

ANALISIS SISTEM INFORMASI PENGELOLAAN DATA ALUMNI BERBASIS CODEIGNITER PHP FRAMEWORK

Alcianno G. Gani
localghost@gmail.com

ABSTRACT

Data management of alumni who still use the printed document means that data management can not be done quickly and there are still frequent errors and difficult to update. This problem can be overcome with a method that is developing the programming system. One of them with software. With this information management system software is expected to help the data management of alumni. Furthermore, the information management system software is expected to produce the information needed effectively and efficiently when in making decisions in managing human resources coming from the alumni. In order to be able to establish fellow alumni relationships for the better.

Keywords: alumni data, web application, codeigniter

PENDAHULUAN

Pengelolaan data alumni yang masih menggunakan cara dokumen cetak, mengakibatkan pengelolaan data belum dapat dilakukan dengan cepat dan masih sering terjadi kesalahan serta sulit untuk diperbarui. Permasalahan ini dapat ditanggulangi dengan suatu metode yang sedang berkembang yaitu sistem pemrograman. Salah satunya dengan perangkat lunak.

Dengan perangkat lunak sistem manajemen informasi ini diharapkan dapat membantu pengelolaan data alumni. Selanjutnya, perangkat lunak sistem manajemen informasi ini diharapkan dapat menghasilkan informasi yang dibutuhkan secara efektif dan efisien ketika dalam pengambilan keputusan dalam mengelola sumber daya manusia yang berasal dari alumni. Agar dapat terjalin hubungan sesama alumni menjadi lebih baik.

Dalam hal ini ada beberapa masalah

yang dapat identifikasi, yaitu:

1. Pentingnya melakukan pendataan alumni dengan baik.
2. Kesulitan dalam melakukan pengelolaan data alumni dengan administrasi dalam dokumen cetak, buku, dsb.
3. Kesulitan dalam melakukan pendataan alumni jika pendaftaran harus dilakukan langsung.
4. Perlu adanya analisis kualitas perangkat lunak untuk menjamin kualitas perangkat lunak tersebut.

Berdasarkan uraian di atas tentang pentingnya pengelolaan data yang baik dan efisien maka permasalahan dititikberatkan pada komputerisasi sistem pengelolaan data alumni sekolah. Permasalahan yang diselesaikan dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana membuat perangkat lunak sistem manajemen informasi untuk pengelolaan data alumni?
2. Bagaimana tingkat kualitas perangkat

lunak yang dikembangkan dari sisi *functionality, security, usability, efficiency, portability,* dan *maintainability*?

Pengertian Data dan Informasi

Data dapat didefinisikan sebagai deskripsi dari suatu dan kejadian yang dihadapi (Al-B ahra bin Ladjamudin, 2005). Data dapat berupa catatan-catatan dalam kertas, dokumen, buku, kondisi, situasi, ide, objek, dsb. Data akan menjadi bahan dalam suatu proses pengolahan data, yang kemudian akan diolah menjadi suatu bentuk yang lebih mempunyai arti dan memiliki manfaat. Pengolahan data adalah proses perubahan bentuk data menjadi informasi yang memiliki kegunaan. Semakin banyak data dan kompleksnya aktivitas pengolahan data, maka metode pengelolaan data yang tepat sangat dibutuhkan. Salah satu metode untuk mengolah data adalah dengan menggunakan komputer. Permasalahan yang menggunakan perhitungan matematis atau fungsi-fungsi lainnya, dapat diselesaikan dengan sedikit mungkin kesalahan yang akan terjadi.

Sistem adalah kumpulan dari komponen-komponen yang memiliki unsur keterkaitan antara satu dan lainnya. Sehingga dapat dikatakan bahwa sistem adalah merupakan suatu hal yang saling terkait satu sama lain untuk mencapai sebuah tujuan yang sama (Indrajit, 2000). Informasi, adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang (Davis & Olson, 1974). Kemudian menurut Nugroho (2004), informasi merupakan data yang telah diolah sedemikian rupa sehingga memiliki makna tertentu bagi penggunaannya. Berdasar

definisi di atas maka dapat dikatakan bahwa sebuah informasi adalah data yang mempunyai makna, artinya ketika sesuatu hal (data) tidak mempunyai makna maka belum dapat dikatakan sebagai sebuah informasi. Menurut Oetomo (2002), Sistem Informasi adalah kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi. Secara umum Sistem Informasi merupakan kombinasi dari orang (*people*), perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi (*communications networks*), dan sumber data yang dihimpun, ditransformasi, dan mengalami proses pengaliran dalam suatu organisasi (Kristanto, 2003).

Pengembangan dan Rekayasa Aplikasi Web

Rekayasa Web adalah suatu model rekayasa perangkat lunak (*Software Engineering*) yang digunakan untuk pengembangan aplikasi-aplikasi web. Sebuah aplikasi web adalah suatu sistem perangkat lunak yang berbasiskan teknologi dan standard dari konsorsium World Wide Web (W3C) yang menyediakan sumber yang bersifat spesifik seperti konten atau layanan melalui sebuah tampilan antar muka (*user interface*) yang disebut *Web Browser* (Adri, 2008). Aplikasi web mempunyai beberapa karakteristik yang beberapa diantaranya adalah (Woojong, 2005):

1. Aplikasi web berbeda dengan *Software Aplikasi*. Kontennya terdiri dari berbagai bentuk dan format data seperti teks, grafis, image, audio, video, yang terintegrasikan oleh pemrosesan prosedur (*web programming*), metoda yang digunakan dalam menampilkan dan mengatur

konten tersebut akan berdampak terhadap waktu respon dari sistem (*performance*).

2. Aplikasi web ditujukan untuk digunakan oleh komunitas pengguna yang besar, beragam, dan sejumlah user yang tidak dikenali (*public user*) dengan berbagai kebutuhan, harapan, dan kemampuan. Oleh karena itu pada saat mengembangkan aplikasi web, tampilan antar muka dan fitur-fitur kemudahan dalam penggunaan (*usability features*) harus mampu menjawab kebutuhan dari semua user tersebut tanpa harus melalui suatu program latihan. Semua hal ini merupakan gabungan dari *Human-Web Interaction* (HWI), *user interface*, dan *information presentation*. Semua aplikasi web menuntut aspek “*good look and feel*”, suatu aspek yang memenuhi nilai estetika dan *art* sehingga pengguna merasa nyaman dalam menggunakan dan mengakses aplikasi web yang dikembangkan.
3. Media pengiriman konten untuk aplikasi web sangat berbeda dengan *software* tradisional. Aplikasi web membutuhkan kecocokan dengan berbagai jenis perangkat *display*, format tampilan, dukungan *hardware* dan *software*.
4. Aspek *security* dan privasi lebih dibutuhkan oleh aplikasi berbasis web jika dibandingkan perangkat lunak tradisional.

Kerangka Kerja (*Framework*) Pengembangan Aplikasi Web

PHP adalah bahasa pemrograman *server-side* yang didesain spesifik untuk pengembangan aplikasi berbasis web. Banyak kelebihan dari bahasa pemrograman PHP, antarlain pada aspek performa, skalabilitas, portabilitas, *open source*, dan terutama untuk terkoneksi

dan melakukan manipulasi terhadap sebuah basis data (Supaartagorn, 2011). Manajemen basis data dilakukan dengan *Structure Query Language* (SQL). Beberapa studi menyatakan bahwa bahasa *query database* tradisional tidak mudah digunakan untuk pengguna teknologi basis data yang tidak berpengalaman, sebagai konsekuensi karena interaksinya berbasis bahasa tekstual, seperti SQL (Avensano, Canfora, De Lucia, & Stefanucci, 2002).

Kerangka kerja perangkat lunak (*software framework*) adalah desain dasar yang dapat digunakan dan dikembangkan kembali untuk sebuah sistem atau subsistem aplikasi. Sebuah *software framework* menyediakan kumpulan kode dasar yang dapat membantu dalam proses pengembangan dan penggabungan komponen yang berbeda pada sebuah perangkat lunak (Paikens & Arnicans, 2008). Sebuah kerangka kerja pemrograman dapat menyederhanakan proses menyusun kode fungsi program dengan mengurangi kode operasi-operasi yang bersifat *repetitive* (Upton, 2007). Karena tujuan *framework* adalah membantu melakukan aktivitas umum, banyak *framework* menyediakan pustaka (*library*) untuk akses database, manajemen data sesi, dsb (DocForge, 2010). Kerangka kerja pemrograman web berbasis bahasa pemrograman *PHP-Hypertext Preprocessor* memudahkan proses pengembangan aplikasi, membantu menyusun fungsi-fungsi sebuah sistem dengan waktu lebih cepat karena tidak harus menulisnya dari awal. Ini juga dapat meningkatkan kualitas dan stabilitas susunan kode pemrograman (Yicheng, 2011). *Framework* secara signifikan mengurangi waktu, sumber daya, usaha, yang dibutuhkan untuk mengembangkan dan mengelola aplikasi web. Selain itu, *framework* adalah arsi-

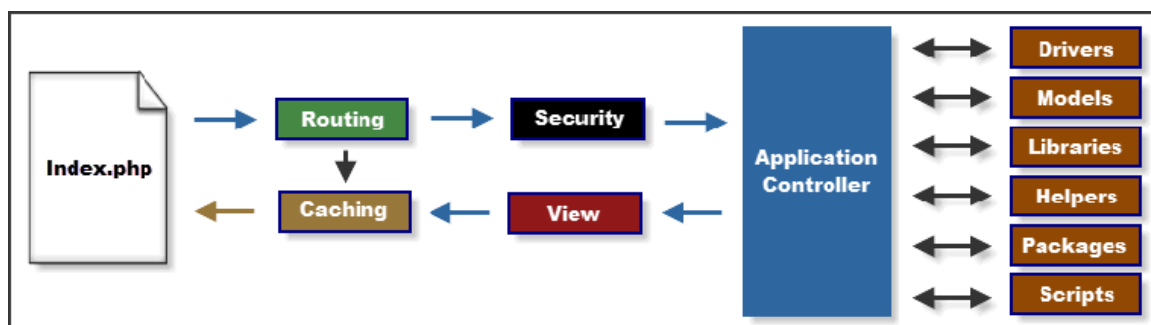
tektur terbuka yang berbasis standard umum yang digunakan (Shan & Hua, 2006).

Dalam desain sistem (*design pattern*), dikenal dengan baik salah satunya pendekatan *Model-View-Controller* (Buschmann, 1996), yang dapat membuat mudah dalam proses pengembangan dan mengelola sebuah aplikasi, karena (Leff & Rayfield, 2001): (1) tampilan (*output*) aplikasi dapat berubah drastis tanpa merubah struktur data dan *business logic*, (2) aplikasi dapat dengan mudah dikelola/digunakan dengan antarmuka (*interfaces*) yang berbeda-beda, misal adalah multi bahasa, atau pengaturan hak akses *user* yang berbeda-beda.

Pendekatan *Model-View-Controller design pattern* adalah cara yang mudah untuk mengembangkan arsitektur sistem perangkat lunak interaktif (Krasner & Pope, 1988). Dikenal juga sebagai *Presentation/Abstraction/Control (PAC) design pattern*, gagasan utamanya adalah untuk memisahkan antarmuka dan data dibawahnya (Coutaz, 1987). Pola *Model-View-Controller* terbukti efektif untuk

menciptakan dan mengorganisir aplikasi *modular* (Hofmeister, Nord, & Soni, 2000).

Salah satu macam kerangka kerja pemrograman web berbasis bahasa pemrograman PHP dan menggunakan pendekatan *Model-View-Controller (MVC)* adalah *CodeIgniter* yang dikembangkan oleh EllisLab, Inc. *CodeIgniter* adalah yang memiliki banyak fitur dengan kebutuhan sistem yang ringan dan dapat mempermudah dalam pengembangan sebuah aplikasi web yang kaya akan *content (rich application)*. Dengan menggunakan perangkat kerangka kerja *CodeIgniter framework*, dapat mengurangi jumlah baris kode program, sehingga dapat meminimalisasi kesalahan penulisan kode program, mengurangi besarnya ukuran *file*, dan mempercepat eksekusi program (Upton, 2007). *Codeigniter framework* memiliki banyak *library* dengan kelebihan lainnya, terutama untuk menangani keamanan data melalui proses *POST data filtering* saat eksekusi dan kemampuan untuk melakukan kompresi dokumen (EllisLab Inc., 2011).



Gambar Arsitektur CodeIgniter Framework
(Sumber: CodeIgniter User Guide)

Cara kerja *framework* Codeigniter dimulai dari awal saat *client* melakukan *request* ke server, kemudian *framework* akan melakukan cek terlebih dahulu pada *routing database* untuk mengetahui *controller* apa yang akan dieksekusi. Jika pernah dieksekusi dan disimpan dalam *cache*, *framework* akan melakukan *redirect* ke *cache* tersebut, sehingga dapat mengurangi waktu proses. Data *input* akan melalui *security layer* dimana akan dilakukan *filtering* dan sanitasi data, sehingga *input* data yang masuk ke dalam sistem sudah bersih dan valid, tidak mengandung kode asing yang dapat membahayakan sistem. Berikutnya, sistem menjalankan *controller* yang diminta, sekaligus menyediakan *library* yang dibutuhkan. Terakhir, *output* data ditampilkan melalui proses *View*.

PEMBAHASAN

Penelitian dan pengembangan adalah suatu proses atau langkah-langkah untuk mengembangkan suatu produk baru atau menyempurnakan produk yang telah ada, yang dapat dipertanggungjawabkan (Sukmadinata, 2010). Metode penelitian dan pengembangan (*Research and Development*) adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut (Sugiyono, 2010).

Dengan demikian, pengembangan lebih diarahkan pada upaya menghasilkan produk siap untuk digunakan secara nyata di lapangan, bukan hanya menemukan pengetahuan atau menguji hipotesis atau teori tertentu. Penelitian dan pengembangan yang dilakukan difokuskan pada pembuatan perangkat lunak sistem informasi untuk pengelolaan data alumni pada Sekolah Menengah Atas (SMA).

Tahap Analisis Kebutuhan

Pada tahap awal ini dilakukan analisis kebutuhan awal mengenai kebutuhan serta problem-problem yang perlu diselesaikan. Dibutuhkan berbagai sumber informasi mengenai kasus dan kebutuhan yang dibutuhkan oleh pengguna, diantaranya pada pihak sekolah, pengelola administrasi sekolah, serta alumni sekolah yang bersangkutan. Kemudian dibuat daftar permintaan atau kebutuhan pengguna (*user requirement list*) yang perlu disediakan pada aplikasi pengelolaan data alumni sekolah yang dikembangkan dalam penelitian ini.

Tahap Desain

Berdasarkan dari analisis kebutuhan maka dapat diketahui apa saja yang menjadi kebutuhan dari pengembangan sistem informasi pengelolaan data alumni untuk sekolah, sehingga sistem yang dibuat nantinya sesuai dengan apa yang diharapkan. Tahap desain meliputi:

- a) Perancangan Unified Modeling Language (UML)
Perancangan cara kerja program menggunakan UML yang meliputi desain *Use Case Diagram* dan *Class Diagram*.
- b) Perancangan Diagram Alir (*Flowchart*)
Diagram (Flowchart) digunakan untuk memperjelas perancangan dan algoritma yang dibuat.
- c) Perancangan antar muka pengguna (*User Interface*)
Rancangan *interfaces* dari aplikasi web ini yang menggunakan CSS terintegrasi dengan HTML.
- d) Perancangan desain basis data.
Mendesain tabel-tabel data yang dibutuhkan serta menjelaskan relasi antar tabel.

Tahap Implementasi

Implementasi merupakan tahap saat pembuatan aplikasi dimulai setelah dilakukan analisis dan desain. Rancangan program yang telah disiapkan kemudian diimplementasikan dalam bahasa pemrograman, sehingga semua fungsi dapat dijalankan dengan baik oleh pengguna.

Tahap Pengujian

Pada tahap ini aplikasi yang telah dikembangkan kemudian diberikan berbagai rangkaian pengujian kualitas perangkat lunak yang menggunakan beberapa instrumen penelitian sesuai standard ISO 9126, sehingga dapat dilakukan evaluasi sistem sebelum akhirnya dapat digunakan oleh banyak pengguna.

Untuk mengetahui kualitas perangkat lunak yang dikembangkan dalam penelitian ini, perangkat lunak diuji dan dianalisis memakai standard ISO 9126, terutama pada aspek *functionality*, *security*, *usability*, *efficiency*, *maintainability*, dan *portability*.

a) Pengujian *Functionality*

Pengujian ini berfokus pada kesesuaian satu set fungsi untuk dapat melakukan tugas-tugas tertentu. Pengujian ini menggunakan metode *checklist* yang dilakukan pada 5 responden ahli dengan kriteria responden memiliki pekerjaan sehari-hari sebagai pengembang aplikasi *web*.

b) Pengujian *Security*

Pengujian ini berfokus pada jaminan kemampuan dalam mencegah akses yang

tidak sah, baik secara sengaja maupun tidak disengaja. Pengujian ini dilakukan untuk menemukan berbagai celah keamanan. Pengujian akan dilakukan dengan menggunakan perangkat lunak khusus untuk menguji kualitas sebuah aplikasi web dari sisi *security*.

c) Pengujian *Usability*

Pengujian ini dilakukan dengan menilai seberapa mudah tampilan antarmuka, navigasi, pada perangkat lunak yang dikembangkan untuk digunakan. Pengujian ini menggunakan kuisioner yang dilakukan pada 15 responden dengan kriteria bahwa responden tersebut terbiasa dengan aktifitas internet.

d) Pengujian *Efficiency*

Pengujian aspek efisiensi yang dilakukan adalah kualitas *performance* aplikasi saat diakses pengguna (*client side*), antarlain adalah kecepatan akses, pemakaian *resources*, dan kecepatan proses data saat eksekusi. Pengujian ini menggunakan perangkat lunak khusus untuk pengukuran besar dokumen kemudian dilakukan analisis *load test*.

e) Pengujian *Maintainability*

Pengujian aspek *maintainability* yang dilakukan adalah dengan menguji perangkat lunak pada aspek *instrumentation*, *consistency*, dan *simplicity*.

f) Pengujian *Portability*

Pengujian untuk aspek *portability* ini dilakukan dengan menjalankan sistem pengolahan data pada *browser* berbasis *desktop* dan pada *browser* berbasis *mobile*.

1. Instrumen Functionality

Tabel 4. Instrumen *Functionality*

No.	Fungsi (kebutuhan)	Pertanyaan
1.	Navigasi	Apakah menu navigasi utama dapat difungsikan?
2.	Berita / Informasi	Apakah fungsi untuk mengakses berita sudah berfungsi dengan benar?
3.	Pencarian Data	Apakah fungsi proses pencarian cepat profil alumni sudah berfungsi dengan benar?
4.	Statistik	Apakah pada halaman depan informasi statistik jumlah data alumni sudah berfungsi dengan benar?
5.	Registrasi	Apakah fungsi registrasi pengguna baru sudah berfungsi dengan benar?
6.	Log in / Otentifikasi	Apakah fungsi <i>login</i> masuk ke dalam sistem sudah berfungsi dengan benar?
7.	Pengelolaan Data	Apakah fungsi untuk menyunting data profil alumni sudah berfungsi dengan benar?
8.	Pengelolaan Data	Apakah fungsi untuk mengunggah foto ke data profil alumni sudah berfungsi dengan benar?
9.	Pengelolaan Data	Apakah fungsi untuk melihat profil alumni sudah berfungsi dengan benar?
10.	Pengelolaan Data	Apakah fungsi untuk merubah kata kunci (<i>password</i>) untuk masuk dalam sistem (<i>login</i>) sudah berfungsi dengan benar?
11.	Pencarian Data	Apakah fungsi untuk mencari data alumni sudah berfungsi dengan benar?
12.	Berita / Informasi	Apakah fungsi untuk mengakses informasi beasiswa sudah berfungsi dengan benar?
13.	Statistik	Apakah fungsi untuk mengakses statistik lengkap data alumni terdaftar sudah berfungsi dengan benar?
14.	Kontak	Apakah fungsi untuk menghubungi pengelola situs sudah berfungsi dengan baik?
15.	Cetak Data	Apakah fungsi untuk mencetak data alumni sudah berfungsi dengan baik?

2. Instrumen *Security*

Pengujian ini akan menggunakan perangkat lunak *Acunetix Web Vulnerability Scanner* versi 8. Pengujian akan dilakukan dengan menggunakan

parameter pengujian *Default* untuk menguji dan menemukan berbagai jenis celah keamanan. Pengujian dilakukan dengan menggunakan *login sequence* untuk sebuah pengguna biasa.

Tabel 5. Instrumen *Security*

No.	Modul dan Manipulasi Parameter	Aktif
1.	<i>Cross Site Scripting (XSS)</i>	Ya
2.	<i>SQL Injection</i>	Ya

3. Instrumen *Usability*

Pengujian ini menggunakan kuisioner *Computer System Usability Questionnaire* (CSUQ) yang

dikembangkan oleh IBM untuk standar pengukuran *usability* perangkat lunak (Lewis, 1993).

Tabel 6. Instrumen *Usability*

No.	Pertanyaan
1	Secara keseluruhan, saya puas dengan kemudahan pemakaian aplikasi ini.
2	Sangat sederhana penggunaan aplikasi ini.
3	Saya dapat dengan sempurna menyelesaikan pekerjaan dengan aplikasi ini.
4	Saya dapat menyelesaikan pekerjaan saya dengan cepat menggunakan aplikasi ini.
5	Saya dapat menyelesaikan pekerjaan saya secara efisien menggunakan aplikasi ini.
6	Saya merasa nyaman menggunakan aplikasi ini.
7	Sangat mudah mempelajari penggunaan aplikasi ini.
8	Saya yakin saya bisa menjadi produktif dengan cepat berkat aplikasi ini.
9	Pesan kesalahan yang diberikan aplikasi ini menjelaskan dengan gambling cara mengatasinya.
10	Kapanpun saya membuat kesalahan, saya bisa memperbaikinya dengan cepat dan mudah.
11	Informasi yang disediakan aplikasi ini cukup jelas.
12	Sangat mudah mencari informasi di aplikasi ini.
13	Informasi yang disediakan aplikasi sangat mudah dipahami.
14	Informasi yang disediakan efektif membantu saya menyelesaikan tugas dan skenario.
15	Pengorganisasian informasi yang ditampilkan aplikasi jelas.
16	Antarmuka aplikasi menyenangkan.
17	Saya menyukai menggunakan antarmuka aplikasi ini.
18	Aplikasi ini memiliki fungsi dan kapabilitas sesuai harapan saya.
19	Secara keseluruhan, saya puas dengan aplikasi ini.

4. Instrumen *Efficiency*

Pengujian ini menggunakan alat ukur YSlow yang dikembangkan oleh Yahoo Developer Network untuk mengukur performa efisiensi sebuah halaman

website. Performa yang akan diukur adalah besarnya *bytes* data dokumen, jumlah HTTP request, minifikasi, kompresi GZIP, dan *score / grade* akhir (Yahoo Developer Network, 2011).

Tabel 7. Instrumen *Efficiency*

No.	Parameter Dasar YSlow	Aktif
1	<i>Make fewer HTTP requests</i>	Ya
2	<i>Compress components with GZIP</i>	Ya
3	<i>Minify JavaScript and CSS</i>	Ya
4	<i>Reduce DNS lookups</i>	Ya
5	<i>Reduce cookie size</i>	Ya
6	<i>Reduce the number of DOM elements</i>	Ya
7	<i>Configure entity tags (ETags)</i>	Ya
8	<i>Use cookie-free domains</i>	Ya
9	<i>Make JavaScript and CSS external</i>	Ya

5. Instrumen *Maintainability*

Pengujian untuk aspek *maintainability* ini menggunakan ukuran-ukuran (*metrics*). Kemudian pengujian dilakukan peneliti dengan diuji secara operasional (Land, 2002). Instrumen pengujian dapat dilihat pada tabel dibawah ini.

Tabel 8. Instrumen *Maintainability*

Aspek	Aspek yang dinilai	Hasil yang akan diperoleh
<i>Instrumentation</i>	Terdapat peringatan pada sistem pengolah data untuk mengidentifikasi kesalahan	Ketika ada kesalahan yang dilakukan oleh user, maka sistem akan mengeluarkan peringatan untuk mengidentifikasi kesalahan.
<i>Consistency</i>	Penggunaan satu bentuk rancangan pada seluruh rancangan sistem	Bentuk rancangan sistem pengolah data mempunyai satu bentuk yang sama. Hal ini dapat dilihat pada bagian implementasi sistem.
<i>Simplicity</i>	Kemudahan dalam pengelolaan, perbaikan, dan pengembangan sistem	Mudah untuk dikelola, diperbaiki, dan dikembangkan. Hal ini dapat dilihat pada tahapan-tahapan proses penulisan kode program.

6. Instrumen *Portability*

Pengujian untuk aspek *portability* ini dilakukan dengan menjalankan sistem

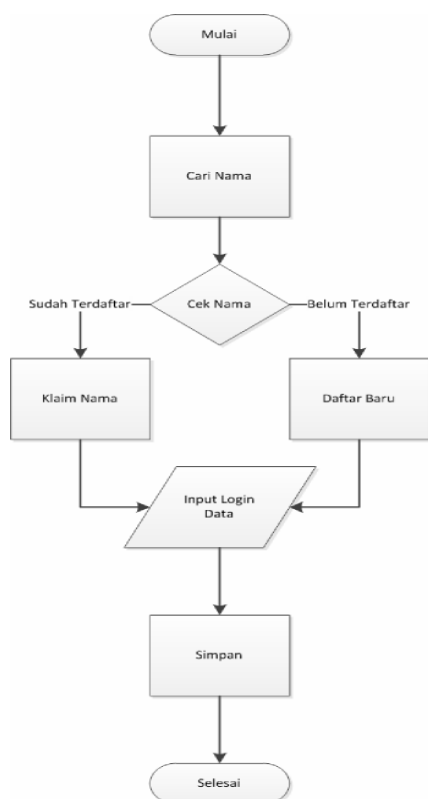
pada *browser* berbasis *desktop* dan pada *browser* berbasis *mobile* (World Wide Web Consortium, 2012).

Tabel 9. Instrumen *Portability*

Aspek yang dinilai	Hasil yang akan diperoleh
Sistem dapat berjalan pada <i>browser</i> berbasis desktop	Sistem kompatibel dengan beberapa browser ternama. Hal ini terbukti dari hasil pengujian bahwa sistem dapat di akses di beberapa browser seperti <i>Mozilla Firefox</i> , <i>Internet Explorer</i> , <i>Opera</i> , dan <i>Google Chrome</i> tanpa terdapat pesan error.
Sistem dapat berjalan pada <i>browser</i> berbasis mobile	Sistem dapat diakses melalui <i>browser</i> berbasis mobile yaitu <i>Opera Mini</i> tanpa terdapat pesan error.

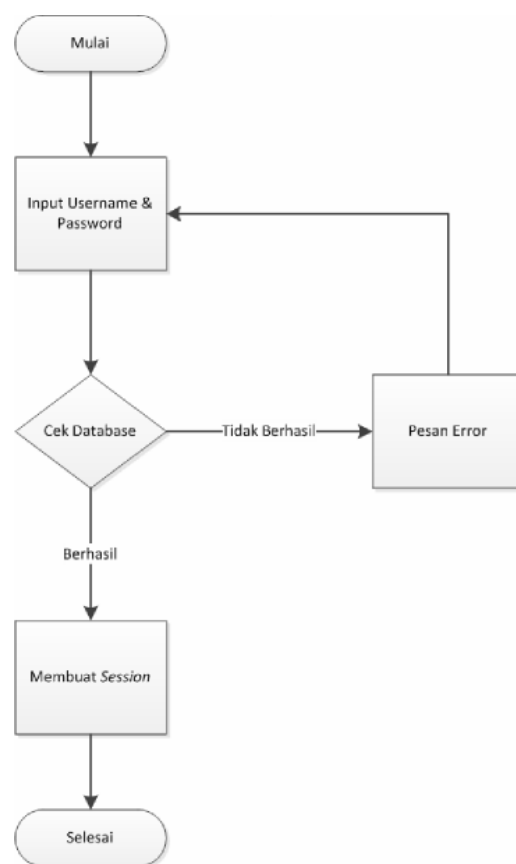
Perancangan Diagram Alir (Flowchart)

a) *Flowchart* proses registrasi
Flowchart yang digunakan untuk menjelaskan proses registrasi berjalan.



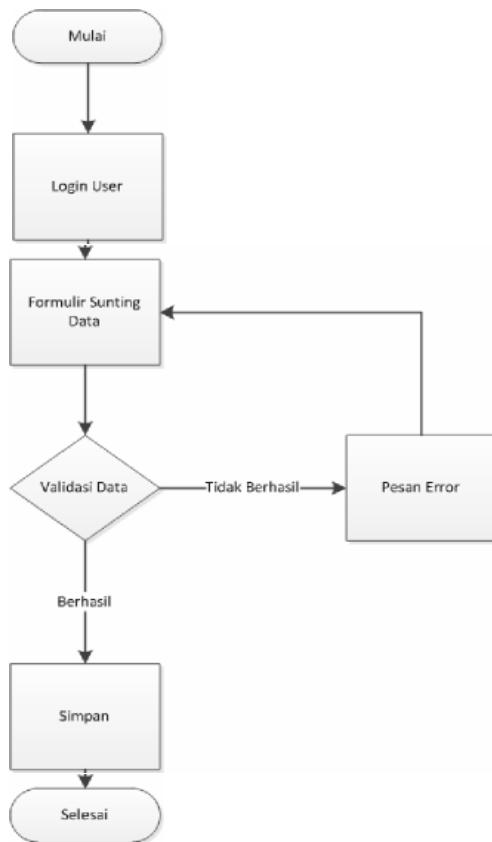
Gambar 16. *Flowchart* proses registrasi

b) *Flowchart* proses *Log in*
Flowchart yang digunakan untuk menjelaskan proses *Log in* berjalan.



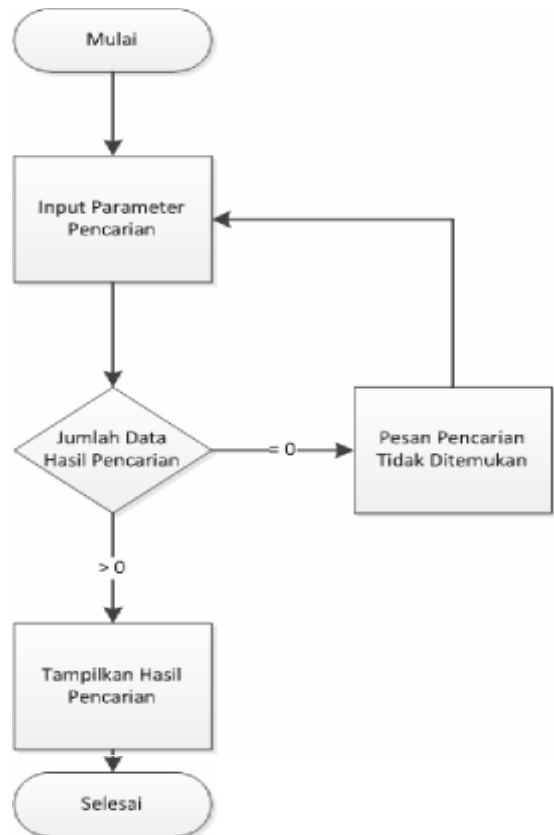
Gambar 17. *Flowchart* proses *Log in*

c) *Flowchart* proses sunting data
Flowchart yang digunakan untuk menjelaskan proses sunting data berjalan.



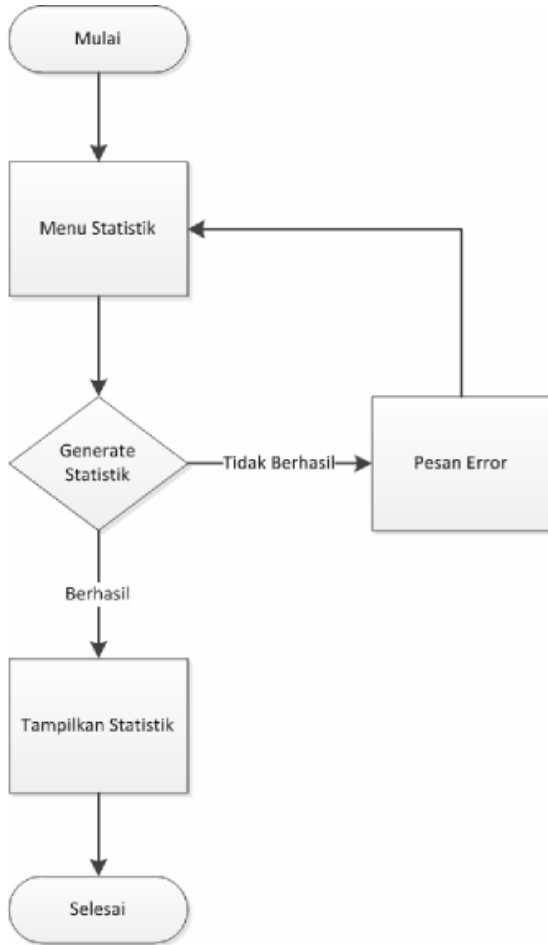
Gambar 18. *Flowchart* proses sunting data

d) *Flowchart* proses pencarian data
Flowchart yang digunakan untuk menjelaskan proses pencarian data berjalan.



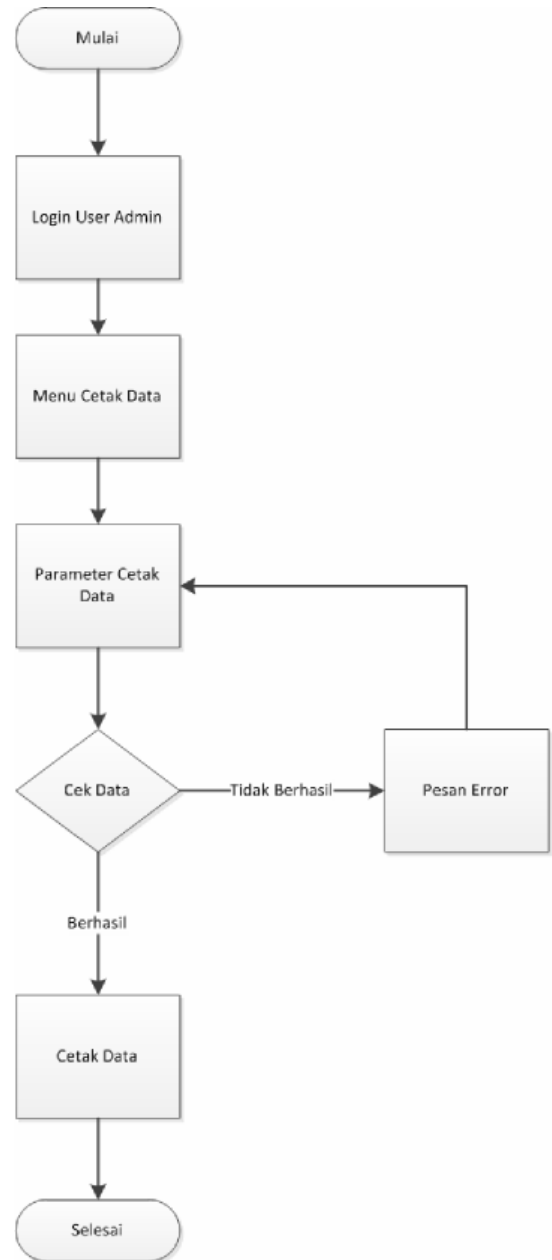
Gambar 19. *Flowchart* proses pencarian data

e) *Flowchart* proses statistik
Flowchart yang digunakan untuk menjelaskan proses statistik berjalan.



Gambar 20. *Flowchart* proses statistik

f) *Flowchart* proses cetak data
Flowchart yang digunakan untuk menjelaskan proses cetak data berjalan.



Gambar 21. *Flowchart* proses cetak data

Perancangan *User Interface* (Antar Muka Pengguna).

a) Halaman Depan

Pada rancangan halaman depan memuat judul aplikasi, kolom navigasi, kotak login, kotak berita, serta menu pencarian cepat

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

Login
Email
Password
Login [Registrasi](#)

[Halaman Depan](#)
[Cari Data](#)
[Statistik](#)
[Kontak](#)

Header Foto

Informasi Terbaru
Judul Berita
Tanggal
Isi Berita

Pencarian Cepat
Enter Text

Daftar Link
[Hyperlink](#)

Statistik
Jumlah: 5999
Update: 12-01-1999

Footer Copyright [Feedback](#)

Gambar Rancangan Halaman Depan

b) Halaman Registrasi Langkah ke-1

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

Login
Email
Password
Login [Registrasi](#)

[Halaman Depan](#)
[Cari Data](#)
[Statistik](#)
[Kontak](#)

Registrasi Alumni Step 1

Periksa Nama
Enter text Periksa Nama

Footer Copyright [Feedback](#)

Gambar Rancangan Halaman Registrasi Langkah ke-1

c) Halaman Registrasi Langkah ke-2

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

<p>Login</p> <input type="text" value="Email"/> <input type="password" value="Password"/> <p><input type="button" value="Login"/> Registrasi</p> <p>Halaman Depan</p> <p>Cari Data</p> <p>Statistik</p> <p>Kontak</p>	<p>Registrasi Alumni Step 2</p> <p>Hasil Pencarian Nama Terdaftar</p> <table border="1"><tr><th>Nama</th></tr><tr><td>Text</td></tr><tr><td>Text</td></tr><tr><td>Text</td></tr><tr><td>Text</td></tr></table> <p>Daftar Baru</p>	Nama	Text	Text	Text	Text
Nama						
Text						
Text						
Text						
Text						

Footer Copyright [Feedback](#)

Gambar Rancangan Halaman Registrasi Langkah ke-2

d) Halaman Registrasi Langkah ke-3

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

<p>Login</p> <input type="text" value="Email"/> <input type="password" value="Password"/> <p><input type="button" value="Login"/> Registrasi</p> <p>Halaman Depan</p> <p>Cari Data</p> <p>Statistik</p> <p>Kontak</p>	<p>Registrasi Alumni Step 3</p> <p>Registrasi Data Alumni</p> <p>Nama Lengkap: <input type="text" value="Enter Text"/></p> <p>Email: <input type="text" value="Enter Text"/></p> <p>Password: <input type="password" value="Enter Text"/></p> <p><input type="button" value="Daftar"/></p>
---	--

Footer Copyright [Feedback](#)

Gambar Rancangan Halaman Registrasi Langkah ke-3

e) Halaman Registrasi Langkah ke-4

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

Registrasi Alumni Step 4

Registrasi telah berhasil, silahkan login dengan menggunakan email dan password Anda

Login

Email

Password

Login Registrasi

[Halaman Depan](#)

[Cari Data](#)

[Statistik](#)

[Kontak](#)

Footer Copyright [Feedback](#)

Gambar Rancangan Halaman Registrasi Langkah ke-4

f) Halaman *Log in* pengguna

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

Login

Email

Password

Login

Gambar Rancangan Halaman *Log in* pengguna

g) Halaman Sunting Data

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

Informasi Pengguna Nama User Logout Halaman Depan Cari Data Statistik Kontak	Mengedit Data Alumni Nama Lengkap: <input type="text" value="Enter Text"/> Email: <input type="text" value="Enter Text"/> Password: <input type="text" value="Enter Text"/> TTL: <input type="text" value="Enter Text"/> Alamat: <input type="text" value="Enter Text"/> Pendidikan: <input type="text" value="Enter Text"/> <input type="button" value="Simpan"/>
---	--

Footer Copyright [Feedback](#)

Gambar Rancangan Halaman Sunting Data

h) Halaman Tampilan Profil

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

Informasi Pengguna Nama User Logout Halaman Depan Cari Data Statistik Kontak	Lihat Data Alumni Nama Lengkap: Fulan Email: fulan@test.com TTL: Yogyakarta, 15 November 1988 Alamat: Yogyakarta Pendidikan: PTU UNY <input type="button" value="Edit"/>
---	---

Footer Copyright [Feedback](#)

Gambar Rancangan Halaman Tampilan Profil

i) Halaman Pencarian Data Alumni

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

<p>Informasi Pengguna</p> <p>Nama User Logout</p>	<p>Pencarian Data Alumni</p> <p>Nama Lengkap: <input type="text" value="Enter Text"/></p> <p><input type="button" value="Cari"/></p>
<p>Halaman Depan</p> <p>Cari Data</p> <p>Statistik</p> <p>Kontak</p>	
<p>Footer Copyright</p>	<p>Feedback</p>

Gambar Rancangan Halaman Pencarian Data Alumni

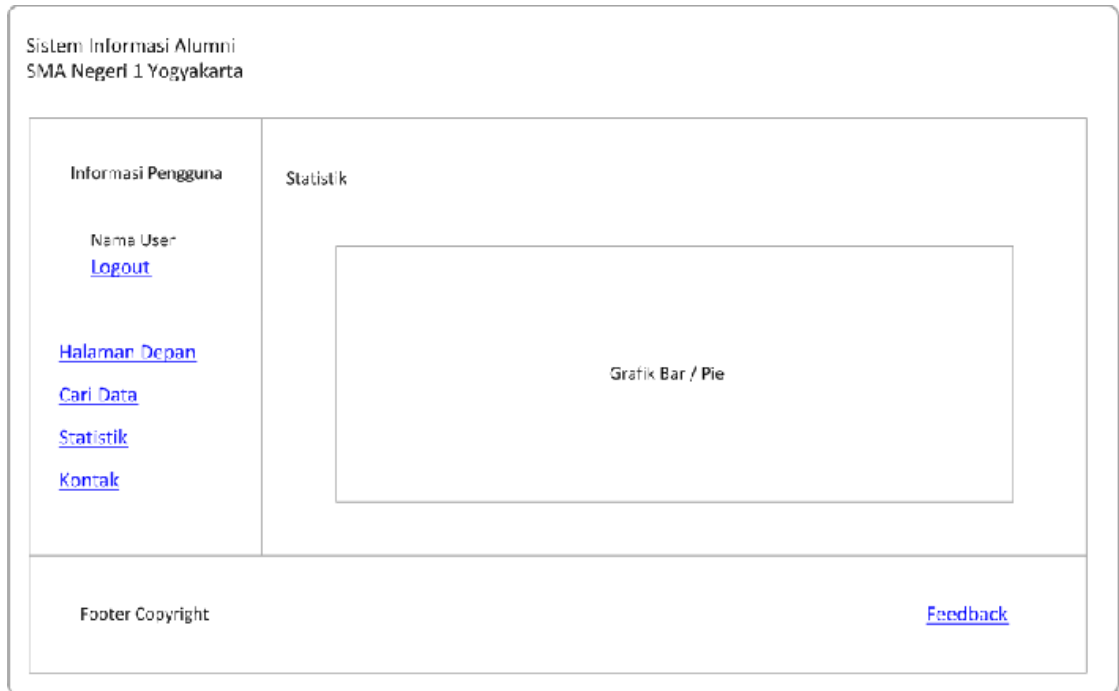
j) Halaman Hasil Pencarian Data

Sistem Informasi Alumni
SMA Negeri 1 Yogyakarta

<p>Informasi Pengguna</p> <p>Nama User Logout</p>	<p>Hasil Pencarian Data Alumni</p> <table border="1"><thead><tr><th>Nama</th></tr></thead><tbody><tr><td>Text</td></tr><tr><td>Text</td></tr><tr><td>Text</td></tr><tr><td>Text</td></tr></tbody></table>	Nama	Text	Text	Text	Text
Nama						
Text						
Text						
Text						
Text						
<p>Halaman Depan</p> <p>Cari Data</p> <p>Statistik</p> <p>Kontak</p>						
<p>Footer Copyright</p>	<p>Feedback</p>					

Gambar Rancangan Halaman Hasil Pencarian Data

k) Halaman Statistik



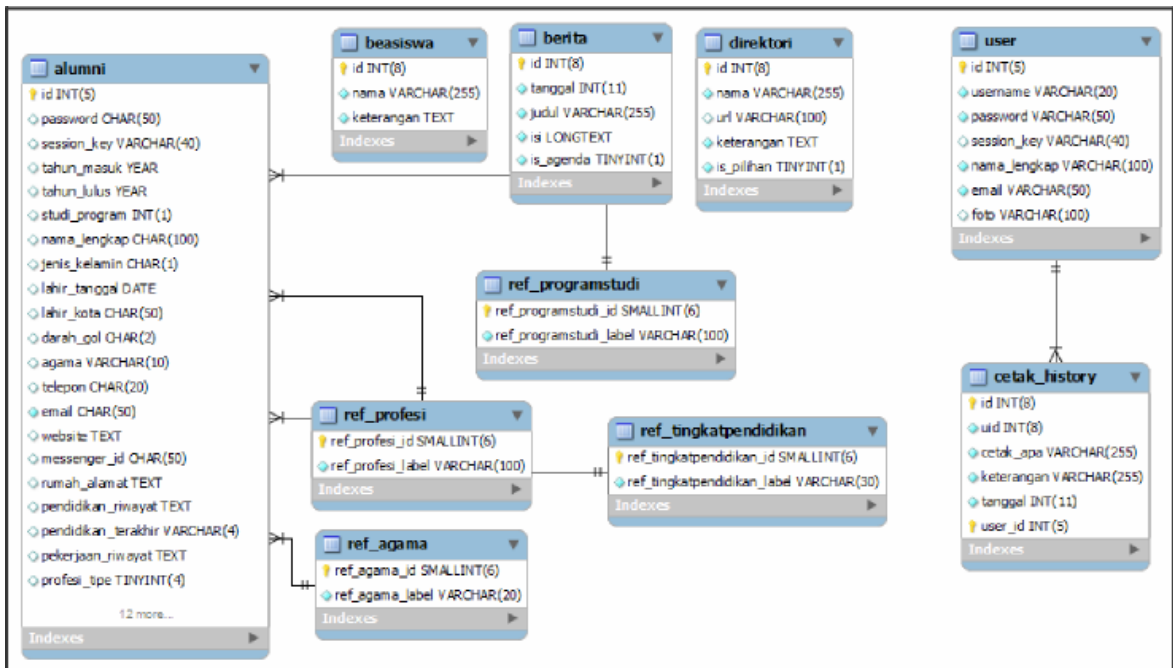
Gambar Rancangan Halaman Statistik

l) Halaman Cetak Data Alumni



Gambar Rancangan Halaman Rancangan Cetak Data

Perancangan Desain Basis Data



Gambar Rancangan Desain Basis Data

KESIMPULAN

Berdasarkan hasil pembahasan yang telah dilakukan, dapat disimpulkan bahwa:

1. Dengan menggunakan *framework* CodeIgniter PHP dapat dibuat sebuah perangkat lunak aplikasi *web* sistem informasi untuk membantu pengelolaan data. Dalam penelitian ini, perangkat lunak dikembangkan untuk membantu mengelola data alumni sekolah.
 - a. Kualitas perangkat lunak yang dikembangkan dari sisi *functionality* adalah sudah cukup baik, setelah dilakukan pengujian, didapatkan 94,6% fungsi telah berjalan dengan benar sesuai kebutuhan.
 - b. Dengan menggunakan *security layer* pada CodeIgniter PHP *framework* dapat dilakukan *data filtering* untuk mencegah eksploitasi celah keamanan yang antara lain berupa *Cross-site Scripting (XSS)* dan *SQL Injection*. Hal ini bisa dilihat dari hasil pengujian yang hanya menghasilkan peringatan jenis rendah (*Low Level*). Jadi kualitas perangkat lunak yang dikembangkan dari sisi *security* sudah cukup bagus.
 - c. Kualitas perangkat lunak yang dikembangkan dari sisi *usability* sudah cukup baik, dimana setelah dilakukan pengujian, nilai *usability* yang didapatkan menyatakan 79% pengguna dapat dengan mudah menggunakan sistem. Dari skor persentase yang didapat maka

kualitas dari sisi *usability* perangkat lunak telah sesuai dengan yang diharapkan pada atribut *usability*.

- d. *Framework* CodeIgniter PHP dapat melakukan kombinasi dokumen, minifikasi data, dan melakukan kompresi GZIP sebelum data dikirim dari *server* ke *client*. Hal ini dapat mengurangi jumlah *HTTP Request* dan mengurangi besar data dokumen secara signifikan (rata-rata dibawah 50 Kilobytes). Dari pengujian yang dilakukan, didapatkan 70% halaman *web* telah memiliki tipe *A grade* dengan skor di atas 90 dan 30% halaman *web* telah memiliki tipe *B grade* dengan skor diantara 80-89. Dengan menggunakan data faktor kemauan pengguna dalam menunggu *load* dari sebuah website, didapatkan sebanyak minimal 84% pengguna tidak akan pergi meninggalkan halaman web tersebut. Dengan demikian kualitas perangkat lunak yang dikembangkan dari sisi *efficiency* sudah bagus.
- e. *Framework* CodeIgniter PHP yang digunakan memiliki kemampuan untuk melakukan validasi data *input*, sehingga setiap pengguna memberi masukan data, sistem akan memberi peringatan jika data yang dimasukkan tidak sesuai. Dengan adanya peringatan ini, dapat membantu pengguna untuk memperbaikinya kembali. Halaman-halaman *web* yang dikembangkan juga terlihat konsisten secara bentuk, warna, tata letak, dsb. Terakhir,

framework CodeIgniter PHP dengan pendekatan *ModelView-Controller* membuat proses penulisan kode program saat pengembangan aplikasi berbasis web menjadi lebih mudah. Jadi kualitas perangkat lunak yang dikembangkan dari sisi *maintainability* sudah baik

- f. Kualitas perangkat lunak yang dikembangkan dari sisi *portability* sudah baik, dapat dilihat dari keseluruhan halaman aplikasi web dapat diakses tanpa *error* dengan menggunakan berbagai *web browser* baik *desktop* maupun *mobile*.

REFERENSI

- Aaby, A. (2004). *Quality Characteristics*. Retrieved Januari 5, 2012, from Software Acquisition: Software from the Customer's Perspective: <http://cs.wallawalla.edu/~aabyan/Colloquia/Acquisition/qualities.html> (diakses pada tanggal 20 Februari 2016)
- Acunetix. (2011). *Audit Your Website Security With Acunetix Web Vulnerability Scanner*. Retrieved Januari 5, 2012, from <http://www.acunetix.com/vulnerability-scanner/> (diakses pada tanggal 20 Februari 2016)
- Adri, M. (2008). *Konsep Dasar Web Engineering*. Retrieved Januari 5, 2012, from <http://muhammadadri.files.wordpress.com/2008/04/01-materi-1.pdf>

- Al-Bahra bin Ladjamudin. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Anley, C. (2002). Advanced SQL Injection In SQL Server Applications. *An NGSSoftware Insight Security Research (NISR) Publication*.
- Avensano, L., Canfora, G., De Lucia, A., & Stefanucci, S. (2002). Understanding SQL Through Iconic Interfaces. *Computer Software and Applications Conference (COMPSAC)*, (pp. 703-708).
- Buschmann, F. (1996). Pattern-Oriented Software Architecture: A System of Patterns. 123-168.
- Centre for Software Engineering. (1991). *ISO/IEC 9126 : Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for Their Use*. Retrieved Januari 5, 2012, from ISO 9126: The Standard of Reference: <http://www.cse.dcu.ie/essiscope/s/m2/9126ref.html>
- Coutaz, J. (1987). PAC, An Object-Oriented Model for Dialog Design. *Proceedings of Human-Computer Interaction (INTERACT)* (pp. 431-436). Elsevier Science Publishers.
- Davis, G. B., & Olson, H. M. (1974). *Management Information System: Conceptual Foundation, Structure, and Development*. Auckland: McGraw-Hill International Book Company.
- Departemen Pendidikan Nasional. (2002). *Manajemen Peningkatan Mutu Berbasis Sekolah, Konsep Dasar*. Jakarta: Ditjen Pendidikan Dasar dan Menengah.
- EllisLab Inc. (2011). *CodeIgniter User Guide Version 2.1.0*. Retrieved Januari 5, 2012, from http://codeigniter.com/user_guide/index.html
- Hofmeister, C., Nord, R. L., & Soni, D. (2000). *Applied Software Architecture*. Addison-Wesley.
- Huang, Y. W., Huang, S. K., Lin, T. P., & Tsai, C. H. (2003). Web Application Security Assessment by Fault Injection and Behavior Monitoring. *Proceedings of the 12th International Conference on World Wide Web* (pp. 148-159). New York, NY, USA: ACM.
- Indrajit, R. E. (2000). *Manajemen Sistem Informasi dan Teknologi Informasi : Pengantar Konsep*. Jakarta: Gramedia.
- Kan, Z. (2010). *Web Interoperability, An Obstruction of FLOSS Development*. Tsinghua University.
- Krasner, G. E., & Pope, S. T. (1988). A Cookbook for Using the Model-View-Controller User-Interface Paradigm in Smalltalk-80. *Journal of Object-Oriented Programming* (pp. 26-49). SIGS Publication.

- Kristanto, A. (2003). *Perancangan Sistem Informasi*. Yogyakarta: Gava Media.
- Land, R. (2002). *Measurements of Software Maintainability*. (ICEBE'06).
- Spinellis, D. D. (2006). *Code Quality: The Open Source Perspective*. Boston: Addison-Wesley.
- Subraya, B. M. (2006). *Integrated Approach to Web Performance Testing: A Practitioner's Guide*. Idea Group Inc.
- Sugiyono. (2010). *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta.
- Sukmadinata, N. S. (2010). *Metode Penelitian Pendidikan*. Bandung: Remaja Rosdakarya.
- Supaartagorn, C. (2011). PHP Framework For Database Management Based On MVC Pattern. *International Journal of Computer Science & Information Technology (IJCSIT) Vol 3 No 2*, 251-258.
- Sutanta, E. (2011). *Basis Data Dalam Tinjauan Konseptual*. Yogyakarta: ANDI.
- Upton, D. (2007). *CodeIgniter for Rapid PHP Application Development*. Birmingham: Packt Publishing.
- Woojong, S. (2005). *Web Engineering: Principles and Techniques*. USA: Idea Group Publishing.
- Yahoo Developer Network. (2011). *Best Practices for Speeding Up Your Web Site*. Retrieved Januari 5, 2012, from <http://developer.yahoo.com/performance/rules.html>
- Kristanto, A. (2003). *Perancangan Sistem Informasi*. Yogyakarta: Gava Media.
- Land, R. (2002). *Measurements of Software Maintainability*.
- Leff, A., & Rayfield, J. T. (2001). Web-Application Development Using the Model/View/Controller Design Pattern. *Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International*, (pp. 118-127). Seattle, WA.
- Lewis, J. R. (1993). *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use*. Boca Raton: IBM Corporation.
- McCall, J. A., Richards, P. K., & Walters, G. F. (1977). *Factors in Software Quality*. US Rome Air Development Center Reports.
- Nielsen, J. (2003). *Introduction to Usability*. Retrieved Januari 1, 2012, from <http://www.useit.com/alertbox/20030825.html>
- Paikens, A., & Arnicans, G. (2008). *Use of Design Patterns in PHP-Based Web Application Frameworks*. Department of Computing University of Latvia.
- Shan, T. C., & Hua, W. W. (2006). Taxonomy of Java Web Applications Frameworks. *IEEE International Conference on e-Business Engineering*

Yicheng, L. (2011). *Development of a Blog System Using CodeIgniter Framework*. Finland: Oulu University of Applied Sciences.

<http://www.isixsigma.com/tools-templates/qfd-house-of-quality/software-quality-function-deployment/>

Zyrmiak, D. (2001). *Software Quality Function Deployment*. Retrieved Januari 5, 2012, from