DESIGN AND IMPLEMENTATION OF LABORATORY CONTENT MANAGEMENT SYSTEM (CMS) ‘iLAB’ USING CakePHP FRAMEWORK

Sunu Wibirama¹, Ir.Lukito Edi Nugroho, M.Sc., Ph.D.², Indriana Hidayah, S.T.³

ABSTRACT

The making of ‘iLab’ Content Management System (CMS) begins with a fact that there are only a few amount of web based open source CMS implemented for laboratory management. Also, laboratory management in Electrical Engineering UGM has not optimalized the advantage of information technology yet. Thus, the making of web based laboratory CMS hopefully can increase the quality and effectivity of laboratory management in Electrical Engineering UGM.

iLab CMS was made with CakePHP framework. CakePHP is a framework based on PHP scripting language with Model-View-Controller (MVC) architecture and using Object Oriented Programming (OOP) concept for source code writing. The synthesis of MVC and OOP concept and also adequate framework documentation are the main reasons of using CakePHP as a framework to built a laboratory CMS application which is structured, easy to understand, and easy to develop.

Keyword: Content Management System (CMS), laboratory, CakePHP, iLab, Model-View-Controller (MVC).

¹ Author, Fresh Graduate Student of Electrical Engineering Department, UGM
² Supervisor I, Lecturer of Electrical Engineering Department, UGM
³ Supervisor II, Lecturer of Electrical Engineering Department, UGM
1. Introduction

Information technology as a branch of knowledge in contemporary computer engineering has given a lot of alternative solutions for management development and automatization of data transfer in various fields of work.

One of information technology implementation which is required by student, lecturer, laboratory staff, and institution staff of university is the use of Content Management System (CMS) for laboratory and practicum management.

The goal of the research is learning and understanding implementation of CakePHP framework for creating CMS, also developing it into an application which is used for laboratory management.

2. Basic Theory

2.1. Content Management System

Content Management System (CMS) is an application which is used for managing several methods related with web publishing. Based on Douglas [7], a CMS generally can be customized by adding or subtracting specific features, so that only several features needed which will be published to the user. Currently, CMS is widely used to create bulletin board, online trading website, community website, online photo gallery and so on.

Fraser [8] explained that Content Management System consists of at least three elements:

- Content Management Application (CMA). CMA will manage application content components, including picture, text, and so on.
- Metacontent Management Application (MMA). MMA will manage information carried by application content components.
- Content Delivery Application (CDA). CDA will pick content components, read the information within, and render the result for user consumption.

One of several various kind CMS is used and implemented for laboratory. It is called LIMS (Laboratory Information Management System). LIMS, based on Crandall dan Auping [6] can be explained briefly as a combination between software and computer hardware used in laboratory to manage laboratory sample, user, instrument, standard and other functionalities using database, report generator and computer network capabilities.

2.2. Framework CakePHP

Framework is a basic tool used for reducing overhead resulting from web development activities [13]. Framework generally has a class library for accessing database, templating, session management and several reusable program codes. CakePHP is an open source framework used for developing web application which has basic concept of CRUD (Create, Read, Update, Delete) [1].

![Picture 1. CakePHP Architecture [3]](image)

3. System Design

3.1. Requirement Analysis

Almost all of laboratories in Electrical Engineering Department, Gadjah Mada University had same problems related to practicum (lab work) and resource management. Problems and solutions were tried to be solved by design and implementation of laboratory Content Management System are shown in table 1.

<table>
<thead>
<tr>
<th>Problems</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Traditional practicum registration</td>
<td>1. Automatization of practicum registration</td>
</tr>
<tr>
<td>2. Traditional data</td>
<td>2. Automatization of</td>
</tr>
</tbody>
</table>
3. No resource repository for accessing modules and all of practicum administrations

4. No online information media for laboratory news

5. Restricting access and administering privileges for common user (not admin).

6. Automatization of installation process

7. Creating a user friendly system.

3.2. Interface design

CMS iLab consists of four interface design:
- User page
- Login page.
- Administration page.
- Administration menu page.

3.3. Database design

Creating association between database entities was the first step of CMS iLab database design. There are 15 entities associating 15 CMS iLab functionalities. Several of them allocated for practicum management are Practicumnames, Practicumschedules, Practicians, and Assistants. Others allocated for administration page, news board and repository are Userstatuses, Users, Projects, Resources, Newscategories, News, Links, Guestbooks, Homes, Profiles, and Settings.

3.4. System Architecture Design

Picture 2 explains implementation of Model-View-Controller (MVC) architecture in CMS iLab. Model is a part of application which consists of class related to database and manages association between tables. Controller is a part of application which consists of class handling request from CMS user and being the business logic of application. View is a part of application directly interacting with user, rendering the result of data processed by Controller.

Picture 3 shown Use Case diagram of CMS iLab. There are 5 users in CMS iLab: Admin, Lecturer, Laboran, Member, and Guest. Guest is lab participant (practician) dan lab assistant. The others are different actor with different access privileges.

This Use Case diagram has 10 sub systems (modules). Default system, located on the left side, is accessed by all users, including Guest. Administration system, located on the right side, is a system accessed only by user having an account in database.

Picture 4 explains briefly the program flow of CMS iLab in MVC architecture. Model handles data sanitation and input validation from user. Sanitized and validated data then processed by Controller. The result of data processing is rendered by View in web interface made with pieces HTML, CSS, Javascript and PHP codes. User gets the result and all information processed by the system via this web interface using their web browser.
3.5. CMS iLab Components

The application is divided into several parts explained below.

CakePHP Framework

CakePHP Framework is the main part which is used to build CMS iLab. Within the application structure, class files are located in `cake/` folder. The framework’s version is 1.1.10.3825 (stable).

Main Library (Webroot)

Main library (webroot) consists of several files needed for supporting application interface. It consists of pictures, CSS (Cascading Style Sheet), Javascript, and some files for installation process and upload-download mechanism. The main library is located in `app/webroot/` folder.

Secondary Library

Other part of iLab is secondary library. It supports main library by providing extra functionalities for user. Some of them are:
- Kaptcha Library;
- Pagination Library;
- MySQL to XLS Converter Library.

Configuration Class

CMS iLab needs several configuration variable, related to installation type, database configuration, session configuration, production type configuration, and so on. CakePHP framework locates all of configuration classes inside `app/config/` folder. The most important configuration files are `core.php`, `routes.php` and `database.php`.

Main Module

CMS iLab has ten main modules designed to fulfill general requirement for each laboratory. Each module consists of three parts called Model, Controller, and View. There are Home Module, News Module, Profile Module, Practicum Module, Resource Module, Project and Research Module, Guestbook Module, Link Module, User Module, and Setting Module which are built as main module of CMS iLab.
Secondary Module

Beside main module, CMS iLab uses secondary module. There are Login Module and Installer Module. They use no database table because none of them have database entities.

4. System Implementation

4.1. System Evaluation

Evaluation Method

Evaluation process during implementation section is aimed to get information related to performance and durability from each module. Evaluation process will give us detail information of application bug and several errors occurred during implementation process. The evaluation process is divided into three steps:

1. Interface Evaluation.
   Interface evaluation has been done by running application with different internet browsers which are often used by user to access internet, such as Internet Explorer, Mozilla Firefox, Opera, and Safari.

2. System Installation Evaluation.
   System installation evaluation has been done by implementing CMS iLab on three different operating systems and running it through different network architectures, without and with proxy server.

   This evaluation shows the user response upon CMS iLab performance. The evaluation has been done by asking several testers to fill some quizzes related to CMS iLab. The testers are laboratory staff, laboratory administrator, laboratory assistant, laboratory participant (practitioner) and common user.

Interface Evaluation

CMS iLab generally could run without significant problem when it was accessed with different web browsers. Nevertheless, CMS iLab user interface had not accommodated several browsers yet. There were some bugs related to CSS and Javascript rendering depended on the browser.

CMS iLab ran perfect without any problem on Mozilla Firefox, although it needed more time to render the interface compared with other browsers. The use of Mozilla Firefox is recommended to guarantee all functionalities run without problem.

Picture 5. CMS iLab Interface

System Installation Evaluation

System installation evaluation had been done by implementing CMS iLab on three different operating systems, i.e. Microsoft Windows XP Service Pack II, Ubuntu Linux 6.0, and OpenBSD 3.9.

Installation implementation on three different operating systems with different characteristics needs specific procedures. Several requirements which are needed to run CMS iLab on all operating systems are:

- Activation of mod_rewrite module in Apache Web Server. By default, CakePHP uses mod_rewrite to access all directories within.
- GD Library (for image rendering) and XSLT (for spreadsheet rendering) must be activated in PHP installation.
- The use of .htaccess is allowed.
- app/config directory should be accessible by system, especially if the operating systems are Ubuntu Linux and OpenBSD. Therefore, its access privilege should be set in 755 or 777 with command chmod through command prompt or konsole. If the system uses Microsoft Windows XP, we don’t have to set the access privilege.
- Configuration of iBrowser plugin in TinyMCE library is manually set by adding absolute path of iLab installation in the configuration file.
User and System Interaction

The last evaluation showed that iLab user interface generally easy to access and use by user. Some bugs related to user interface will be fixed on the next iLab development.

5. Conclusion

CakePHP Framework can be used as a tool to develop an integrated application such as Content Management System which is involved a lot of modules. By understanding the functionality of CakePHP library class, CakePHP Framework can be developed into various web applications. The process of designing and implementing CMS iLab is successfully done. CMS iLab has ten modules which are used to accommodate general requirement of laboratory in Electrical Engineering Gadjah Mada University.

6. Future Work

CMS iLab is still possible to be improved into more complex application. Several things related to iLab which can be improved i.e :
- Advanced development of iLab interface and simplification of secondary navigation on iLab sidebar.
- Development of system logic in Model and Controller classes.
- Development of system module by creating new module or special mechanism for module installation based on web interface which support iLab user to easily add or subtract suitable module.
- Development of installation module so that user can easily install the system without any manually-done-by-user configuration on various operating system.
- Integration of CMS iLab with practicum (lab work) registration based on SMS (Short Message Service) by mobile phone.

Bibliography


