use determine the most competent respondent, less competent respondent and the carelessness of the respondent in giving the answer. For instance, the structure of answer given by the respondent is four (agree) and five (very agree). However, in the last answer the respondent tick one (very disagree). In this case, this respondent is careless in giving the answer. Thus, it can fall into less competent respondent.

h) Person and Item Map

The person maps show the person who tends to agree or disagree with the items given. The maps are to check the most item, and the easiest item is being endorsed by the respondent. In person map, the maximum location of a person indicates that the person who tends to agree with the items. Thus, this person is the most competent respondent. While the minimum location of the person indicates the person, who tends to disagree with the item. On the other hand, in item map, the maximum location is the most item that is being endorsed and vice versa.

3.3.4 Model Development

The third stage in this study is a development of the model as shown in Figure 3.9. The development process will start after analyzing the second stage results which depend on questionnaires results. The model is developed based on conceptual model (see Figure 2.18 and Table 2.25) that focus on the factors influence of software cost estimation in public sectors. Furthermore, the model can be developed based on the empirical study results which related with software development in public sectors, the current software cost estimation technique in Indonesia, and also the previous research that is related with software cost estimation process. Besides that, the model also develops based on the weaknesses and strengths of the previous model to get suitable factors that can be adopted to the model. Likewise, after the conceptual model developed, the empirical study (see Figure 3.3 for empirical study process explanation) also conducted by gather the data from the government employees who involve in the software cost estimation through interview and spreading the questionnaire. Thus, the final deliverables in this stage will be a model.

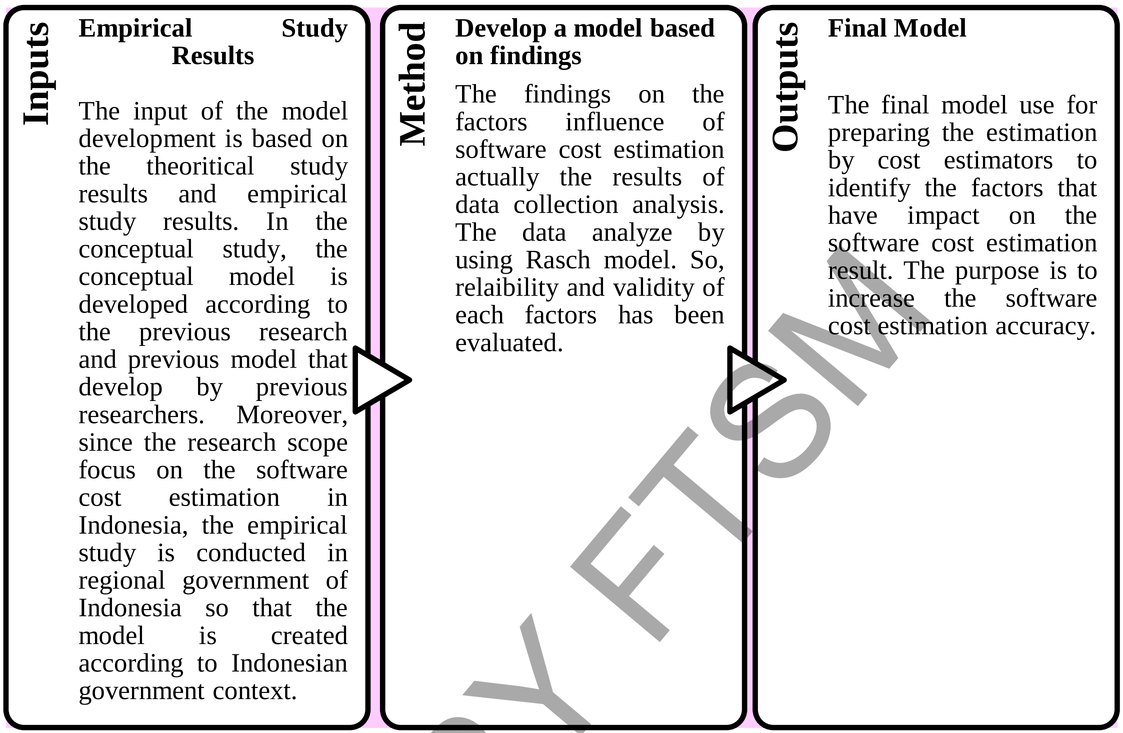

Figure 3.9 Model Development

3.3.5 Model Validation

Model validation is the last stage of this study. The validation stage determines whether the proposed model of the study is reliable to the users. The method adapted for model validation is by distributing the final model to some respondents. The respondents are government employees of Indonesia who involve in estimating the software cost. The purpose is to get the feedback from the users on the final model in order to ensure that the model can assist them in preparing the estimation. They also can provide any input or suggestions on the final model so that the software cost estimation process is more effective and efficient. The model validation conducted one time but from three agencies of regional government of West Sumatera Province. The summary of this phase is shown in Figure 3.10.

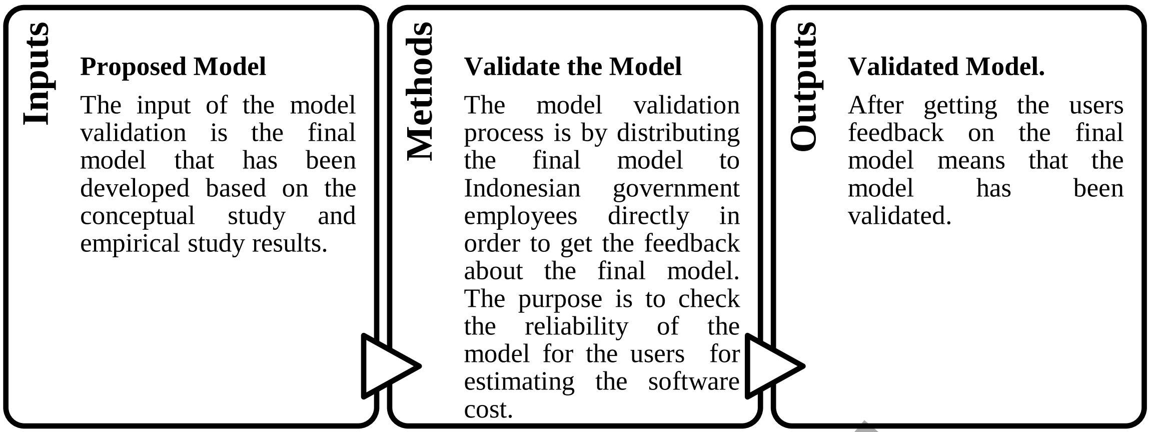

Figure 3.10 Validation of the Model

3.4 CONCLUSION

In conclusion, this study consists of four research phases which are conceptual study, empirical study, model development and model validation. Each of phases has a significant impact on the research due to each phase has different inputs, activities, and different outcomes. The first phase of this study is a conceptual study which is important to identify the research problems and research gaps. The conceptual study is completed based on previous research that has been conducted by previous researchers. Then, the second phase is empirical study whereby the data gathered from the government employees who are involved in estimating the software cost through pilot testing and actual data collection. So, the quantitative and qualitative approaches are conducted during this second phase. The third phase is a model development which means that the model is developed based on conceptual study and empirical study results. Lastly, model validation which is validated by the experts who are involved in estimating the software cost in government projects.

CHAPTER IV

RESULTS AND DISCUSSION

4.1 INTRODUCTION

Based on the conceptual study which has been discussed in the previous chapter, there are many factors that influence software cost estimation results which is shown in Table 2.4. Nevertheless, the accuracy results of software cost estimation still an issue even though many techniques and models that are available to perform the software cost estimation. As the impact, the project cannot be completed within the time, budget and resources that has been estimated before the project begins. Thus, this study is essential to determine the factors as well as the current practice of the software cost estimation in public sector. Besides that, few research discusses the software cost estimation in the government project, since most had focused on the software cost estimation in private sectors.

In this study, an empirical study is also conducted by using the quantitative and qualitative approaches. The respondents for quantitative study are 96 people who are government employees who work in *Diskominfo*[7], *Bappeda*[8], and *LPSE*[9] in the regional agency in West Sumatera Province. While for qualitative approach, five people were interviewed. Both approaches the result is significant to develop a

---

[7]*DISKOMINFO (Dinas Komunikasi danInformatika*) is communications and informatics agency

[8] *BAPPEDA (Badan Perencanaan Pembangunan Daerah)* is regional agency for planning and development

[9]*LPSE (Lembaga Pengadaan Secara Elektronik)* is a provider of electronic procurement systems for government goods / services which operates by using electronic procurement system.

software cost estimation model in the government project, specifically for regional government in Indonesia.

4.2 QUANTITATIVE APPROACH AND RESULTS ANALYSIS

The findings of the quantitative approach are divided into three points which are:

- The respondent profiles.

  The purpose is to analyze the gender, age, education, position and experience of the respondents.

- The current practice of software cost estimation

  The aims of the questions are to answer the first research question of this study which the software cost estimation techniques is used in Indonesian regional government.

- The factors influence of software cost estimation.

  The objective of this question is to answer the first and third research question which the current software cost estimation techniques is implemented, and factors influence of SCE in the regional government of Indonesia.

While for qualitative approach, respondents will answer the first, second and third of research questions.

4.3 THE QUANTITATIVE APPROACH RESULTS ANALYSIS

According to the questionnaire part A (see Appendix B) is about the respondent profile. The result of the respondent profile on this study is shown in Table 4.1.

Table 4.1 Respondent Profile

No	Description	Frequency	Percentage (%)
1)	**Gender**		
	Male	55	57.30%
	Female	41	42.70%
2)	**Age**		
	20-30	22	22.91%
	31-40	47	48.95%
	41-above	26	27.08%
3)	**Highest educational level**		
	SMA and SMK (Senior High School)	9	9.37%
	Diploma	10	10.41%
	Bachelor's degree	46	47.91%
	Postgraduates	31	32.29%
4)	**Current Position in Organization**		
	Head of Department	1	1.04%
	Head of Division	4	4.16%
	Sector Chiefs	16	16.66%
	Section Head	35	36.45%
	Staff	40	41.66%
5)	**Experiences**		
	Less than 5 years	23	23.95%
	5-10 years	28	29.16%
	10-15 years	18	18.75%
	15 years and above	27	28.12%

The ratio of the male respondent is higher than female respondents who are 57.30%, while female respondents are 42.70%. So, the most dominant gender in this study is male which is more involved in the software cost estimation process which is shown in Figure 4.1. Likewise, most of the respondent age is 31 to 40 years old which is 48.95% that is described in Figure 4.2. While 27.08% is 41 and above years old, the rest is between 20 to thirty years old.

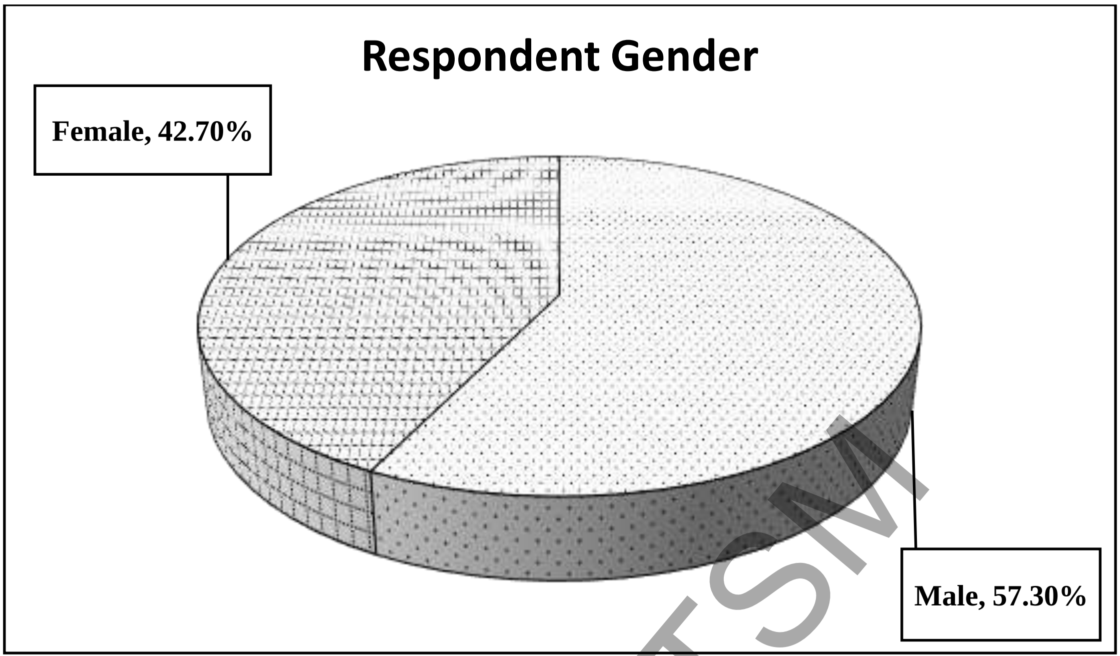

Figure 4.1 Respondent Gender

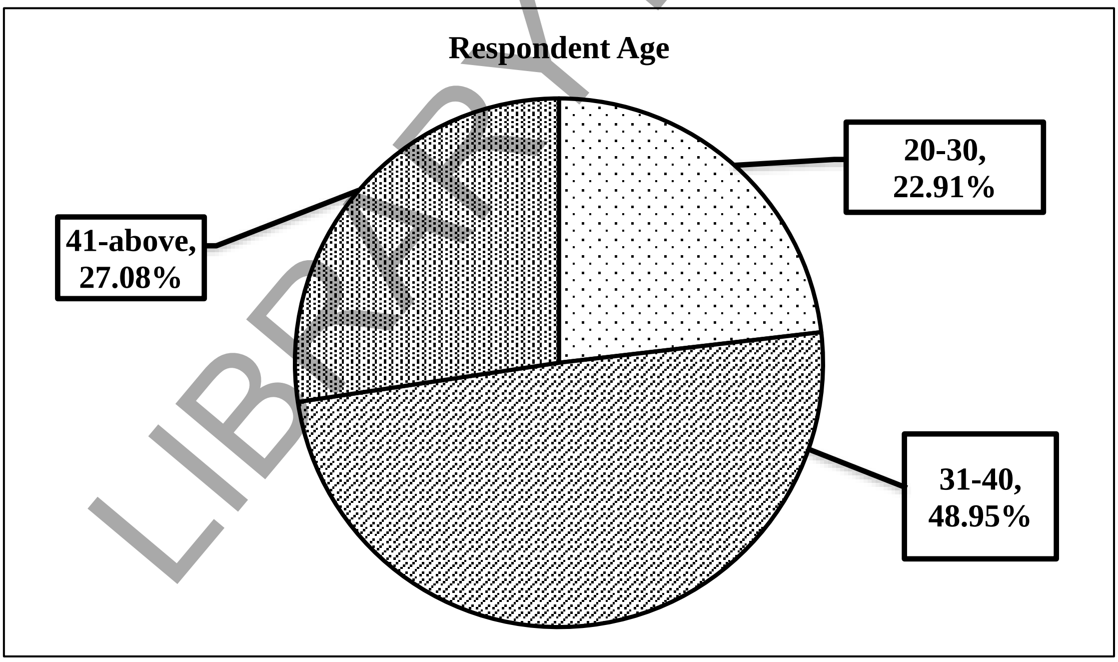

Figure 4.2 Respondent Age

Furthermore, most of respondent educational level is under graduation and post-graduation. It shows that they are knowledgeable and competent owing to their academic background. The highest educational level of respondents is described in Figure 4.3. Likewise, the dominant position of respondents is as staff in the agency which is 41.66%. It follows by the section head, section chiefs, head of division and head of the department. Figure 4.4 illustrates the current position of respondents in the organization.

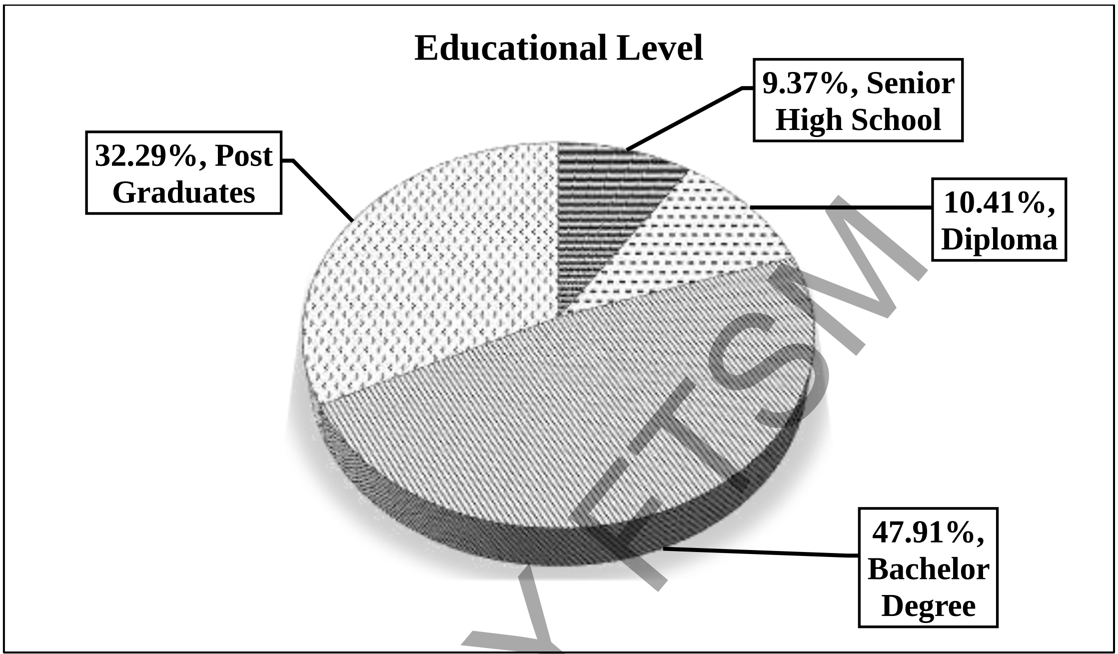

Figure 4.3 Educational Level

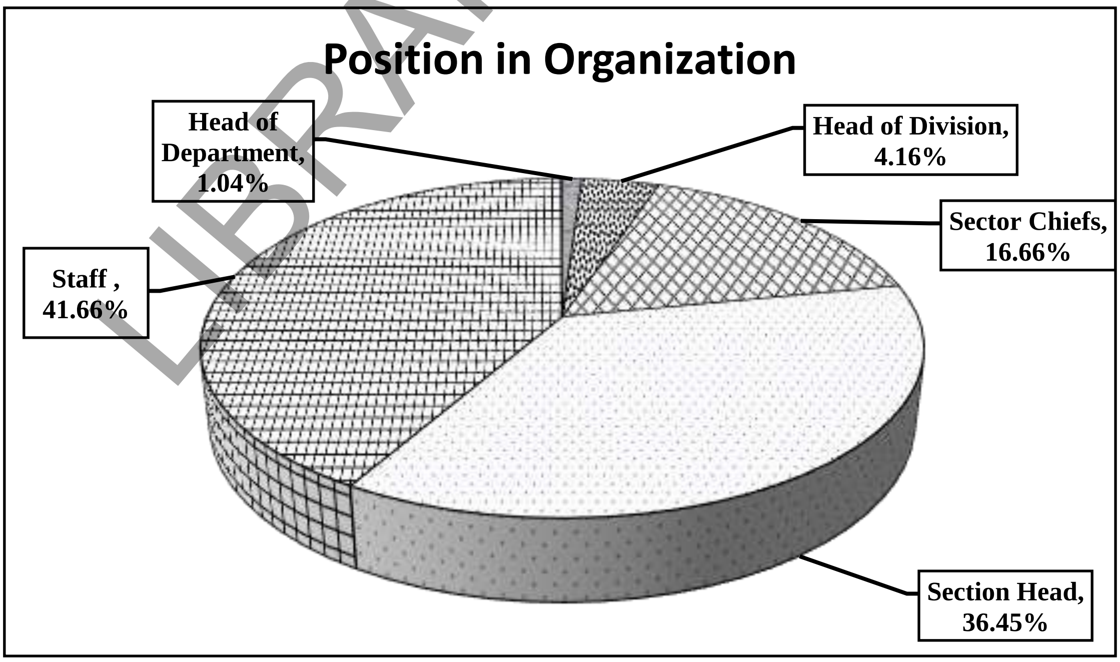

Figure 4.4 Current Position in Organization

Lastly, the result reveals that most respondents have a good experience since only 23.95% who work for the less five years. 28.12% has been working for 15 years and above, while 29.16% is working between five to ten years, while 18.75% is working between 10 to 15 years. The result of the experience of the respondents is illustrated in Figure 4.5.

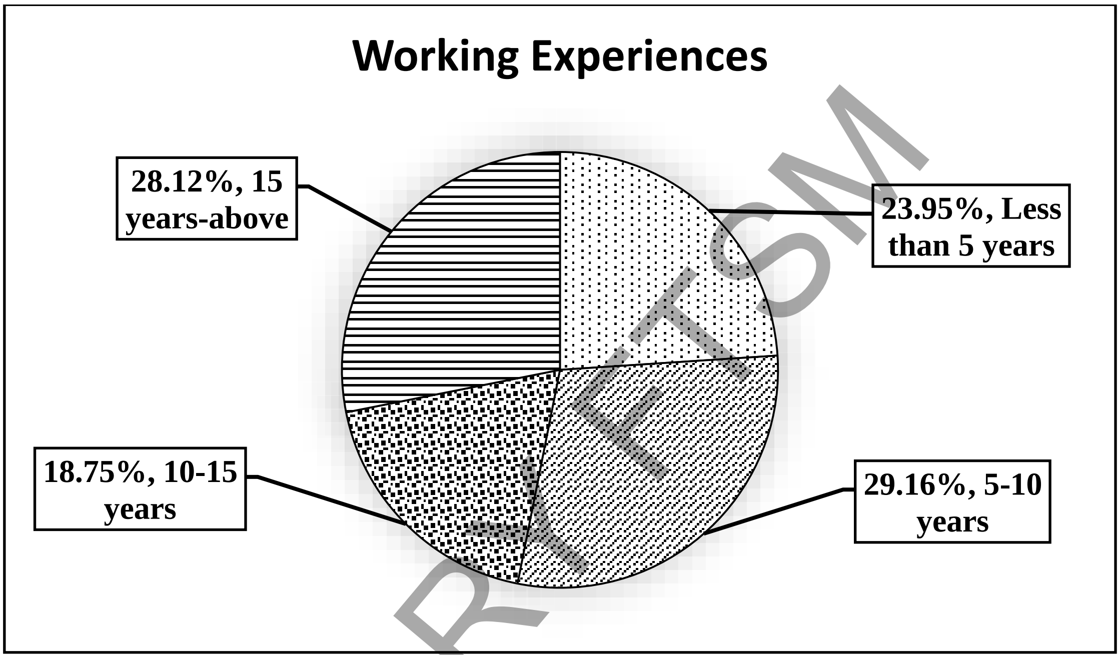

Figure 4.5 Respondent Experience

4.3.1 Findings on the Software Cost Estimation Practice Indonesian Regional Government

The survey on the software cost estimation practice in government project is important to analyze the person who performs the estimation, the time and the most influential factors of the software cost estimation. The result of the software cost estimation practice in government project is illustrated in Table 4.2. The result is based on part B questionnaire in Appendix B analysis.

Table 4.2 Result on the practice of software cost estimation survey

No	Questions	Frequency	Percentage (%)
1	**People who have authority to estimate the software cost:**		
	IT officer	37	38.54%
	Accountant	3	3.12%
	Project Manager	17	17.70%
	Cost Analyst	26	27.08%
	Others	13	13.54%
2	**The software cost estimation is conducted during:**		
	Project proposal phase	83	86.45%
	Feasibility study	5	5.20%
	Scope and Requirement Analysis	5	5.20%
	Later Phases	3	3.12%
3	**The most influential factor of the inaccuracy of software cost estimation.**		
	Scope and requirements are not clear	15	15.62%
	There is no tool to estimate the software cost	36	37.5%
	Not using an available software cost estimation	12	12.5%
	Lack of experiences in previous related projects	28	29.16%
	Lack of top management support	5	5.20%
4	**The frequency of changing the scope and requirements of the project.**		
	Never	3	3.12%
	Seldom	23	23.95%
	Sometimes	52	54.16%
	Often	18	18.75%
5	**The number of changed scope and requirements of a software project.**		
	No	3	3.12%
	A little	66	68.75%
	Moderately	19	19.80%
	Substantially	8	8.33%

a. Findings on the Factor Influence of SCE in terms of The Cost Estimator

The result revealed that the person who has authority to estimate the software cost is information and technology (IT) staff with ratio 38.54%. Then, followed by cost analyst, project manager, others and accountant.

Figure 4.6 illustrates the person who has authority to estimate the software cost.

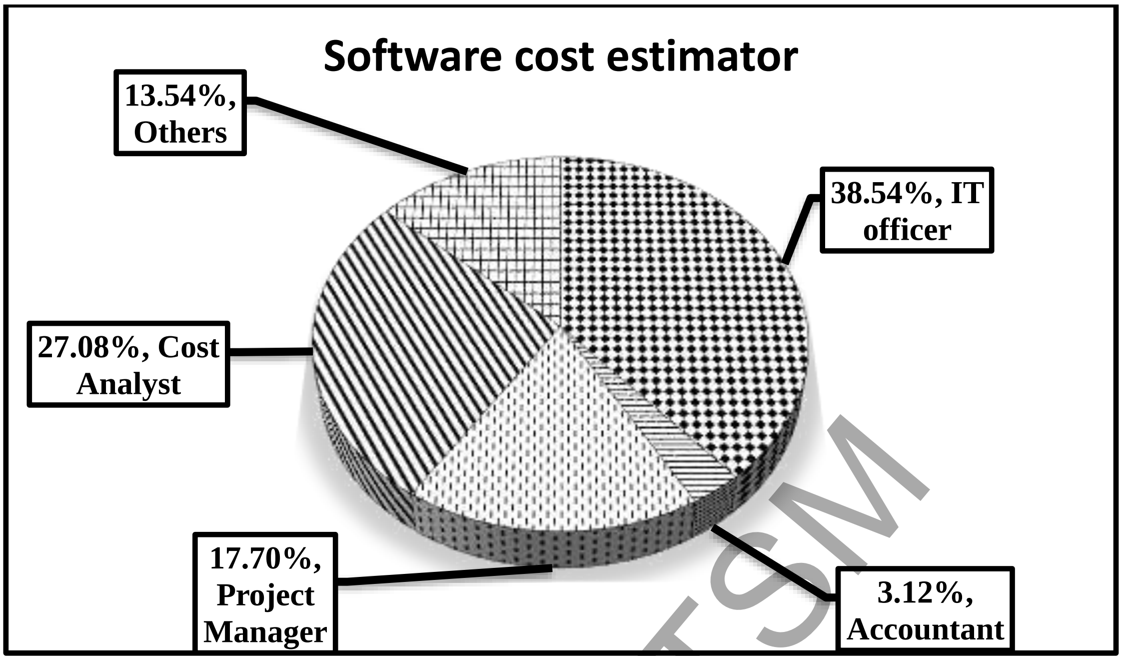

Figure 4.6 Findings on the factor influence of SCE in terms of Software Cost Estimator

Based on the explanation above, the software cost estimation should estimate by cost analyst since it requires experience and knowledge from multi-disciplines. On the other hand, the result shows that most of the estimation is done by IT officer. It also estimates by an accountant, project managers, and others. Thus, it might cause the issue in software cost estimation result, especially the SCE looks from many aspects.

b. Findings on Factor Influence SCE in Terms of Estimation Time

In terms of the software cost estimation time, it is indicated that the software cost estimation in public sector conducted during the project proposal phase with ratio 86.45%. Only a few projects estimated during the feasibility study, scope and requirement analysis as well as the later phases. The estimation time is described in Figure 4.7.

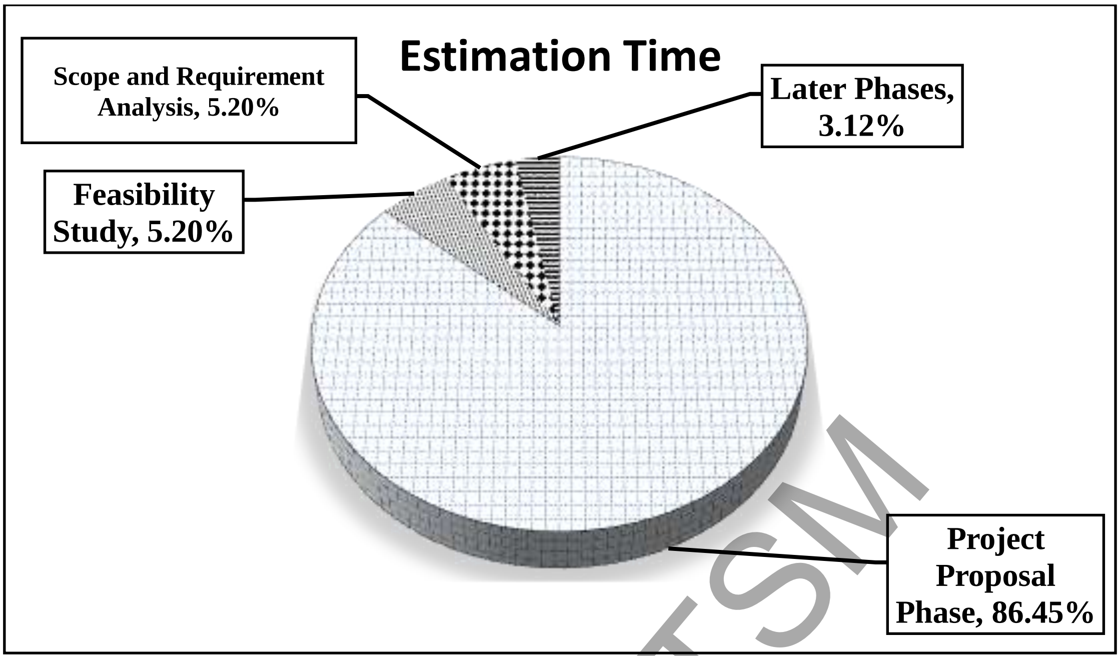

Figure 4.7 Findings on the estimation factor influence the SCE in terms of the estimation time

Hence, based on the result above most of the public sectors conducted the estimation at the early phase of the project which is during the project proposal phase whereby the scope and requirements are not complete. In this case, it just describes the software that will develop in general. So, in order to produce an accurate result, the estimation should be done when the project scope and requirements are completed which has its details information.

c. Findings on The Most Influential Factor in SCE

Based on Figure 4.8, the most influential factor of the inaccuracy of software cost estimation no tool to estimate the software cost with ratio 37.5%. 29.16% the inaccuracy result is caused by lack of experiences in previous related projects. While 15.62% of respondents agree that the scope and requirements are not clear as a factor that causes the inaccurate result. 12.5% of respondents believe that the factor because inaccurate estimation is not using an available of software cost tool. Lastly, 5.2% of respondents agree on the incorrect estimation caused by lack of top management support. So, the most two reasons for the significant factors of software cost estimation are no proper tool to estimate the software cost and lack of experiences in previous related projects see Table 4.3.

Table 4.3 The Most influential Factor in SCE

No	Factors	Percentage (%)
1	No tool to estimate the software cost	37.5 %
2	Lack of experiences in previous related projects	29.16%
3	The scope and requirements are not clear	15.62%
4	Not using an available software cost estimation tool	12.5%
5	Lack of top management support	5.2%

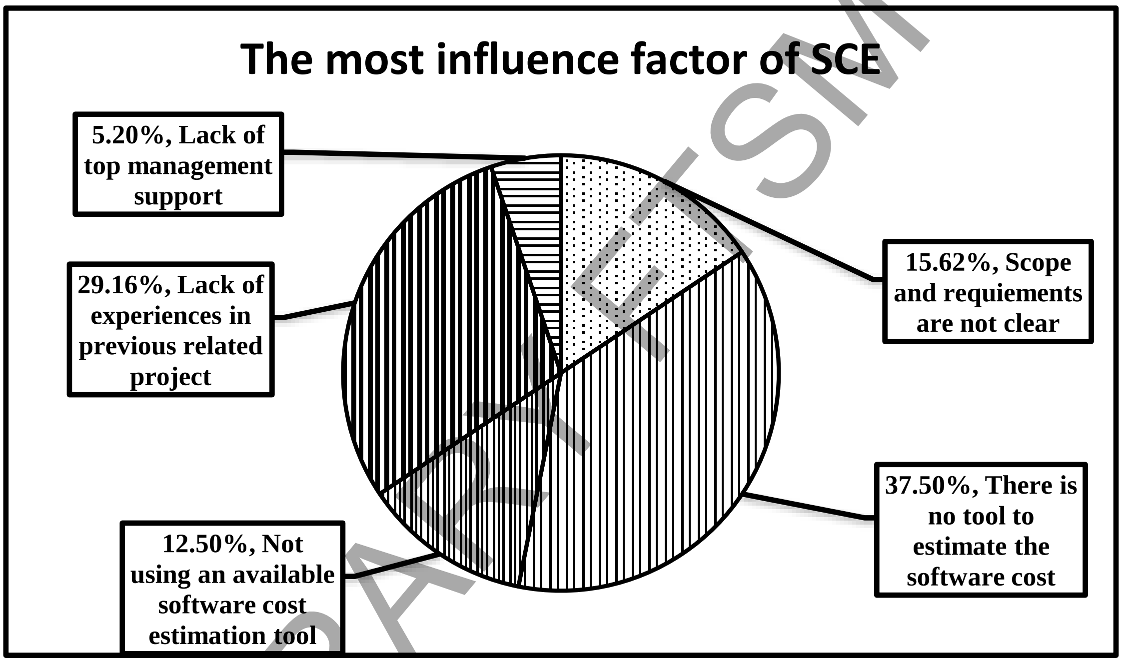

Figure 4.8 Findings on the most influential factor of SCE

Based on the results and the explanation above, the public sector should have a computerized tool that supports the software cost estimation process. Besides having a proper tool, it requires the experience and information of the previous projects.

d. Findings on the Frequency of Changing the Scope and Requirements of project

The result reveals that more than 50 percent of a project often changes the scope and requirements. While 3.12% of the project never changes the scope and requirements and 18.75% of a project often change it. Lastly, 23.95% is seldom made changed the scope and requirements of the project. The frequent changed of scope and requirements of the project is shown in Figure 4.9.

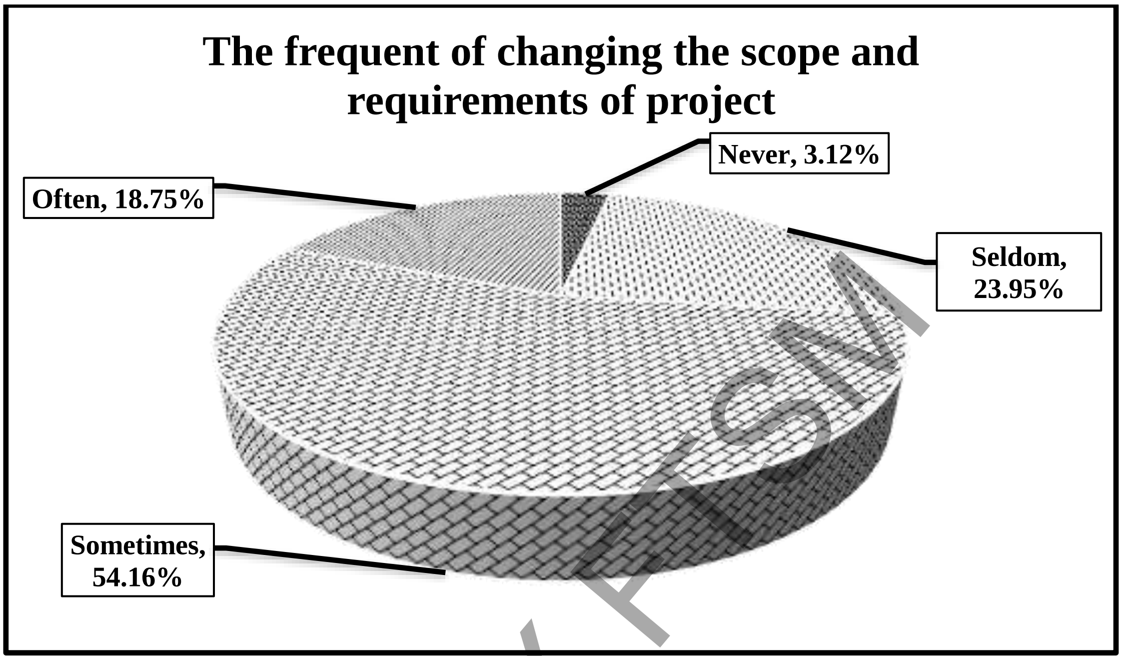

Figure 4.9 The Frequency of Changing the Scope and Requirements of Project

e. The number of change of scope and requirements

The number of changes made in scope and requirements are a little with ratio 68.75%. Furthermore, there is no change made in the scope and requirements (3.12%), and 8.33% of the project has a substantial change the scope and requirements. Lastly, 19.80% of the project is moderately changing the project scope and requirements. Figure 4.10 illustrates the number of changes of scope and requirements.

Although the number of changed scope and requirements of a software project is a little, but yet it also influence the software development process.

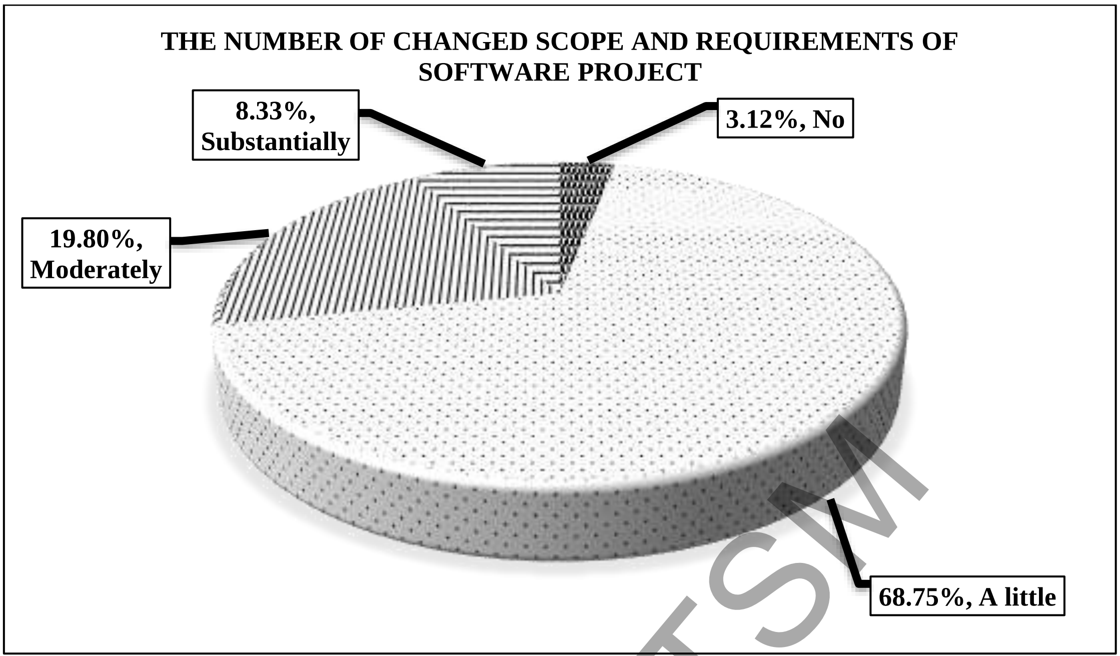

Figure 4.10 The number of changed scope and requirements of software project

4.3.2 Instrument Measurement Validation

Instrument validation is to measure instrument for what it supposed to measure. The instrument measurement is by using Rasch analysis. The result analyses based on summary statistic result (see Figure 4.11).

The summary statistic results use to validate the respondents and instruments quality in general. It also describes the correlation between the person and the items. The validation is analyzed based on the value of Cronbach alpha, person reliability, item reliability, infit and outfit MNSQ, infit and outfit ZSTD, person mean, and item mean.

```
TABLE 3.1 C:\Users\User\Desktop\analysisdata.prn ZOU694WS.TXT  Nov 28  1:38 2017
INPUT: 96 Person  50 Item  REPORTED: 96 Person  50 Item  5 CATS  WINSTEPS 3.72.3
--------------------------------------------------------------------------------

     SUMMARY OF 96 MEASURED Person
-------------------------------------------------------------------------------
|          TOTAL                         MODEL         INFIT        OUTFIT    |
|          SCORE     COUNT     MEASURE   ERROR      MNSQ   ZSTD   MNSQ   ZSTD |
|-----------------------------------------------------------------------------|
| MEAN     201.4      50.0        1.79     .23      1.08    -.1   1.05    -.2 |
| S.D.      16.1        .1         .85     .03       .78    2.7    .76    2.6 |
| MAX.     233.0      50.0        3.76     .30      5.44    9.9   5.48    9.9 |
| MIN.     157.0      49.0        -.04     .17       .25   -4.7    .23   -5.0 |
|-----------------------------------------------------------------------------|
| REAL RMSE    .27 TRUE SD     .81  SEPARATION  3.05  Person RELIABILITY  .90 |
|MODEL RMSE    .23 TRUE SD     .82  SEPARATION  3.55  Person RELIABILITY  .93 |
| S.E. OF Person MEAN = .09                                                   |
-------------------------------------------------------------------------------
Person RAW SCORE-TO-MEASURE CORRELATION = .99
CRONBACH ALPHA (KR-20) Person RAW SCORE "TEST" RELIABILITY = .92

     SUMMARY OF 50 MEASURED Item
-------------------------------------------------------------------------------
|          TOTAL                         MODEL         INFIT        OUTFIT    |
|          SCORE     COUNT     MEASURE   ERROR      MNSQ   ZSTD   MNSQ   ZSTD |
|-----------------------------------------------------------------------------|
| MEAN     386.8      96.0         .00     .17      1.01     .0   1.05     .1 |
| S.D.      26.8        .1         .71     .02       .32    2.0    .36    2.2 |
| MAX.     444.0      96.0        1.57     .21      1.86    4.8   1.98    5.1 |
| MIN.     315.0      95.0       -1.86     .13       .41   -4.6    .42   -4.7 |
|-----------------------------------------------------------------------------|
| REAL RMSE    .18 TRUE SD     .69  SEPARATION  3.91  Item   RELIABILITY  .94 |
|MODEL RMSE    .17 TRUE SD     .69  SEPARATION  4.16  Item   RELIABILITY  .95 |
| S.E. OF Item MEAN = .10                                                     |
-------------------------------------------------------------------------------
UMEAN=.0000 USCALE=1.0000
Item RAW SCORE-TO-MEASURE CORRELATION = -.99
4799 DATA POINTS. LOG-LIKELIHOOD CHI-SQUARE: 8701.22 with 4651 d.f. p=.0000
Global Root-Mean-Square Residual (excluding extreme scores): .6394
```

Figure 4.11 Summary Statistic Result of the Overall Factors

a. Cronbach Alpha

Based on the result above, the Cronbach alpha value is 0.92 which shows that the correlation between item and person is very good. It reveals that it has a high reliability and a high consistency in the raw score (instrument) in this study.

b. Person Reliability

The person reliability value is 0.90 that indicates that very good ability spread of the sample involved in this study. Hence, the respondent is qualified in responding to this study. The mean person given is 1.79 which means most of the respondents tend to agree with many items because the mean values are greater than 0.00. The mean logit is indicating that overall the respondents agree that those factors affect the software cost estimation result in the public sector. Besides that, if the person mean is higher than the item means, then the whole test meets the expectation of this study. So, the person reliability value and the comparison between the person mean and item mean touse for personability measure in this study.

c. Item Reliability

The item's quality in the questionnaire is very good with value 0.94. It indicates that if the test given to a different group of respondents, the possibility of the item difficulties is still the same. Consequently, it can be concluded that the item difficulties among the items are spread well.

d. Infit and Outfit MNSQ

The infit use to identify the unexpected respond given by the respondents near the level capability of the respondent. The outfit is to consider the expected answer, and the actual answer given by the respondent which shows how far away the item being agreed is the personability. Furthermore, the infit and outfit MNSQ person are 1.08 and 1.05. Furthermore, the infit and outfit MNSQ item are 1.01 and 1.03 which have a good value since the mean-square fit statistic value should be between 0.50 and 1.50 so that the items are not easy to guess or predict the answer. The ideal value of MNSQ is 1, so the value of the result above is close to the ideal value. Thus, it also indicates that it is productive for measurement (Bambang & Wahyu 2014).

e. Infit and Outfit ZSTD

The data also has the reasonable logic if the value infit and outfit falls in $-1.90<y<1.90$ (Bambang & Wahyu 2014). Based on the result above, infit and outfit ZSTD person is -0.10 and -0.20, while the infit and outfit ZSTD item are 0.00 and 0.10. Hence, those value are close to the ideal value of infit and outfit ZSTD which is 0. It indicates that data fit the model that can measure what is supposed to be measured.

f. Item Separation

The separation value gives a clue the quality of the separation between person and items. The separation is good when the value of separation is high which means the quality instruments are better too. The value of the separation is 3.05. The separation of the respondent group can also be seen by using the following formula (see Figure 4.13). As a result, it shows that there are four group of respondents.

H = [(4 X person separation) +1]/3

= [(4X3.05)+1]/3

= 4.4

= 4

Figure 4.12 Item Separation Formula

g. Item Dimensionality

The item dimensionality (see Figure 4.13) is 29.4% which is higher than 20%. It reveals that the instruments are able to measure that supposed to measure. Hence, it fulfills the item dimensionality requirement. Moreover, there are also value unexplained variance value which has ideal result since it is not more than 15%. The value is 11.1% and the others value 10% (Bambang & Wahyu 2014).

```
TABLE 23.0 C:\Users\User\Desktop\analysisdata.pr ZOU694WS.TXT  Nov 28  1:38 2017
INPUT: 96 Person  50 Item  REPORTED: 96 Person  50 Item  5 CATS  WINSTEPS 3.72.3
--------------------------------------------------------------------------------

     Table of STANDARDIZED RESIDUAL variance (in Eigenvalue units)
                                                  -- Empirical --    Modeled
Total raw variance in observations     =        70.8 100.0%          100.0%
  Raw variance explained by measures   =        20.8  29.4%           32.1%
    Raw variance explained by persons  =         7.3  10.3%           11.2%
    Raw Variance explained by items    =        13.5  19.1%           20.8%
  Raw unexplained variance (total)     =        50.0  70.6% 100.0%    67.9%
    Unexplned variance in 1st contrast =         5.5   7.8%  11.1%
    Unexplned variance in 2nd contrast =         4.4   6.2%   8.8%
    Unexplned variance in 3rd contrast =         3.4   4.9%   6.9%
    Unexplned variance in 4th contrast =         3.0   4.2%   5.9%
    Unexplned variance in 5th contrast =         2.7   3.8%   5.4%
```

Figure 4.13 Item Dimensionality

h. Scalogram

The scalogram result reveals that the consistency answer given by the respondents. As shown in , the respondent 66 and 43 should answer four and five instead two and three. They are the competence respondent, but they might be simply ticked their answer. In the scalogram, it can identify the item from the easiest item to the most difficult items to be endorsed and the most competence respondent to the less competence respondent which illustrate in Figure. It also can check the careless ans wer given by the respondents, while they fall in the most competence category.

```
TABLE 22.1 C:\Users\User\Desktop\analysisdata.pr ZOU978WS.TXT  Dec  1 16:26 2017
INPUT: 96 Person  50 Item  REPORTED: 96 Person  50 Item  5 CATS  WINSTEPS 3.72.3
--------------------------------------------------------------------------------

GUTTMAN SCALOGRAM OF RESPONSES:
Person |Item
       |111 21 22 211344111244 2424433 2 23333 354 421343
       |13445551678673560899729042832164135498860031720972
       |--------------------------------------------------
    66 +55555555555555555555445545545555455554544544544454 2  4
     3 +55555555555555555555545355555555555434443545444553444  5
    26 +5555555555555555555554535555555555543444354544553444  5
    91 +5555555554555455455544455544445555555545444444454454  5
    20 +5555555554454554555554454443455445555555555444445445  4
    18 +555555555455555544455553545555544355555545455545332  4
    43 +55555555555555555555545554545555555555552445432524 2  5
    62 +55455455555555555555545555555555455554353445534324 3  5
    79 +54445444444344543434333333333433343454332444333332  3
    55 +44422444344434344422444242444422423433444423444444  4
    70 +45344434334542334333444232444333323332443533452344  3
    71 +54444424333544333444234334344334332332244434433244 4  3
    49 +44434333234344433333434432443432223433444423443444  2
    54 +44422444244434344422434242424442242343344442234443 4  4
    75 +55433423344422342333443243434322233323334323443323  2
       |--------------------------------------------------
```

Item Numbers

The easiest item -The most difficult items to

The careless answer given by the respondents

The most competence -The less

Figure 4.14 Guttman Scalogram

i. Person Map

The person map (see Figure 4.15) indicates the spread of person and item according to their respective ability measure. Most of the respondents tend to agree that those factors have a significant impact on the software cost estimation result accuracy in the public sector. The personability is divided into three categories which are the most competence, moderate competence, and less competence respondent. It is divided based on the mean logit (1.79) which shown in Figure 4.16 above. So, the most competence respondent position is above the mean logit, while the moderate and less competence is the mean logit.

Among the most competent respondents, the respondent in the maximum is 66MVBHE which means that the respondent is a male who is 41-above years old. The highest educational level is bachelor degree who has the position as a section head and has experiences for 15 years and above. Therefore, it shows that he is the respondent with the most capable of estimating the software cost. The map above reveals that he agrees that those factors are essential for the software cost estimation so that the project successful.

The person map shows that 40.62% of the most competence respondents, 45.83% of the moderate competence, 13.54% of the less competence respondents.

```
TABLE 16.3 C:\Users\User\Desktop\Overall Data.pr ZOU897WS.TXT  Dec  1 17:43 2017
INPUT: 96 Person  50 Item REPORTED: 96 Person  50 Item  5 CATS  WINSTEPS 3.72.3
--------------------------------------------------------------------------------

       Item - MAP - Person
<rare>|<more>
    4        +
             |  66MVBHE
             |  03MVBHE  26MOPSE  91MVPCE
             |T 18MYSSE  20MYDSE  43FOPSE
             |  62MOBHE  64MOBHE
             |  63MOBHE
    3        +  04MOPSE  30MOPCE  84MYPHE
             |
             |S 08FOPBE  19MOPHE  32MOPHE  82FVSSE
             |  05FYBCE  11FVPCE  15FOPHE  40MOSSE  61MOPHE
             |  29FYBSE  42MYSSE  52FVBHE  65MVBDE
             |  90MOBHE  94FOBSE
    2        +  13MOBSE  22MOBHE  23FYBSE  50MOBCE
             |M 02FYBSE  12MOBHE  17MOPHE  31MOBSE  34MOBSE  47FVBHE  78FYBSE
             |  01FYSSE  39MODBE  41FVBCE  44MYBSE  45FVBHE  46MOBSE  68MODSE
                69FOPHE  74MOBHE  76MOPHE  81MVPCE  83MVBBE  88MYBSE
          XX T| 07MOPHE  10MOPHE  24MODSE  25MYBSE  27FYBSE  28FYBSE  38MYBSE
                53MVSSE  56MVSHE  72MVBHE  73FOBHE  86MOPHE  92MVBCE  95MODSE
             |  33FOBSE  36MOBHE  48FODSE  77FYDSE  85MOBCE
             |  06FYBSE  16FVPCE  51MYDSE  57FVPCE  58FVPCE  59FVPHE  60MVPBE
                87MOPHE  93FVDCE  96FOBSE
    1     XXXX  +S 09FOPHE  89FOBSE
          XXX   |  37FYBSE  67FVPCE  80FOBSE
              S|  14FOBHE  21FOPCE  35FOSHE  79MYSSE
          XXXX  |  55FOPHE
            XX  |  49FYBSE  70MVPHE  71FVBCE
        XXXXXX  |T 54MVDHEs
    0   XXXXXX M+  75MOPSE
       XXXXXXX  |
          XXXX  |
          XXXX  |
            XX S|
             X  |
   -1       XX  +
             X  |
                |
               T|
             X  |
             X  |
   -2           +
<frequ>|<less>
```

Item Free Person

Most Competence

Moderate competence respondent

Most Competence Respondent

Person

Free Item

Figure 4.15 Person Map

As a conclusion, based on the instrument validity analysis above it proved that the instrument is able to measure what supposed to be measured in software cost estimation factors, influence the software cost estimation accuracy result. Besides that, the correlation between the item and the person also good which has a high consistency.

4.3.3 Findings on the critical factors that significantly impact software cost estimation in developing software in Indonesian regional government context.

Critical factors that significantly impact software cost estimation in developing software in Indonesian regional government context identify by using person distribution map in Rasch model. The analysis conducted based on the part C of the questionnaire in APPENDIX B.

Furthermore, the main factor of the software cost estimation is divided into four categories which are technology, people, process and organizational factors. The purpose is to examine the critical factors that significantly impact the software cost estimation accuracy. First, the technology factor uses to measure the impact of the proper tool availability and usability in estimating the software cost. Second, the people factor is to examine the effect of the software cost estimation that influential by people who are involved in the project. Third, process factor is to measure the software cost estimation in public sector and its influence on the software cost estimation. Lastly, organizational factor is to examine the effect of the environment on the software cost estimation.

Furthermore, identifying the critical factor is analyzed by using the person distribution map which shown in Figure 4.16. Based on the person distribution map, the item is divided into three categories which are the easiest items being endorsed by the respondents, the moderate items being endorsed by the respondents, and the most difficult items being endorsed by the respondents. These are divided based on the mean logit item which is 0.00, and the standard deviation is 0.71. The most difficult items being endorsed by the respondents fall above the mean logit (0.00). Second, the moderate items being endorsed by the respondents fall between -0.71<logit<0.00. Lastly, the easiest item being endorsed by the respondent falls in -1.42<logit<-0.71.

Based on people distribution map, there are six critical factors influence of software cost estimation in public sectors. The result is determined by the most items that being endorsed by the respondents which is located below standard deviation (S) value (see Figure 4.16). The first critical factor is the I0011 item (programmer capability significance towards the project), I0013 (Top management support is essential for the project to be successful), I0014 (Top management is understood the objectives of the project), I0025 (Risks that occurs during the software development project manage well), I0004 (The project manager is knowledgeable and competence in ICT), and I0015 (Top management is involved and committed to the project).

It reveals that the easiest items consist of six factors (see Table 4.3) which indicates that those items are the critical factors significantly impact the software cost estimation accuracy in developing software in Indonesian regional government context. The most critical factors of the software cost estimation depend on the people and process factors. It indicates that if the public sector has a good proper tool for software cost estimation and good environment to estimate the software cost, but the people is not competence and the process is not effective. It can lead to the inaccurate result of the software cost estimation. Hence, the people who are involved in estimating the software cost, as well as people who are involved in the software development project, should be competence, knowledgeable and have a good experience. Besides that, the process of estimating the software cost should be effective and efficient so that everything will be well managed and organized.

Table 4.4 Findings on the critical factors that significantly impact software cost estimation in developing software in Indonesian regional government context

Item Number	Item Measure	Factors	Main Factor	Sub-Factor
I0011	-1.86	Programmer capability significance towards the successful project.	People	Personnel/Team Capabilities
I0013	-1.65	Top management support is essential for the project to be successful.	People	Top Management

to be continued…

…a continuation

I0014	-1.11	Top management is understood the objectives of the project.	People	Top Management
I0025	-1.04	Risks that occurs during the software development project manage well.	Process	Risk Management
I0004	-1.04	The project manager is knowledgeable and competence in ICT	People	Project Manager
I0015	-0.84	Top management is involved and committed to the project.	People	Top Management

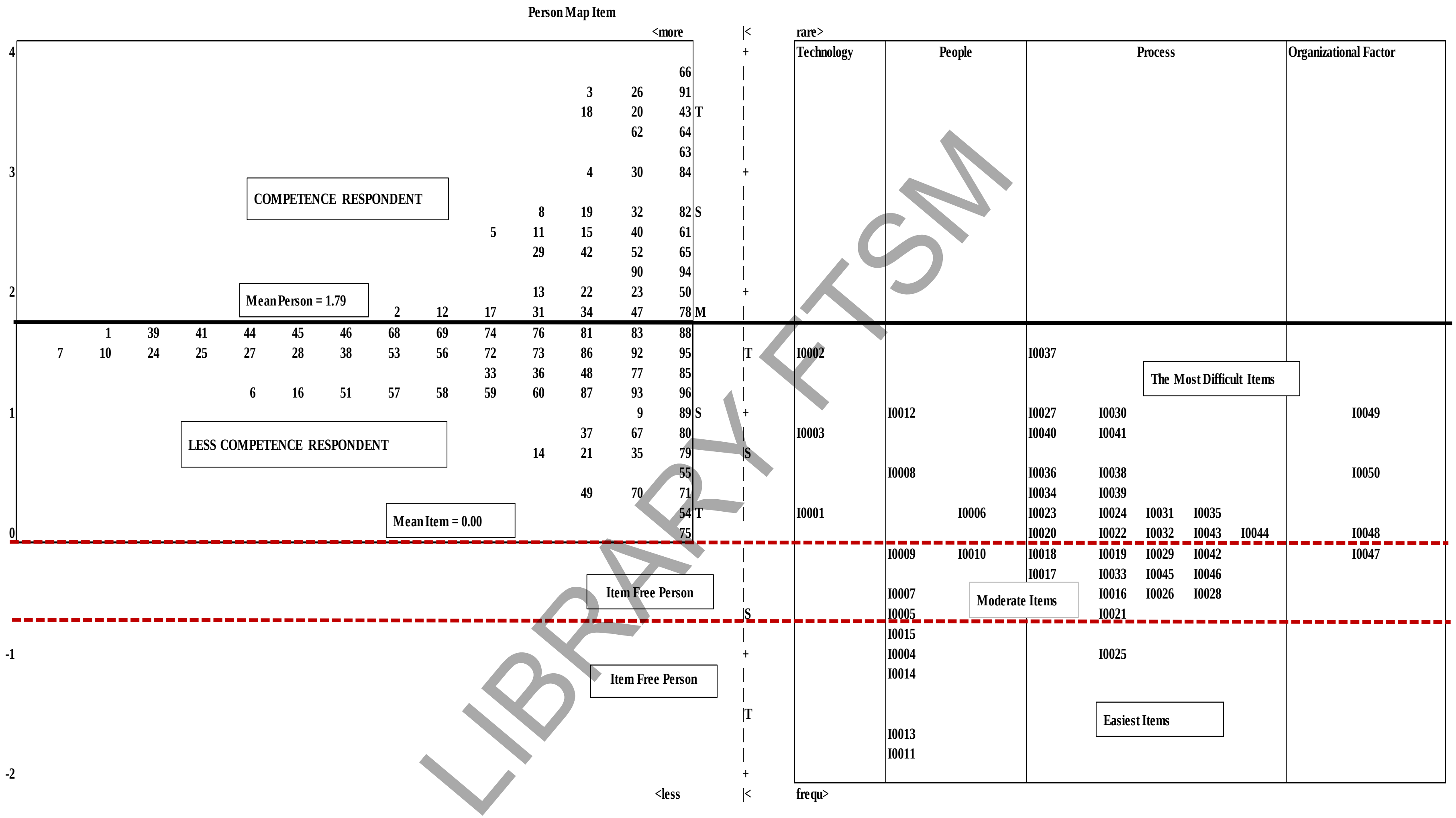

Figure 4.16 Person Distribution Map

According to Table 4.4, the first critical factor is *the programmer capability significance towards the successful project*. It indicates the respondent who is involved in the project development understand that the important role of the programmer in the software development which has a big impact on the software cost estimation. This can be seen by the fact that the progress and the accomplishing of the project depend on the programmer due to the programmer is the one who develops the software. Hence, if lack of skill, knowledge, and experience of the programmer, it can cause the project delay which means the project is not completed within the time that has been estimated before. As a result, it might increase the cost which can be increased from the operational cost and so forth.

Furthermore, the second factor *top management support is essential for the project to be successful* under the people factor too. Based on the scalogram above, most of the respondent tends to choose 4 (agree) and 5 (very agree) which reveals that the top management support is necessary for the project. This can be reasoned by the fact that the role of top management in decision making in the organization. The top management is the one who has power in decision making, in other words, the top management has the power to approve or to reject a project as well as the software cost estimation result.

The third factor is still under the people factor which is *"top management is understood the objectives of the project."*It shows that the people who are involved in the project believe that the top management understood the objectives of the software development. Hence, it shows that the top management sees the necessity of a particular software to enhancing the effectiveness of the agency activities. As a result, the top management approves the software cost estimation although it requires a high cost.

The fourth factor is under the process factor specifically under the risk management which is *"Risks that occurs during the software development project manage well."*. The software cost estimation has many uncertainties. Hence many risks might occur during the software development project. So, if the risks occurred do not manage well, it has the possibility of a project to be a failure which might cost need the additional cost or project cannot finish within the time estimation. According to Potdar et al. (2014), the risk management is vital for identifying, managing and eliminating the software risk that becomes threats to the project successful. There are many types of risks need take into consideration such as people risks, size risks, process risks, technology risks, tools risk, organizational and managerial risks, estimation risks and so forth. The risks can be identified by examine of the previous projects, brainstorming, analogous conditions of the project, and so forth.

The fifth factor is *"the project manager is knowledgeable and competence in ICT"* which is under the people factor. It is reasonable to factor that influences of the software cost estimation result due to even though the software is developed by the programmer, the whole project is managed and controlled by the project manager so that it is on track with the estimation. The estimation includes the time, cost and the quality of the software which has been stated in the scope and requirements of the project.

The last critical factor that significantly impacts the software cost estimation is under people as well which is *"top management is involved and committed to the project."* it reveals that the respondents understand that the importance of top management involvement in the project influence the software cost estimation. If the top management actively involved and committed to the project, it will assist the top management to understand about the project and to know the progress of the project. Hence, the top management and project manager can collaborate in managing the project so that it can be completed within the cost and time that has been estimated before. Besides that, it also important to ensure whether it fulfills the scope and requirements of the project or not. As a result, all the project objectives are achieved. As stated by Potdar et al. (2014), the top management commitment is substantial for the software cost estimation that determines the success or failure a project during the development and implementation of the project.

4.4 THE QUALITATIVE APPROACH RESULTS ANALYSIS

In this study, it is also using the qualitative approach. For the qualitative approach, it consists of five respondents who are working from the different agency in West Sumatera province. The qualitative method is done by interviewing those respondents with consists of seven questions. The questions (see APPENDIX C) of this approach are used to answer the first, second and fourth of research questions. For the result analysis, the respondent name is concealed. Hence, each respondent will have the initial name as follows:

- Respondent 1: Dayat is the section in *DISKOMINFO* who have experience for 15 years. The highest educational level is Bachelor degree.

- Respondent 2: Aan has working experience for five to10 years in LPSE. His educational level is Bachelor degree.

- Respondent 3: Rizki has the position as the secretary at *DISKOMINFO* who have been working for more than 15 years. The highest educational level is postgraduate.

- Respondent 4: Leni is the staff of government agency in Baperlitbang who is above 41 years old. The highest of the educational level is Bachelor's degree who has been work for above 15 years. The current position is the section head.

- Respondent 5: Zainal has been working in *DISKOMINFO* for 10 to 15 years with the current position is as the section head. The highest educational level is bachelor degree.

4.4.1 The government employees understanding of the software cost estimation concept

According to the four respondents defines that software cost estimation is used to estimate the cost that is required to develop software. While the third respondent is Aan, who work in LPSE, explains software cost estimation is like estimating the cost required to develop a software which has the indicators for predicting the cost. However, those indicators still are not specified yet.

So, based on the statements above, it can conclude that the understanding on the software cost estimation is not clear because the software cost estimation is not only about the cost, but also the schedule and the resources require to complete the project. Rosmala & Akbar (2010) defines that the software cost estimation is a process of predicting human resources, time and cost requirements to complete a project. The most crucial part in software cost estimation is in estimating the efforts required to complete the project since it requires the parameters to measure it. Huang et al. (2007), software cost estimation consist of predicting the cost, quality, risk analysis and other factors that influence of software cost estimation. Furthermore, according to Sharma, Bajpai & Litoriya (2012) stated that software cost estimation is a process to estimate the resources, schedule, software size, effort and the whole cost of the project. So, the software cost estimation is not only about estimating the cost required to complete the project. However, there are aspects that need to estimate too which are important for the software development.

4.4.2 Findings on the parameters used for estimating the software cost in Indonesian regional government.

According to Dayat, the parameter uses to estimate the software cost is the level of difficulties the programming language and the user quantity who will use the software. Also, Rizki describes that the parameter contains the office stationery, hardware, software, man-months, transportation, 15% profit of the IT consultant from the total cost and 11.5% tax which include income tax (1.5%) as well as value added or good and services (GST) tax (10%). While for maintenance, it is under warranty of the IT consultant for six months. While for the next maintenance, it includes in the next year budget. Likewise, Leni defines the parameter consist of the operational cost, maintenance, specification of the software and the effectiveness of the result of the project. Lastly, Zainal said that the input to estimate the software cost is the level difficulties of the software being developed.

Based on the explanations above, the inputs use to estimating the software cost are project complexity, the number of users, the level of difficulties the programming language, hardware, man-months, transportation, tax and the project requirements.

4.4.3 Findings on the software cost estimation practice Indonesian regional government.

a. The software cost estimation process

Dayat said that the software cost estimation is calculated by the system analyst according to the factors influence the software development. While according to Aan, the people who involve in estimating the software cost are head of the department, head of division, sector chief, section head and financial officer because of their responsibility on the agency expenditures. While IT staff did not have right to make the decision, they just allow giving a recommendation for the software development project. The decision making decided by the head of the department. Another problem in government, other staff who do not have IT background, they tend to generalize IT staff capability which means that they understand everything. In fact, system analyst, programmer, and database administrator, they have different skill with from each other. Thus, the IT consultant often becomes the system analyst. The software cost estimation is conducted one year before the project begins, however, the scope and requirements not clear yet.

Likewise, Rizki who work in *DISKOMINFO* province stated that the IT staff is knowledgeable due to most of their educational level are postgraduate. The IT staff responsible for estimating the scope and requirements of a project such as a server, capacity and so forth. After that, the decision made by procurement of good and