

## iLab: Sistem Manajemen Konten Laboratorium Terdistribusi

Dhomas Hatta Fudholi<sup>1</sup>, Sunu Wibirama<sup>2</sup>

<sup>1</sup> Jurusan Teknik Informatika, Universitas Islam Indonesia

<sup>2</sup> Jurusan Teknik Elektro, Fakultas Teknik, Universitas Gadjah Mada

e-mail: <sup>1</sup>hatta.fudholi@yahoo.co.id, <sup>2</sup>sunu@mail.te.ugm.ac.id

---

### Abstrak

Pembuatan Sistem Manajemen Konten Laboratorium (SMKL) iLab berawal dari fakta kurangnya jumlah aplikasi SMKL open source berbasis web yang diimplementasikan untuk manajemen laboratorium. SMKL iLab dikembangkan untuk memudahkan laboratorium dalam mengelola jadwal praktikum, berita, arsip penelitian dan dokumentasi yang ada pada laboratorium tersebut. Sebuah instansi yang memiliki lebih dari satu laboratorium dapat menggunakan SMKL iLab terdistribusi dalam pengelolaannya. SMKL iLab dikembangkan dengan framework CakePHP. Integrator berbasis web service dibuat untuk memudahkan pengguna dalam mencari informasi pada laboratorium-laboratorium dalam sebuah instansi. Hasil pengujian dan implementasi menunjukkan bahwa SMKL iLab layak untuk menjadi salah satu prototipe sistem informasi laboratorium terdistribusi.

Kata kunci: sistem informasi laboratorium, aplikasi web, sistem terdistribusi, teknologi informasi

---

### 1. Pendahuluan

Sistem Manajemen Konten (SMK) berbasis aplikasi web dikenal sebagai salah satu implementasi teknologi informasi lintas platform sistem operasi yang dikembangkan untuk pembuatan forum diskusi, forum komunitas, sistem *file sharing*, dan berbagai implementasi lainnya. Menurut Fraser [6], *Konten* bisa diartikan “sesuatu” yang terkandung dalam sebuah *website*. “Sesuatu” ini bisa diklasifikasikan kembali menjadi dua kategori:

- Informasi, seperti teks dan gambar, yang bisa dilihat saat sebuah situs diakses oleh pengunjung.
- Aplikasi atau perangkat lunak yang berjalan pada *server website* dan menampilkan informasi tersebut.

Menurut Douglas [5], sebuah SMK secara umum bisa dikustomasi dengan menambahkan atau mengurangi fitur yang spesifik, sehingga hanya fitur-fitur tertentu yang diinginkan saja yang akan ditampilkan kepada publik.

Dari berbagai jenis SMK tersebut, ada SMK yang diimplementasikan untuk laboratorium, dikenal dengan Sistem Manajemen Konten Laboratorium (SMKL). SMKL menurut Crandall dan Auping [4] bisa diartikan sebagai sebuah kombinasi antara perangkat lunak dan perangkat keras komputer yang digunakan di laboratorium untuk manajemen sampel, pengguna laboratorium (praktikan dan laboran), instrumen, standar, dan fungsi laboratorium lainnya dengan menggunakan *database*,

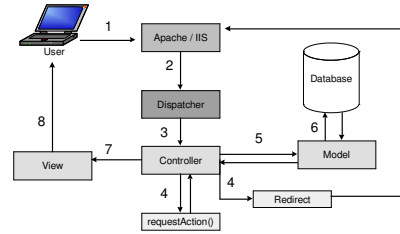
*report generator*, dan kapabilitas jaringan komputer.

Aplikasi SMKLab ini dikembangkan seiring dengan adanya kebutuhan untuk membuat pengolahan informasi pada laboratorium menjadi lebih sederhana dan memberi fitur otomatisasi pada proses-proses pengelolaan informasinya. [1] Pertimbangan instansi yang memiliki lebih dari satu laboratorium dan memerlukan suatu sistem informasi terpusat yang mampu menarik informasi dari masing-masing laboratorium, maka muncul sebuah konsep integrasi SMKLab dengan menggunakan *web services XML-RPC*.

## 2. Framework CakePHP

Siswoutomo [8] mendefinisikan *framework* adalah sebuah kerangka kerja yang penggunaannya bertujuan untuk mengurangi beban dari aktivitas-aktivitas yang sering dilakukan pada saat pelaksanaan proses pengembangan *web*. *Framework* biasanya menyediakan pustaka untuk akses *database*, *templating framework*, dan *session management* serta menawarkan kode-kode program yang dapat digunakan kembali (*reusable code*).

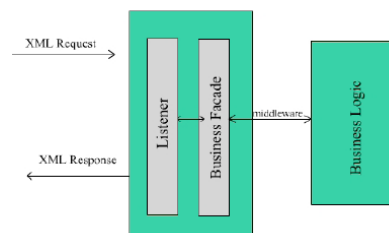
CakePHP adalah sebuah *framework open source* yang digunakan untuk mengembangkan aplikasi *web* dengan dasar kerja CRUD (*Create, Read, Update, Delete*) [3]. CakePHP berbasis bahasa pemrograman PHP dengan arsitektur *Model-View-Controller* (MVC) dan menerapkan konsep *Object Oriented Programming* (OOP) dalam penulisan kode sumbernya [2]. Arsitektur *framework* CakePHP dapat dilihat pada gambar 1.



Gambar 1. Arsitektur CakePHP

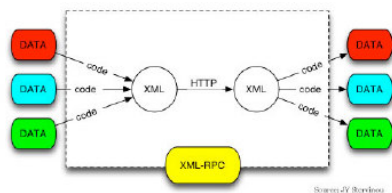
## 3. Web Service XML-RPC

W3C (*World Wide Web Consortium*) Web Services Architecture Working Group mendefinisikan *web service* sebagai sistem perangkat lunak yang didesain untuk mendukung interoperabilitas interaksi antar mesin pada sebuah jaringan. Richards [7] mendefinisikan *web service* sebagai aplikasi yang diakses melalui internet menggunakan protokol standar internet dan menggunakan XML sebagai format standar dalam penyampaian pesan. Salah satu bentuk arsitektur *web service* melibatkan XML sebagai media dalam melakukan *request* dan *response*. *Request* diterima oleh pembaca (*listener*), diartikan sehingga *business logic* mampu membaca dan mengolahnya, kemudian diolah sesuai dengan *request* yang diminta, lalu dikembalikan dalam bentuk yang dapat dimengerti oleh penerima (Gambar 2).



Gambar 2. Arsitektur Umum Web Service

XML-RPC kependekan dari *XML-based Remote Procedure Call*. *Procedure call*, sebuah istilah yang sebenarnya mengacu pada perwujudan dari sebuah *function call* sederhana pada bahasa pemrograman web PHP. Ini adalah kode yang bisa dipanggil, bertukar parameter yang digunakan oleh kumpulan kode tersebut, dan mendapatkan nilai hasil. Pertukaran data yang terjadi pada sistem yang berbeda diperlukan adanya sebuah penerjemah sehingga kedua mesin mampu mengerti informasi yang diterima, oleh karena itu diperlukan *marshaling*. *Marshaling* adalah proses pengambilan data dan merubahnya ke dalam format yang umum atau sudah terstandarkan (dalam hal ini adalah XML) supaya dapat dikirim, kemudian diterima oleh mesin penerima lalu diterjemahkan kembali sehingga mesin penerima mampu membacanya. Konsep XML-RPC terlihat pada gambar 3.



Gambar 3. Konsep XML-RPC

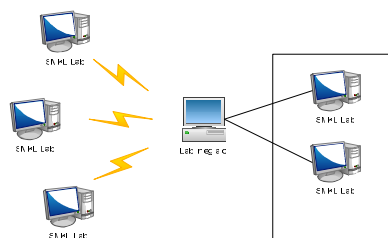
#### 4. Perancangan Sistem

Perancangan dan integrasi SMK iLab dilakukan dengan mempertimbangkan beberapa kebutuhan di bawah ini:

1. Perlunya sebuah sistem informasi yang mengotomatisasi pendaftaran praktikum, pengolahan data praktikum, penyajian petunjuk praktikum secara online dan pembatasan hak akses pengguna

(administrator, laboran, dosen dan mahasiswa).

2. Instansi pendidikan memiliki lebih dari satu laboratorium dan memerlukan lebih dari satu SMK iLab untuk tiap-tiap laboratorium.
3. Informasi lengkap yang terpusat dapat memudahkan pengguna untuk mencari lokasi praktikum, mencari bahan-bahan pendukung dan informasi terbaru seluruh laboratorium.



Gambar 4. Arsitektur Sistem iLab

Perancangan sistem untuk pengaplikasian iLab terlihat pada gambar 4. SMK iLab dapat di tempatkan pada jaringan lokal yang memiliki IP private atau jaringan internet yang memiliki IP public. Integrator berfungsi dalam memudahkan pencarian data pada ke semua SMK iLab yang terdaftar, dan tentunya memiliki basis data lokasi SMK iLab.

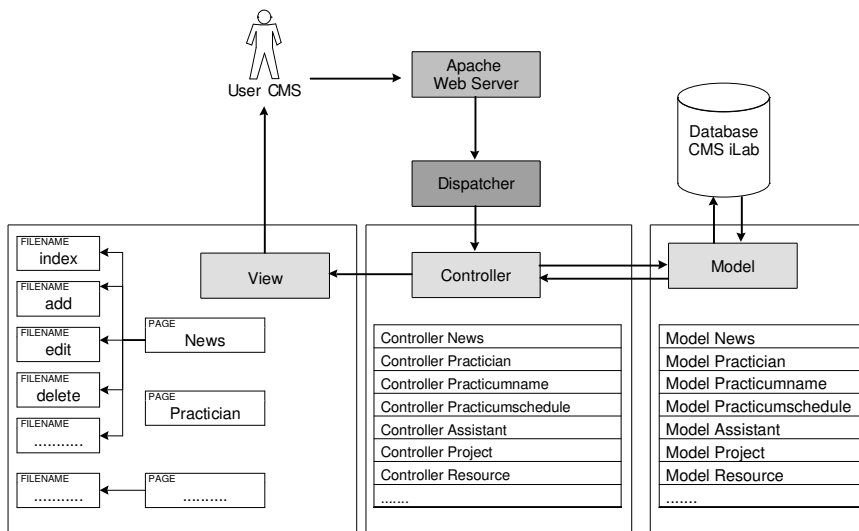
Langkah pertama perancangan basis data SMK iLab adalah menyusun hubungan antar entitas basis data yang ada. Ada 15 entitas yang mewakili fungsionalitas iLab. Dari 15 entitas yang ada, sebagian besar dialokasikan untuk manajemen praktikum, yakni *Practicumnames*, *Practicumschedules*, *Practicians*, dan *Assistants*. Sedangkan entitas lainnya adalah *Userstatuses*, *Users*, *Projects*,

*Resources, Newscategories, News, Links, Guestbooks, Homes, Profiles, dan Settings.*

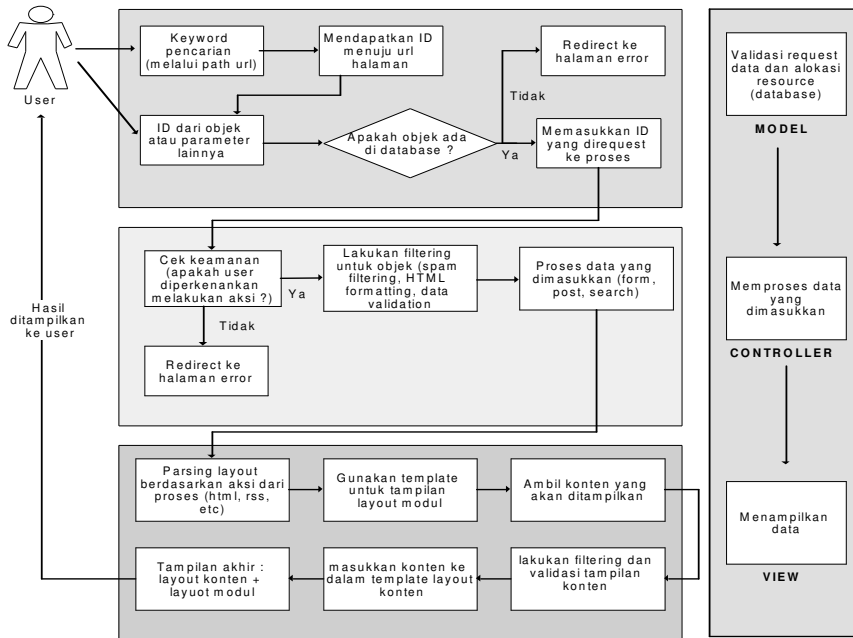
Gambar 5 di bawah menjelaskan penerapan arsitektur MVC pada SMKLab iLab. Bagian *Model* terdiri dari *class-class* yang berhubungan dengan basis data dan mengatur hubungan antar tabel. Bagian *Controller* terdiri dari *class-class* yang menangani *request* dari pengguna CMS dan mengolah data dari *database*. Sedangkan bagian *View* bertugas menampilkan data yang sudah diolah oleh bagian *Controller*. Alur dari SMKLab iLab kemudian terangkan pada gambar 6.

*Web service* pada integrator iLab menggunakan konsep *client-server*, dimana *client* adalah integrator iLab yang meminta atau melakukan *request* untuk mendapatkan informasi dari masing-masing SMKLab iLab.

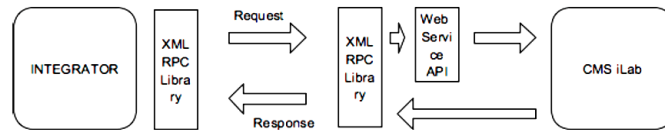
SMKLab iLab bertindak sebagai *server*. Proses *request* dan *response* terlihat seperti Gambar 7. Pada saat integrator melakukan *request* dan sebelum data dikirimkan, terjadi proses *marshalling* untuk membentuk data dengan format XML yang kemudian dikirim ke SMKLab iLab. *Request* yang diterima kemudian dibaca oleh *XML-RPC library*. *Request* tersebut akan dicek terlebih dahulu pada *web service API*. Pengecekan ini dimaksudkan untuk mengecek apakah fungsi yang diminta terdapat pada server atau tidak. Jika fungsi tersedia pada iLab maka pesan diteruskan untuk kemudian diolah oleh iLab. Hasil atau *response* dikembalikan kepada integrator, lalu integrator memunculkan informasi yang diterima sehingga pengguna dapat melihat informasi tersebut.



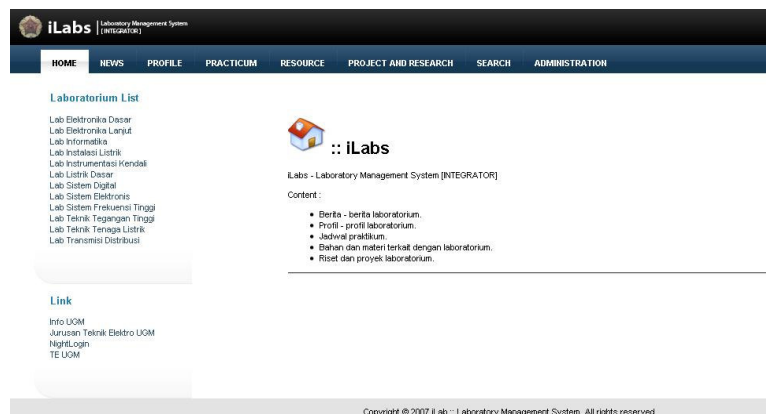
Gambar 5. Arsitektur MVC pada iLab



Gambar 6. Diagram Alir SMKL iLab



Gambar 7. Skema request dan response integrator



Gambar 8. Antarmuka SMKL iLab

## 5. Implementasi dan Pengujian

SMKL iLab diimplementasikan pada sebuah institusi pendidikan dengan dua belas laboratorium dan jaringan intranet yang mendukung kinerja manajemen data terdistribusi. Pengujian antarmuka SMKL iLab dilakukan dengan menggunakan *browser* Mozilla Firefox, Opera, Internet Explorer, dan Safari. SMKL iLab secara umum berjalan dengan baik pada saat diakses dengan *browser* yang berbeda. SMKL iLab berjalan sempurna pada *browser* Mozilla Firefox, meskipun memerlukan waktu *rendering* halaman lebih lama dibandingkan dengan *browser* lainnya. Penggunaan *browser* Mozilla Firefox direkomendasikan untuk menjamin semua fungsionalitas aplikasi berjalan dengan baik.

Pengujian instalasi sistem dilakukan dengan mengimplementasikan SMKL iLab pada tiga buah sistem operasi yang berbeda, yakni Microsoft Windows XP Service Pack II, Ubuntu Linux 6.0, dan OpenBSD 3.9.

Implementasi instalasi pada tiga buah sistem operasi dengan karakteristik yang berbeda menunjukkan perlunya penanganan yang berbeda pula. Beberapa hal yang menjadi syarat utama agar SMKL iLab bisa berjalan dengan baik pada semua sistem operasi yakni:

- Modul `mod_rewrite` harus aktif karena secara *default framework* CakePHP menggunakan `mod_rewrite` untuk mengakses seluruh direktori yang ada di dalamnya.
- Pustaka GD Library (untuk *rendering* gambar) dan XSLT (untuk *rendering file spreadsheet*) harus diaktifkan pada instalasi PHP.

- Penggunaan *file .htaccess* diperbolehkan.
- Direktori `app/config` harus bisa diakses oleh sistem berbasis unix (Ubuntu dan OpenBSD). Dengan demikian, harus diubah terlebih dahulu hak aksesnya menjadi 755 atau 777 dengan perintah *chmod* melalui *command prompt* atau *konsole*.
- Konfigurasi modul/plugins iBrowser pada pustaka TinyMCE dilakukan secara manual dengan menyesuaikan *path absolut* instalasi iLab di server tersebut.

Hasil pengujian interaksi *user* dan sistem memberikan gambaran bahwa antarmuka yang ditunjukkan oleh gambar 8 secara umum mudah digunakan (*user friendly*). Beberapa kelemahan dan kekurangan teknis akan disempurnakan sejalan dengan proses implementasi iLab di laboratorium yang memerlukan.

## 6. Kesimpulan

Perancangan dan Implementasi Sistem Manajemen Konten Laboratorium (SMKL) terdistribusi diperkenalkan pada *paper* ini. SMKL iLab memanfaatkan *framework open source* CakePHP dan *web service* XML-RPC untuk manajemen data terdistribusi. Pengujian dan implementasi dilakukan dengan menjalankan sistem dengan berbagai macam *web browser* yang berbeda, tiga buah sistem operasi, dan *survey* pengguna untuk mengetahui respon dan saran perbaikan sistem. SMKL iLab diimplementasikan pada dua belas laboratorium dengan jaringan intranet yang mendukung manajemen data *online* dan terdistribusi. Pada masa

mendatang, sistem ini dapat digunakan sebagai salah satu prototipe awal untuk sistem informasi laboratorium terdistribusi.

#### **Daftar Pustaka**

- [1] Alan S, Ted P, Christine P, 2007, *Laboratory Needs Assessment Benefits LIMS Implementation*, WaterWorld.
- [2] Anderson, J.; & Masters, L.E. (ed), 2006, *CakePHP Programmer's Reference Guide*. USA: CakePHP Software Foundation, Inc. 141 p.
- [3] Cevasco, Fabio., 2006, *An Overview with CakePHP Framework*, [Online] <http://hades.phparch.com/ceres/public/article/index.php/art::cakephp::overview>, Diakses pada tanggal 3 Januari 2007.
- [4] Crandall, Karen S.; & Auping, Judith V, 1987, *Laboratory Information Management System (LIMS)-A Case Study*, USA, National Aeronautics and Space Administration (NASA).
- [5] Douglas, Robert T.; Little, Mike; & Smith, Jared W., 2006, *Building Online Communities with Drupal, phpBB, and Wordpress*, USA, Apress.
- [6] Fraser, Stephen R.G. 2002. *Real World ASP.NET: Building a Content Management System*. USA: Apress. 405 p.
- [7] Richards, R., 2006, *Pro PHP XML and Web Services*, USA, Apress.
- [8] Siswoutomo, Wiwit, 2005, *PHP Enterprise: Kiat Jitu Membangun Web Skala Besar*. Jakarta: Elex Media Komputindo. 356 h.

