

ATLAS (Application for Tracking and Scheduling) as Location Guide and Academic Scheduling at Campus YSU (Yogyakarta State University)

Pradana Setialana, Muhamad Rizki Fajri, Abdul Rachman Pambudi, and Ratna Wardani, *Member, IACSIT*

Abstract—Yogyakarta State University is an educational institution that has six faculties with wide area. YSU currently has maps in digital form accessible through website and the form of posters which were placed in several locations. However, the digital map provided less detail and little bit features. Therefore, we need a map of YSU in digital form that can be accessed by portable through smartphone devices. This study resulted in a prototype ATLAS software that can be used as a solution to these problems. This application can help all parties in the search location, place and space in the campus area. ATLAS can show (User) student, lecturer and visitor to know the position in YSU. Moreover, it also provides the position information of users (user) other than a digital map in campus environment of YSU either by GPS or location based on the academic schedule. These results indicate, on testing functionality of ATLAS Software cannot be injected with a SQL injection or Cross - Site Scripting. Further On Reliability testing maturity index value of 0.961, there is software development until maturity index values approaching 1.0. Testing Efficiency using Web Page Testing known ATLAS software load time 4.972 seconds. While the testing maintainability index value of 87.725. So the ATLAS software has met the four standards of software quality based on the characteristics of the ISO 9126 functionality, reliability, efficiency and maintainability.

Index Terms—GPS, ISO 9126, maps, schedule.

I. INTRODUCTION

Yogyakarta State University (YSU) is an educational institution that has six faculties with wide area. A large area causing student, lecturer, guest and visitor of YSU has difficulty to find space and buildings in the campus. In addition to causing difficulty to find space, students and lecturers have to communicate anytime with each other for asking their positions in campus. YSU currently has maps in digital form accessible through website and the form of posters which were placed in several locations. However, the digital map provided less detail and little bit features.

Technological developments led to the current map is presented digitally so it can be accessed through electronic devices anywhere and anytime. Digital maps are usually integrated with smartphones, GPS and Internet to determine its own position and others at a given location. However, these technologies have not been used YSU in digital maps.

Manuscript received December 20, 2013; revised April 2, 2014.

The authors are with Department of Electronics Engineering Education, Faculty of Engineering at Yogyakarta State University, Indonesia (e-mail: pradana.setialana@gmail.com, rizki_newlive@yahoo.com, abdulrachman.pambudi@gmail.com, ratna@uny.ac.id)

Therefore, we need a map for YSU in digital form that can be accessed by portable through smartphone devices. Digital map can facilitate student, lecturer and visitor to know the position in YSU. In addition, digital map also provides information of other user position in environment of campus either by GPS or location based on the academic schedule.

II. MODEL ANALYSIS

A. Interface Design

1) Use case program

On the use case diagram of The ATLAS contained two actors that students and lecturer who each has a functionality that can be accessed by any actor (see Fig. 1).

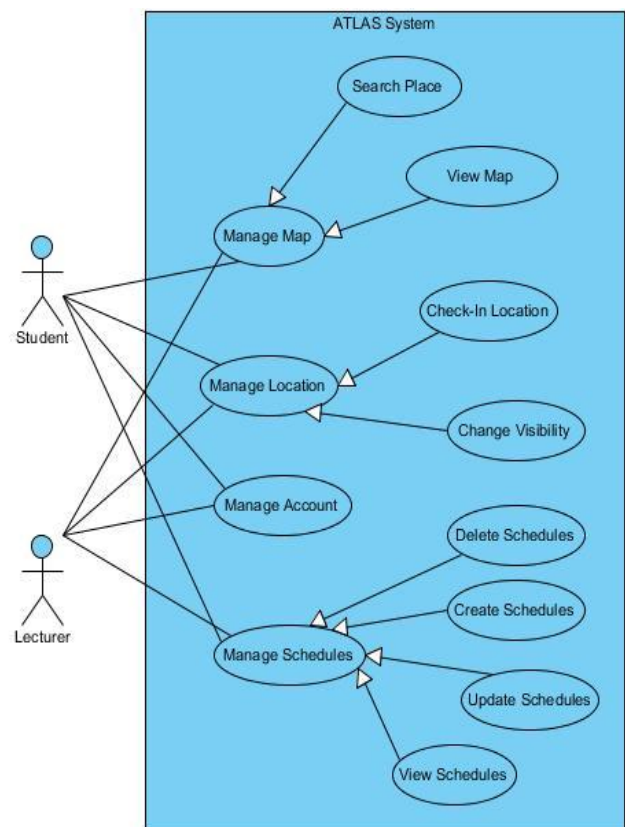


Fig. 1. Use case software ATLAS.

2) Task analysis program

Analysis of the tasks and functions performed by the application in accordance with the following chart picture Fig. 2:

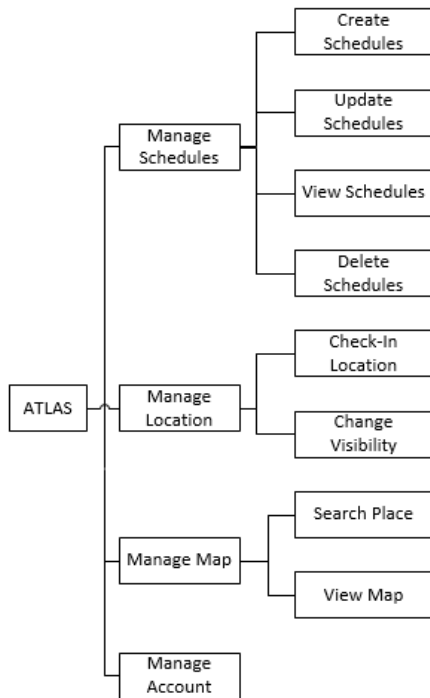


Fig. 2. Chart of task analysis program.

III. DESCRIPTION AND DEVELOPMENT STAGES

The steps are taken in the development of software, they are as follows: (1) needs analysis, (2) design of the model, (3) implementation (coding), (4) testing.

A. Need Analysis

Stages of the analysis phase is to identify and obtain data on the needs of what is required in the design and implementation of systems and thinking to the next design. According to Pressman (2002:319), the analysis is the process of discovery, improvement, and specification. In this stage, the identification of the data required for system development. Steps are taken in the analysis phase include: a) the analysis of the technical specifications, namely: what are the analytical capabilities needed by the system, b) analysis of the system, namely: an analysis of the performance of the work will be performed by the system being designed or built. [1]

B. Design Phase

At this stage will provide output in the form of application design to be built. Design can be architectural design provides an illustration to explain the structure of the program modules and present the relationships among modules. Interface design is a display design in order for the user easily understand the purpose of the application software maker. With the interface, the user does not need to go too far to understand the program. Design flowchart is a picture of how these processes are derived from user input will be processed in a software system.

C. Implementation Phase

Implementation phase will translate the modules that have been planned and designed prior to the application form by using a specific programming language and incorporate them into a more perfect union system so that it becomes operable applications. The steps that have to be done in the

implementation are: (a) collect and choose the modules that will be used, (b) determine the program needed to support programs that has been designed, (c) translating procedure, subroutine and functions of modules to the programming language, (d) unify procedure, subroutine, and functions of the modules that have been made into a unified program.

D. Testing Phase

Application testing is done to find errors that may occur before the programs are implemented. Tests performed on each module and proceed with testing for all the modules that have been strung together. Testing in the field of engineering includes Alpha Testing and Beta Testing. Alpha Testing is a process of testing and revision of the program while making the program. This process includes an assessment of the aspects of the media and programming aspects.

IV. RESULTS AND DISCUSSION

A. Result

Based on the design of the software design which has been made in the design phase, the implementation of the interface of the ATLAS app will look like the following pictures Fig. 3:

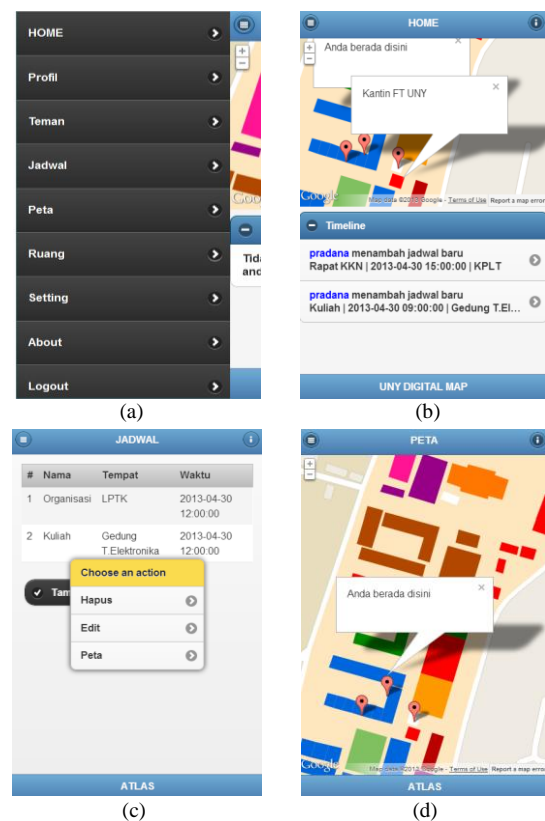


Fig. 3. Picture (a), (b), (c) and (d) is screen captures of software ATLAS.

B. Analysis of Software Quality

1) Functionality testing results

Testing software functionality according to Huang (2003) includes testing SQL Injection and Cross - Site Scripting (XSS) [2] so that ATLAS software to be able to meet the quality of software components must be able to counteract the functionality of SQL injection and XSS. The following

functionality testing phase ATLAS (Application for Tracking Location and Scheduling).

a) *SQL injection*

To test for SQL injection can be done by entering SQL injection script on the input username and password to log in and can be done by using software SQL Injection Me. ATLAS was tested either by software or manually enter SQL injection script. The test results obtained from the test results in the Table I below.

TABLE I: RESULT OF SQL INSPECTION

No	SQL Injection String	Pass/Fail
1	admin' --	Pass
2	admin' #	Pass
3	admin'/*	Pass
4	' or 1=1--	Pass
5	' or 1=1#	Pass
6	' or 1=1/*	Pass
7) or '1='1--	Pass
8) or ('1='1--	Pass

Information passed in the Table means that the program cannot be used via script injection. While the criteria means fail to succeed on the injection software using the script. From the table it can be seen that ATLAS is able to ward off all eight existing script injection SQL Injection on Me.

b) *Cross-site scripting (XSS)*

For testing the ATLAS can counteract Cross - Site Scripting attacks can be done by testing the XSS - scanner web application that can be accessed through <http://xss-scanner.com>. Because XSS - scanner software requires a system online test is done on the server from ATLAS on the internet. Here are the test results by using XSS - scanner (see Table II).

TABLE II: XSS TESTING FOR ATLAS

Variabel	Result
Op	Ok
username	Ok
password	Ok
Idtipe	Ok
no_induk	Ok
nama	Ok
Telepon	Ok
Email	Ok
Idjurusan	Ok

According to the Table II is known that XSS - scanner to extract all the existing functions and test and showed that all the functions secure against XSS.

2) *Reliability testing results*

ATLAS system using PHP, CSS, JavaScript and HTML where each module is separated in files categorized in different folders. In this program there are two categories of modules are android and server assets. Here the number of modules in each folder Table III.

TABLE III: NUMBER OF MODULES IN THE ATLAS

Category	Number of Moduls
Android assets	42
Server ATLAS	9
Total	51

To determine the number of modules during development by looking at the version control when developing software. On software development, there are two changes to last used as a benchmark to test. The first change is the addition of a message that can chat to other users. The addition of this feature cause changes to the modules which can be seen in the Table IV below.

TABLE IV: NUMBER OF INITIAL MODULE CHANGES

Category	Amount of change	Amount of addition	Amount of removal
Android Assets	1	2	0
Server ATLAS	0	1	0
Total	1	3	0

Of these changes can be calculated software maturity Index of ATLAS by using equation proposed by Pressman (2001) [1]. Here is the calculation software maturity index (SMI) at ATLAS.

$$SMI = [MT - (Fa + Fc + Fd)] / MT \tag{1}$$

$$SMI = [51 - (3 + 1 + 0)] / 51$$

$$SMI = 0.922$$

From these calculations is known that the SMI on first change is 0.922. To see whether the SMI close to 1.0, tested again at the last change. Last Modified software is the preparation of the menu layout.

TABLE V: NUMBER OF INITIAL MODULE CHANGES 2

Category	Amount of Change	Amount of addition	Amount of removal
Android assets	2	0	0
Server ATLAS	0	0	0
Total	2	0	0

From the Table V Software Maturity Index can be calculated using the formula described by Pressman (2001). The following calculations.

$$SMI = [MT - (Fa + Fc + Fd)] / MT \tag{2}$$

$$SMI = [51 - (0 + 2 + 0)] / 51$$

$$SMI = 0.961$$

The test results showed that both SMI and approach 1.0 so ATLAS qualified software quality on component reliability.

3) *Efficiency testing results*

ATLAS is an android software composed of a web application that is ported in the form of android so testing efficiency using the principle of testing web application. Testing the effectiveness of sub- component time behavior

(performance) a particular software especially software can be done using a web -based web application performance testing as provided by Web Page Testing with address <http://www.webpagetest.org/>. From the web page load time can be seen for the first time or the next so that the results can be compared according to the standard.

Web Page Testing provides descriptions of the test results in tabular form which can be seen in the Table VI below.

TABLE VI: PERFORMANCE TESTING RESULTS

Load Time	First Byte	Start Render	Result (error code)	Document Complete			Fully Loaded		
				Time	Requests	Bytes In	Time	Requests	Bytes In
4.972s	0.731s	1.265s	0	4.972s	28	619 KB	5.431s	29	619 KB

From the Table VI, it is known that the time required load time is 4.972 seconds. Based on these results it can be seen that the ATLAS quite well and still get a percentage of users to more than 80 % by Subraya’s research (2006) [3]. Meanwhile, when compared with the Fortune research, ATLAS software has performed above average performance application website in the world.

4) Testing results maintainability

To determine the quality of software in terms of the need to know maintainability index(MI) of this software.To calculate the maintainability index of ATLAS can be done by using the formula presented by Ganpati (2012) as follows [4].

$$MI=171-5.2 \times \ln(aveV)-0.23 \times aveV(g) -16.2 \times \ln(av eLOC) \tag{3}$$

From the formula it can be seen MI a software but first need to Calculate Cyclomatic Complexity, LOC and Hastead Volume of ATLAS.

a) Calculating cyclomatic complexity

To calculate cyclomatic complexity should be used in the code snippet ATLAS Information Systems. In this piece of code that is used is the code contained in the server because the system is just a system android AJAX. The code used in this test is a function displays a list of existing space in the map. The function used in the menu Place in ATLAS.

```

1 <?php
2 header('Content-type: application/json'); // (1)
3 include 'config.php'; // (2)
4 if(isset($_GET['op']) && $_GET['op'] == 'all_ruang') { // (3) (4)
5     $data = array(); // (5)
6     $result = mysql_query("SELECT * FROM ruang JOIN fakultas USING(
7         idfakultas)"); // (6)
8     if (mysql_num_rows($result) > 0) { // (7) (8)
9         while ($dat = mysql_fetch_array($result)) { // (9)
10             array_push($data, $dat); // (10)
11         }
12     } else {
13         $data = array("status"=>"error"); // (11)
14     }
15     echo $_GET['callback'] . '(' . json_encode($data) . ')'; // (12)
16 }

```

Fig. 4. Pieces of program code module view place.

Of pieces of the program (Fig. 4) are then given a number to determine the order in which the process occurs. Of the number sequence can be formed a chart that contains nodes and edges. Nodes and edges describe the point process

describing the direction of the arrows. Here's a picture graph cyclomatic complexity Place ATLAS menu code snippets.

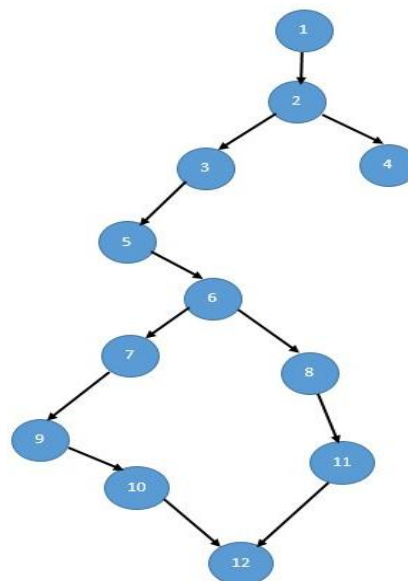


Fig. 5. Graph cyclomatic complexity.

From the Fig. 5, it can be seen the number of Edge arrows) as well as the number of nodes (points). The following is the number of nodes and the number of edges of the pieces of the program.

$$E = 12$$

$$N = 12$$

Unknown Edge Node numbered 12 and totaled 12 so that by using the proposed equation Ganpati (2012) to know the value of cyclomatic complexity. Here's a cyclomatic complexity calculation of ATLAS.

$$V(g)=E-N+2 \tag{4}$$

$$V(g)=12-12+2$$

$$V(g)=2$$

From these calculations it was found that the value of cyclomatic complexity is 2. This value will be used to find the value maintainability index (MI).

b) Looking for value hastead volume

Hastead volumes can be searched by knowing the value of N is the number of operators in program cuts and n is the number of differences in discount program operator (Gorman, 2006) [5]. To determine the number of operators and the amount of the difference operators can be seen in the code snippet on the menu Place ATLAS. From the code snippet below is known as follows.

$$N = 3$$

$$n = 3$$

So with unknown $N = 6$ and $n = 3$ can be discovered by the use of value Hastead volume following equation.

$$V = 0 \times \log(n) \tag{5}$$

$$V = 3 \times \log(3)$$

$$V = 1, 43$$

Of the equation is known that the value Hastead Volume ATLAS is 1.43. By knowing the volume hasted cyclomatic complexity and can be used to calculate the index maintainability. While to know the line of code can be used software program Simple Metric Code on Netbeans. Best knowledge of the software that the average LOC per module is 148.

c) Calculate maintainability index (MI)

Once known cyclomatic complexity, LOC hasted volume and value maintainability Index can be calculated by the following equations.

$$NI = 171 - 5.2 \times \ln(aveV) - 0.23 \times aveV(g) - 16.2 \times \ln(aveMPC)$$

$$NI = 171 - 5.2 \times \ln(1.43) - 0.23 \times 2 - 16.2 \times \ln(148)$$

$$NI = 87.725 \quad (6)$$

From these results it was found that maintainability index of ATLAS software worth 87.725. This means that if the research is based on Ganpati (2012), ATLAS software has high maintainability (Highly maintainable) because maintainability Index of ATLAS value is in the range of 86-100. The conclusion is the ATLAS software meets quality maintainability software components.

V. CONCLUSION

Based on the research and development that has been done, it can be concluded that the ATLAS software has met the four standards of software quality based on the characteristics of the ISO 9126 functionality, reliability, efficiency and maintainability.

A. Characteristics Functionality

Atlas software tested with characteristics sub security aspects namely functionality. Based on the aspects of security, software tested with SQL injection and cross - site scripting. the test results have done, atlas software cannot be injected in SQL injection or cross - site scripting aspects that meets functionality.

B. Characteristics Reliability

Based reliability testing with characteristics maturity index calculation software that get results obtained maturity index values of 0961 that the software contained approach to value index 1.0 maturity software which means that meets standard atlas reliability.

C. Characteristics Efficiency

Component based testing the effectiveness of time behavior (performance) testing using web page load time unknown personal information systems development is 4.972 sec. if compared to meet the load time either category namely software atlas 10 seconds then meets standard efficiency.

D. Characteristics Maintainability

Based on the characteristics of software testing using maintainability then index values obtained atlas software that have value index of maintainability 87.725. if compared to research ganpati (2012) [4], information systems software

development personal care facilities that have high (highly maintainable). ATLAS software that meets standards maintainability.

Finally, The conclusion is information systems software development meets self ISO 9126 standard test with the four characteristics of the namely functionality, reliability, efficiency and maintainability IEEE.

ACKNOWLEDGEMENTS

Authors would like to Yogyakarta State University (YSU) Indonesia for sponsoring this Works.

REFERENCES

- [1] S. P. Roger, *Software Engineering: A Practitioner's Approach 5th Ed.* New York: McGraw-Hill Companies, Inc., 2001.
- [2] Y. W. Huang, S. K. Huang, and T. P. Lin, *Web Application Security Assessment by Fault Injection and Behavior Monitoring*, Association for Computing Machinery (ACM), pp. 148-159, 2003.
- [3] B. M. Subraya, *Integrated Approach to Web Performance Testing: A Practitioner's Guide*, London: IRM Press, 2006.
- [4] G. Anita, A. Kalia, and H. Singh, "Maintainability index over multiple releases: A case study PHP open source software," *International Journal of Engineering Research & Technology*, vol. 1, issue 6, pp. 1-3, 2012.
- [5] G. Jason. (2006). Metric definition. [Online]. Available: <http://www.codemanship.co.uk/parlezuml/metrics/Metrics%20Definitions.pdf>



Pradana Setialana was born on May 6, 1992, Bantul. He is a 7th semester student in Informatics Engineering Education Study Program, Department of Electronics Engineering Education, Faculty of Engineering at Yogyakarta State University, Indonesia. His research interests include web engineering and mobile application development.



Muhamad Rizki Fajri was born on August 19, 1992, in Cilacap. He is a 7th semester student in Informatics Engineering Education Study Program, Department of Electronics Engineering Education, Faculty of Engineering at Yogyakarta State University, Indonesia. His research interests include education media development and mobile application development.



Abdul Rachman Pambudi was born on December 20, 1992, in Kulon Progo. He is a 7th semester student in Informatics Engineering Education Study Program, Department of Electronics Engineering Education, Faculty of Engineering at Yogyakarta State University, Indonesia. His research interests include web engineering and mobile application development.



Ratna Wardani was born on December 18, 1970, in Padang. She is a lecturer in object-oriented systems and software engineering at Department of Electronics Engineering Education, Faculty of Engineering at Yogyakarta State University. Her research interests include QoS for low quality connection.

She received her PhD. in computer and information systems and a M.T. degree from Gajah Mada University, Indonesia.