

# Analysis and Design Web Portal Amazing North Sulawesi using AUP Methodology

Stanley Karouw, ST., MTI

Faculty of Engineering, Sam Ratulangi University, Indonesia  
stanley.karouw@unsrat.ac.id

## ABSTRACT

*Exotic tourism promotion for investment is one of the main business processes carried out by the Office of Governor of North Sulawesi Province. Using Web 2.0 technologies is one of effective strategy to promote exotic tourism of the province of North Sulawesi. Agile Unified Process (AUP) methodology is one of recent method for developing information systems/applications which are concise and comprehensive, emphasize the role of the user (user-oriented) with object-oriented development paradigm. Web Portal developed by AUP methodology would produce web-based applications that meet user expectations and needs in relatively short time.*

**Keywords— Software Development, Web 2.0, AUP, UML**

## I. INTRODUCTION

### A. Background

The using of internet by the government institution and society rapidly grow. Internet user grows in number and services. North Sulawesi Local Government, especially the office of Investment Expert Staff seeing this phenomena as opportunity to promote exotic tourism. Using Web 2.0 technology to create competitive advantage as strategic and crucial considerations to foster investment climate for North Sulawesi tourism. Utilization of Web based 2.0 technology is expected to improve service, market share and shaping public opinion by providing targeted and comprehensive information on North Sulawesi exotic tourism, which is easily obtained by anyone, anywhere, anytime, using any device.

### B. Problems

Exotic tourism places owned by North Sulawesi Province is threatened by the onslaught of tourism promotions of some new entrants such as Wakatobi and Raja Ampat. While the number of foreign tourist relatively declined over the 5 years before, the tourism potential of Bunaken National Park is not fully optimized by stakeholders. Lack of comprehensive information about Bunaken National Park is one of the reasons for the decline of foreign tourist visiting. Structuring a comprehensive information related to the potential of tourism places and tourism activities is one of a strategic solution that must be done to fix the pre-eminent tourism promotion. In addition, the information must be displayed in such as interactive user interface, constantly available for 24/7 viewing and able to be found easily. Web Portal Amazing North Sulawesi is expected to be effective solution.

### C. Research Question

The research questions for this research are 1) How to design architecture application using web-based 2.0 platform?; 2) How to using AUP methodology for build web-based application?

## II. THEORITICAL CONSIDERATIONS

### A. Software Lifecycle Models

Software lifecycle models, presented by Schach<sup>[1]</sup> is an ideal software development phases. This model considers the software as a product produced in a specific sequence steps. The specific sequence steps are: 1) Starting from scratch (i.e starting form nothing); 2) Requirements (or needs) definitions; 3) Analysis Phase; 4) Design Stage; 5) Implementation Phase. Sommerville<sup>[2]</sup> presented four fundamental software lifecycle models stage, which are; 1) Software specification; 2) Software design and implementation; 3) Software validation; 4) Software evolution. Pressman<sup>[3]</sup> proposed a software generic process framework such as: 1) Communication; 2) Planning; 3) Modelling; 4) Construction; 5) Implementation. Dennis, Wixom and Tegarden<sup>[4]</sup> presented information system process models, called System Development Life Cycle (SDLC) constructed by: 1) Planning, 2) Analisis, 3) Design, 4) Implementation. That stages is similar which Bentley and Whitten<sup>[5]</sup> proposed, i.e: 1) System Initiation; 2) System Analysis; 3) System Design and 4) System Implementation. While Kendall and Kendall<sup>[6]</sup> proposes 7 (seven) steps for SDLC, i.e: 1) Identification of issues, opportunities and objectives; 2) Determination of user information requirements; 3) Sistem requirements analysis; 4) designing a system that have been recommended; 5) system development and documentation; 6) System Testing; 7) System Impelementation and Evaluation.

Software Engineering Institute – Carnegie Mellon (SEI)<sup>[7]</sup> proposed a framework which called CMMI for Development (CMMI – DEV). CMMI® (Capability Maturity Model® Integration) framework is a collection of best practices that help organization to develop software process. This model is developed among industry, government and academia in the SEI. CMMI-DEV, provides a comprehensive set of guidelines related to the development of services and software products. According to Schach<sup>[1]</sup>, software lifecycle model contrasted ideally to practical way because of two reasons: 1) software practitioners are humans, tends to make mistakes, 2) users needs and expectation tends to change when the software being developed.

### B. Unified Modelling Language (UML)

UML stand for Unified Modeling Language, a modeling notations for software applications. Schach<sup>[1]</sup> confirms that the UML is a language rather than a method. As a language, UML is used to describe software that is developed with a variety of software development paradigm and methodology. Schach opinion<sup>[1]</sup> is supported by Sommerville<sup>[2]</sup> and Pressman<sup>[3]</sup>.

Dennis, Wixom and Tegarden<sup>[4]</sup> support the notion that the UML is a set of modeling standard, using diagrams. UML aims to provide the vocabulary of object-oriented system development paradigm in order to model all phases of software development life cycle. Bentley and Whitten<sup>[5]</sup>, supports the notion that UML is a collection of commonly agreed modeling tool for explaining the software system. A similar proposed by Kendall and Kendall<sup>[6]</sup>.

Fowler<sup>[8]</sup> provide a simple definition that UML is a collection of graphical notations, backed by single meta-model, which help descriptions and design of software systems, especially systems that are built using object-oriented programming. UML is an open standard governed by the Object Management Group (OMG), an open consortium. OMG serves to create standards that support interoperability of object-oriented systems. The latest version of UML is the UML ver 2.0<sup>[9]</sup>.

According to Kruchten<sup>[10]</sup>, UML is a graphical language for visualizing, specifying, constructing and documenting any artifacts of software systems. UML supports the 4+1 View Model of Architecture, i.e 1) The Logical View, 2) The Implementation View, 3) The Process View and 4) The Deployment View plus 5) The Use Case View. Model is a complete representation of a software system, while the architecture is focus of views on certain parts of the system software. Or it can be said that the system architecture is the application blueprint. Software model and software system architecture connectedness, illustrated by the UML.

### C. Agile Unified Process (AUP) Methodology

Unified Process (UP) is one of methodology for software system development using object-oriented paradigm. UP is a combination of several methodologies which developed by Graddy Booch, James Rumbaugh and Ivar Jacobson<sup>[4]</sup>. UP methodology is component-based development; this means that the working software will consist of software components that are connected through well-defined interface. Characteristics of the UP methodology is intended to consistently adapt to the trend of applications/information systems development, that increasingly large and complex.

Agile approach began to emerge in software development in the era of the 2000s. Principles of agile approaches can be found at Agile Alliance<sup>[11]</sup>. Aoyama<sup>[12]</sup> explains that the agile approach in informatics engineering disciplines is necessary to reduce "slowness" at software development stages, in order to anticipate the needs of demanding business needs which expected to produces software with relatively short time-to-market. Jacobson<sup>[13]</sup> describes agile as a team that is ready to respond to change, the changing needs of users. In this sense means that each development team must closely cooperate with the user in developing web based applications. Response to anticipate the changes are the main characteristics of the agile approach.

## III. METODOLOGY

The AUP stages of problem solving methodology, followed steps which proposed by Center of Computer Science (Pusilkom) - University of Indonesia<sup>[14]</sup>. This Agile UP guide refers to the methodology created by Ambysoft Inc.<sup>[15]</sup> (see Fig. 1 and Fig. 2).

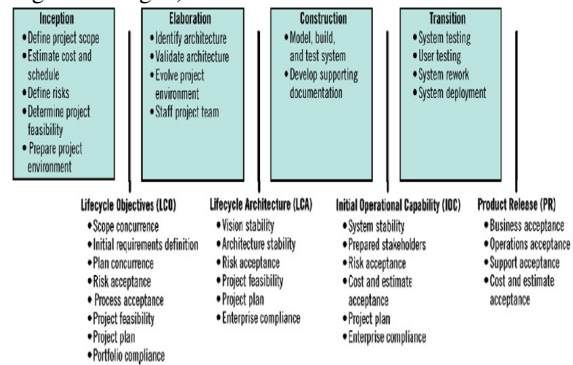


Fig. 1 AUP Phases

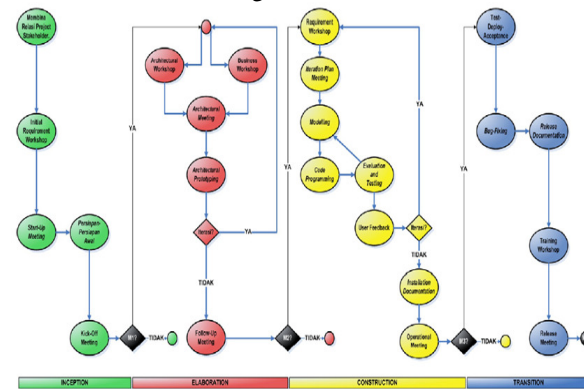


Fig 2. AUP Process Activities

The phases of analysis and design are as follows:

- 1) *Inception*, with the activity of defining project scope, cost estimating and scheduling, define risks, making the feasibility of the project and prepare a project execution environment (team work, installation, and so on). Iteration process is done once. Generated artifacts include Vision, Supplementary Specification, dokumen Glossary, Gantt Chart and Iteration Plan.
- 2) *Elaboration*, with the activity of identifying and validating the application architecture. Iteration process can be done one to two times. The resulting artifacts are UML Use Case, Architecture Model (update and snapshot), Architecture Prototype Code, Scenario Test Plan, Business Rule Document, Updated Supplementary and Glossary Document.
- 3) *Construction*, with modeling activities, build and test system applications (unit testing) as well as supporting documentation. Iteration process can be done two to eight times. The resulting artifacts are Updated Use Case, Updated Supplementary and Glossary, Domain Model (snapshot), UML Activity Diagram (snapshot), UML Class Diagram (snapshot), CRC Card, UML Sequence Diagram (snapshot), Source Code, Code Documentation, Regression Test Suite, Acceptance Test dan Bugs Report.

4) *Transition*, with activity testing the system (system integration and user testing), review the application and the system, and installation for the working application system. Iteration process can be done one to two times. The resulting artifacts are System Requirement Specification Document, System Technical Specification Document, User Installation Manual and User Manual, Training Document, Regression Test Suite, User Acceptance Test and Bugs Report (final update).

AUP guide from Pusikom UI also provides best practice in conducting any activity in each phase. These guidelines also distinguish artifacts generated document, as the main artifact, artifacts, and supporting input artifacts and output artifacts. This guide also provides LCO (Lifecycle Objective) in the form of documents and presentations of each phase, as targets to be achieved before proceeding to the next phase. For the purposes of writing this paper, the authors will limit the artifacts to be displayed.

#### IV. ANALYSIS AND DESIGN

##### A. Inception Phase

One of the main artifacts produced in this phase is the Vision document. This document describes the organization/project management and project scope definition. Project management, especially related to the estimation of the major applications to be developed, number of developer required, included time and costs required. Defining project scope mainly involves the identification and classification of user requirements.

##### A.1 Project Management

An important part of which is related to project management is estimate software size, the number of developers required, working time and costs required. Software size estimation, the number of developers required and working time are counted using Function Point Analysis. While the project cost is calculated by using the ROI and NPV.

##### A.1.1 Function Point Analysis

*Lines of Code (LOC):*  $TAFP * 44 = 152.29 * 44 = 6700$

*Estimate Effort Required:*

$(1.4 * LOC) / 1000 = (1.4 * 6700) = 9381 / 1000 LOC = 9.3 \text{ person-month}$

*Estimate Time Required:*

$(3.0 * \text{person-months}^{1/3}) = 3.0 * 9.3^{1/3} = 9.1 \approx 9 \text{ months}$

##### A.1.2 ROI and NVP

*Net Present Value (NPV Cumulative):* Rp 5.901.567

*Return on Investment (ROI):* 52.24%

*Break Even Point (BEP):* 2.34 bulan

##### A.2 Project Scope Definition

User requirements are classified into functional requirements and non-functional requirements. The focus for application development lies at functional requirements lists. These functional requirements will be modeled through the UML Use Case Diagram and UML Use Case Description. Interviews and Questioning are technique for gathering the functional requirements. Functional requirements are collected from stakeholders such as project champion, users and developer team.

##### A.2.1. Functional Requirements

List of functional requirements are:

##### 1). *Viewing Info*

- 1.1 The system can display information about the ads, profile, headline news and general articles.
- 1.2 The system can display links.
- 1.3 The system can display visitor counter for each pages.
- 1.4 The system can display currency value and weather report.

##### 2). *Managing Info*

- 2.1 Input ads, profile, news, headline and articles.
- 2.2 Edit ads, profile, news, headline and articles.
- 2.3 Delete ads, profile, news, headline and articles
- 2.4 Save ads, profile, news, headline and articles.

##### 3). *Collaborating*

- 3.1 The system must provide facilities for posting and reply comments for news, headline and articles.
- 3.2 The system must provide sharing features for social media, i.e: Facebook, Twitter and G+.
- 3.3 The system must provide polling feature.

##### A.2.2. Non-Functional Requirements

Non-functional requirements are distinguished in terms of operational, performance and security. Some non-functional terms to be met by the system are as follows:

Operational Requirements: 1) The system must be displayed in Indonesian and English; 2) The system can be operated on a smartphone, desktop and notebook on the optimal display resolution; 3) The system must be able to work on all web browsers; 4) The System must running through the operating system Windows and Linux. Performance Requirements: 1) The system must be used or operated within 24 hours a day, 7 days a week and 356 days a year; 2) Each user interaction with the system should not be longer than 3 seconds. Requirements for security are: 1) The system must provide privilege access for groups of admins and users; 2) The system must provide verification procedure for posting comments.

##### B. Elaboration Phase

##### B.1 Use Case Model

Read Fig 3 below for UML Use Case Model.

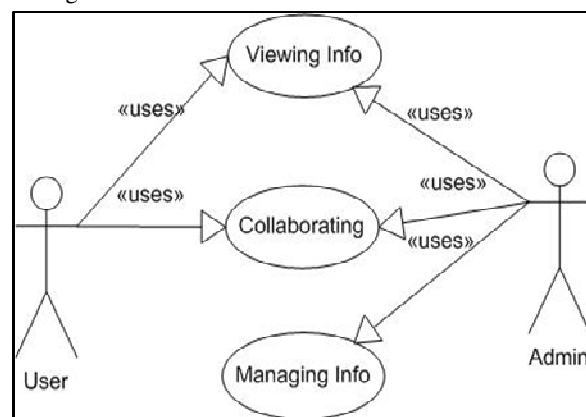


Fig. 3 UML Use Case Diagram

One of the *UML Use Case Description* can be read at Fig. 4.

<b>Nama Use Case:</b>	Managing Infor: Edit Content	
<b>Aktor:</b>	Admin dan sistem	
<b>Deskripsi:</b>	Admin mengubah record dari suatu tabel.	
<b>Normal Course:</b>	<b>Admin</b>	<b>Sistem</b>
	<ol style="list-style-type: none"> <li>Memilih record yang ingin diubah.</li> <li>Mengubah content yang ada dan submit.</li> </ol>	<ol style="list-style-type: none"> <li>Menampilkan form untuk mengubah content.</li> <li>Memvalidasi kelengkapan data yang dimasukkan.</li> <li>5a. Data lengkap, menyimpan data ke dalam basis data</li> <li>6a. Menampilkan pesan data telah diubah.</li> </ol>
<b>Alternate Course:</b>	<b>Admin</b>	<b>Sistem</b>
		<ol style="list-style-type: none"> <li>5b. Data tidak lengkap, tidak menyimpan data kedalam basis data.</li> <li>6b. Menampilkan pesan data yang dimasukkan tidak lengkap.</li> </ol>
<b>Pre-Condition:</b>	Login	
<b>Post-Condition:</b>	Logout	
<b>Assumption:</b>	-	

Fig. 4 UML Use Case Description Edit Content

## B.2 Architecture Application Model

Architecture Application Model defined by object model/data structure model, application behaviour model and interface design layer.

### B.2.1 Object Model/Data Structure and Management

The static structure model of system application describes all the classes and relationships between classes in the system application. The design of data management including data access logic manipulation layer and its actual design of the storage. In this application is used RDBMS (relational database management system). Using ERD Diagram as Fig. 5 below.

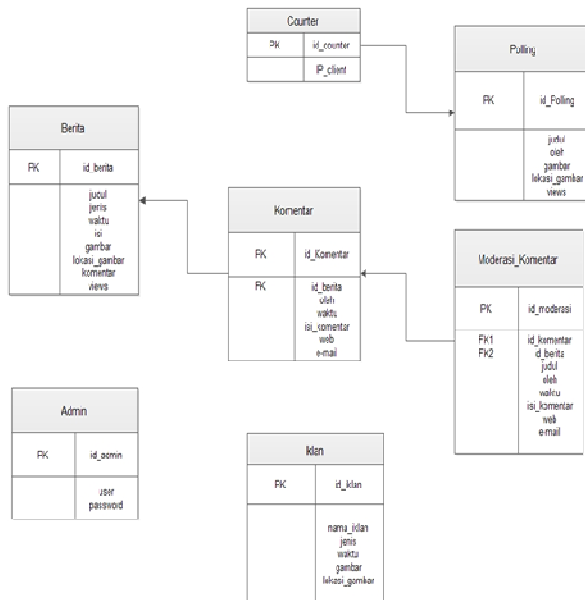


Fig. 5 ERD Diagram Model

## B.2.2 Behaviour Model

Behavior system application model is intended to provide an overview of the behavior of existing objects in the application. For this paper, we just presented one of the functions systems, called Edit Data. We have used UML Sequence Diagram to explain behaviour system application. (Fig. 6)

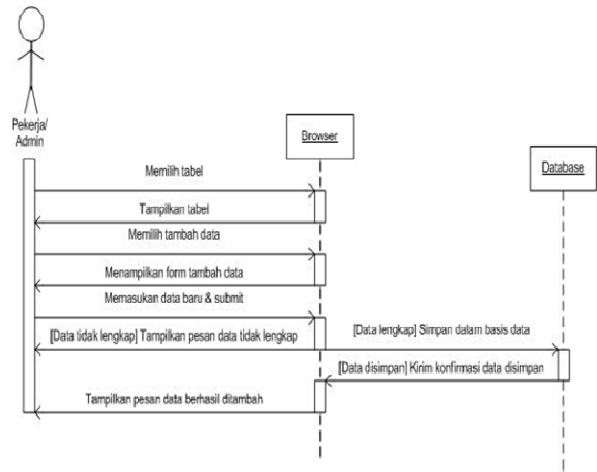


Fig. 6 UML Sequence Diagram Edit Data

## B.2.3 Interface Design Layer

Interface design is the process of defining how the system interacts with an external unit. The user interface consists of 3 (three) basic parts. The first is the navigation mechanism, a way of giving instructions to the user and the system tells the system what to do, such as buttons and menus. The second is the input mechanism, a way of capturing information system (e.g a form to add news). The third is the output mechanism of how the system provides information to users or to other systems (e.g reports, web pages). (Fig. 7, 8 and 9).

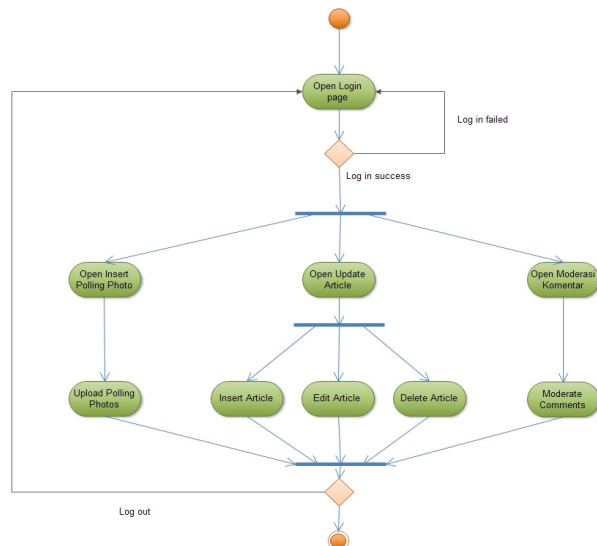


Fig. 7 UML Navigation Diagram for Actor Admin

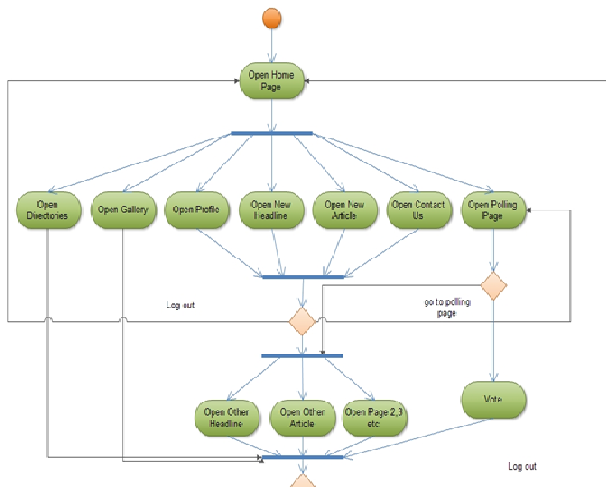


Fig. 8 UML Navigation Diagram for Actor User

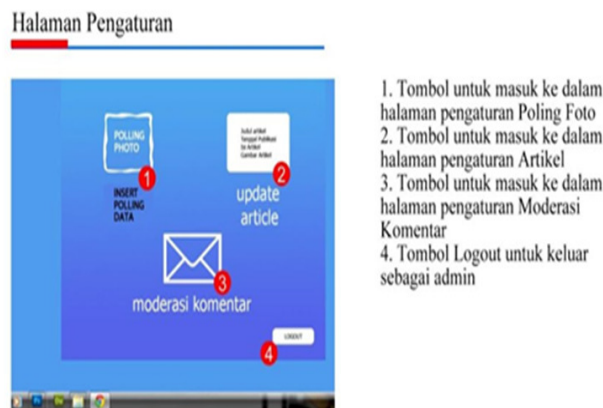


Fig. 9 Admin Page Storyboard

### C. Construction Phase

#### C.1 Coding

Codification process using xHTML, CSS and JavaScript. The coding applications snapshot can be seen in Fig. 10 below

```

<!--
<tr>
 Contact Us: amazingnorth@outlook.com | | |
```

Fig. 10 Coding Snapshot

### C.2 Testing Procedure

#### C.2.1 System Testing Purpose

The purpose of these tests:

- 1). To check whether the user requirements are met.
- 2). To ensure the error does occur in the system and fixes.

#### C.2.2 System Testing Criteria

Criteria for testing the system in general that the application can be run properly and error occurred to be fixes. Some criteria to benchmark the success of this system are as follows:

- 1) Applications are developed in accordance with the results of the analysis and design are done.
- 2) The function or procedure being implemented properly.

#### C.2.3 Testing Case

The test cases can be seen in Fig. 11 and 12 below.

No	Titik Pemeriksaan
1.	<b>Testing login Admin</b> Apakah <i>username</i> dan <i>password</i> yang dimasukkan dapat dibaca oleh sistem? Apakah sistem dapat mengidentifikasi admin?
2.	<b>Testing fungsi menambah berita</b> Apakah validasi kelengkapan data berjalan dengan baik? Apakah sistem akan menambah data dalam basis data jika berhasil mengisi detail data?
3.	<b>Testing fungsi mengubah iklan</b> Apakah validasi kelengkapan data berjalan dengan baik? Apakah sistem akan mengubah detail data dalam basis data jika berhasil mengisi detail data baru?
4.	<b>Testing fungsi menghapus berita</b> Apakah sistem dapat menghapus data yang diinginkan?
5.	<b>Testing logout Admin</b> Apakah sistem akan membawa pengguna kembali ke halaman login? Apakah sistem dapat menghapus <i>session id</i> ketika melakukan <i>logout</i> ?
6.	<b>Testing menampilkan content</b> Apakah sistem dapat menampilkan data sesuai dengan tipe pengguna yang melakukan <i>log in</i> ? Apakah data yang ditampilkan sesuai dengan data yang tersimpan dalam basis data?
7.	<b>Testing paging</b> Apakah sistem dapat menampilkan pengaturan halaman setiap halamannya?

Fig. 11 Test Case

No.	Kegiatan	Hasil yang diharapkan	Hasil yang didapat
1.	Memasukkan data <i>log in</i> : Username: admin Password 1: adminans Password 2: admin1	Sistem akan menampilkan halaman admin.	Sistem menampilkan halaman admin.
2.	Memasukkan data <i>log in</i> : Username: admin Password 1: adminans Password 2: admin2	Sistem akan menampilkan halaman admin.	Sistem menampilkan halaman admin.
3.	Memasukkan data <i>log in</i> : Username: admin Password 1: admin1 Password 2: admin2	Sistem akan menampilkan pesan kesalahan.	Sistem menampilkan pesan kesalahan.
4.	Memasukkan data <i>log in</i> : Username: admin Password 1: adminans Password 2: adminans	Sistem akan menampilkan pesan kesalahan.	Sistem menampilkan pesan kesalahan.
5.	Memasukkan data <i>log in</i> : Username: admin Password 1: [null] Password 2: [null]	Sistem akan menampilkan pesan kesalahan.	Sistem menampilkan pesan kesalahan.

Fig. 12 Test Case Results

### D. Transition Phase

This paper focused on analyzing and designing process, however, planning for implementation phase also important. Physical design architecture explains the implementation plan for this application. We proposed client-server achitecture to maintain a balance between client and server processes which

have the function of each application. Client is responsible for the presentation logic while the server is responsible for application logic, the data access logic (RDBMS) and data storage. Physical Design Architecture can be seen in Fig. 13 below:

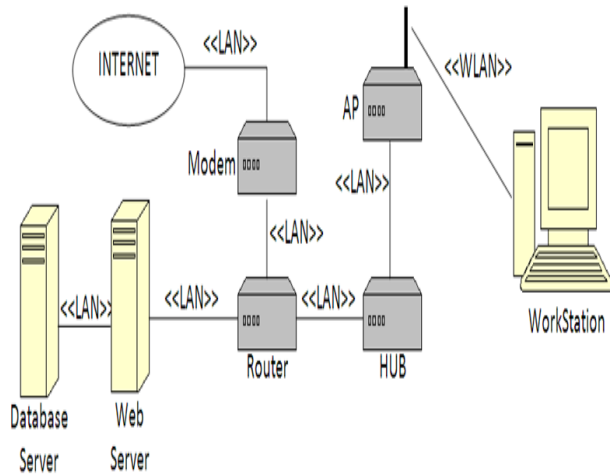


Fig. 13 Physical Architecture

Other documentation for implementation phase are the User and Installation manual. This documents will be presented to stakeholders prior to user training.

## V. CONCLUSIONS

Some conclusions from the writing of this paper include:

- 1) Agile Unified Process Methodology, which is abbreviated AUP, can be used to build Web-based applications with object-oriented approach. Main characteristic of this AUP methodology is stakeholder active involvement.
- 2) UML version 2.0 can be used as tools for making software model in detail and simple, so it is very useful for developers and users. When used in the AUP methodology, precision is required in selecting of UML diagrams appropriate for developers and easily understood by the user; to the effectiveness of each process related with analysis and design process. In practical terms, UML can be represented in the form of sketches.
- 3) Development of Web-based applications using the AUP methodology have higher relative risk of scope creep. It takes a careful risk control process by the developer and stakeholders.
- 4) North Sulawesi Amazing Web Portal can be a reasonable solution as a web-based media 2.0 campaign to foster North Sulawesi Province tourism investment.

## REFERENCES

- [1] Schach., *Object Oriented Software Engineering*, 8<sup>th</sup> Ed, McGrawHill, 2008.
- [2] Sommerville., *Software Engineering*, 8<sup>th</sup> ed, Pearson Education Limited, 2007
- [3] Pressman, *Software Engineering, A Practitioner's Approach*, 6<sup>th</sup> ed, McGrawHill, Singapura, 2005.

- [4] Dennis, Wixom and Tegarden, *System Analysis and Design with UML, An Object-Oriented Approach*, 3<sup>rd</sup> ed, John Wiley & Sons, International Student Edition, 2010.
- [5] Bentley and Whitten, *System Analysis and Design for the Global Enterprise*, 7<sup>th</sup> ed, Mc GrawHill International Edition, 2007.
- [6] Kendall and Kendall, *System Analysis and Design*, 7<sup>th</sup> ed, Pearson Prentice Hall, 2007.
- [7] CMMI Product Team, *CMMI® for Development, Version 1.3, Improving processes for developing better products and services*, November 2010, TECHNICAL REPORT CMU/SEI-2010-TR-033 , ESC-TR-2010-033, Software Engineering Process Management Program, Unlimited distribution subject to the copyright. <http://www.sei.cmu.edu>.
- [8] Martin Fowler, *UML Distilled, A Brief Guide to the Standard Object Modelling Language*, 3<sup>rd</sup> ed, Pearson Education, 2004.
- [9] [www.uml.org](http://www.uml.org)., *Unified Modelling Language: Superstructure Version 2.0*, ptc/03-08-02.
- [10] Philippe Kruchten, *The Rational Unified Process An Introduction*, 3<sup>rd</sup> ed, Pearson Education, 2004.