

PENERAPAN POLA HIERACHICAL MODEL-VIEW-CONTROLLER PADA REKAYASA SISTEM BERBASIS WEB FRAMEWORK

Eka Wahyu Hidayat¹

¹Jurusan Teknik Informatika, Universitas Siliwangi Tasikmalaya

Masuk: 12 Nopember 2010, revisi masuk : 3 Januari 2011, diterima: 24 Januari 2011

ABSTRACT

Found the problem when doing coding to build a web-based application. To speed up the work, web developer typically uses some code from previous projects, especially for the part related to databases, frame systems, and security systems. If there is weakness of unknown code on the previous project, the weakness will be downgraded to the next project. To overcome these problems, most IT companies now use a web framework for web development because it offers many advantages compared to when writing code from scratch. Web framework has several advantages such as having a pattern MVC or Model-View-Controller. One of the web framework that supports the MVC pattern is CodeIgniter. However, using the MVC pattern alone is not sufficient, to produce a modular program needs to be applied Hierarchical MVC pattern with the addition of Modular Extension in CodeIgniter as did developers BackendPro is a freely licensed applications that provide functionality to perform all the simple tasks usually done over and over re-engineering project in the web such as authentication, permissions, and control panels. In this research will be reviewed as a basis for utilization BackendPro early in the development of web-based applications. The study showed the steps to build a simple application into a web framework using BackendPro and benefits to be gained when developing web applications using a web framework.

Keywords: Model-View-Controller, Hierarchical, Framework, Codeigniter, BackendPro

INTISARI

Ditemukan permasalahan ketika melakukan *coding* untuk membangun suatu aplikasi berbasis web. Untuk mempercepat pekerjaan, *developer* sebagai pengembang web biasanya menggunakan sebagian kode dari proyek sebelumnya terutama untuk bagian yang berhubungan dengan basis data, kerangka sistem, dan keamanan sistem. Jika ada kelemahan kode yang tidak diketahui pada proyek sebelumnya, maka kelemahan tersebut akan diturunkan ke proyek berikutnya. Untuk mengatasi permasalahan tersebut, sekarang ini kebanyakan perusahaan IT menggunakan *web framework* untuk pembangunan web karena menawarkan banyak kelebihan dibandingkan jika menulis kode program dari awal. *Web framework* memiliki beberapa kelebihan diantaranya memiliki pola MVC atau *Model-View-Controller*. Salah satu *web framework* yang mendukung pola MVC adalah *Codeigniter*. Akan tetapi menggunakan pola MVC saja belum cukup, untuk menghasilkan program yang modular perlu diterapkan pola *Hierarchical MVC* dengan penambahan *Modular Extension* pada *Codeigniter* seperti yang dilakukan pengembang *BackendPro* yaitu suatu aplikasi berlisensi bebas yang menyediakan fungsionalitas untuk melakukan semua tugas-tugas sederhana yang biasa dikerjakan berulang-ulang dalam proyek rekayasa web seperti *authentication*, *permissions*, dan *panel control*. Penelitian ini dikaji pemanfaatan *BackendPro* sebagai landasan pembangunan aplikasi berbasis web dan didapat langkah untuk membangun sebuah aplikasi sederhana kedalam *web framework* menggunakan *BackendPro* dan keuntungan yang dapat diperoleh jika mengembangkan aplikasi web menggunakan *web framework*.

Kata kunci: Model-View-Controller, Hierarchical, Framework, Codeigniter, BackendPro

¹ kawahyu@yahoo.com

PENDAHULUAN

Pengembangan sistem berbasis web berbeda dengan pengembangan perangkat lunak, sama halnya dengan rekayasa web yang memiliki perbedaan dengan rekayasa perangkat lunak. Meskipun demikian, rekayasa web ini masih melibatkan beberapa pemrograman, pengembangan perangkat lunak dan juga mengadopsi beberapa prinsip-prinsip rekayasa perangkat lunak (Simarmata, 2010). Melihat kenyataan tersebut, terlepas dari adanya perbedaan dalam perancangan, ada sumberdaya yang sama dalam rekayasa perangkat lunak dan rekayasa web yang penting, yaitu program atau *web developer*. *Web developer* sebagai pelaku utama dalam pengayaan produk Teknologi Informasi seperti aplikasi-aplikasi berbasis komputer dituntut untuk bekerja dalam tekanan sesuai penjadwalan yang telah ditetapkan tanpa berkurang performanya guna menghasilkan produk berbasis web yang berkualitas. Aplikasi-aplikasi yang sekarang ini banyak diminati adalah aplikasi-aplikasi bisnis yang dengan alasan tertentu harus dapat dijalankan di dalam sebuah jaringan komputer baik jaringan komputer lokal dan juga jaringan internet. Rekayasa web dibutuhkan untuk menghasilkan aplikasi-aplikasi berbasis web tersebut.

Kebanyakan sistem ini berbasis web perlu dikembangkan dalam waktu yang sangat singkat sehingga sulit untuk membuat dan menerapkan pada tingkat yang sama dengan perencanaan dan pengujian seperti yang digunakan di dalam pengembangan perangkat lunak (Simarmata, 2010). Diawal perkembangannya, saat melakukan rekayasa web yaitu pada tahapan *coding*, seorang programmer atau sering disebut sebagai *Web Developer*, di tuntut untuk bekerja dengan cepat membuat serangkaian prosedur, fungsi dan lain sebagainya agar dapat digunakan dalam suatu proyek rekayasa web. Permasalahan terjadi ketika *web developer* berpindah ke proyek lainnya, *web developer* tersebut melakukan serangkaian aktifitas yang sama atau dalam arti membuat dan memulai *coding* dari awal. Aktifitas yang banyak dilakukan adalah memanfaatkan proyek sebelumnya untuk digunakan di proyek yang

baru dengan penyesuaian tertentu sesuai kebutuhan proyek, misalnya pada kerangka sistem, sistem basis data, dan sistem keamanan. Jika ada kelemahan kode di sistem pada proyek sebelumnya yang tidak diketahui, maka kelemahan tersebut akan di turunkan ke proyek selanjutnya.

Dengan adanya perkembangan teknologi saat ini memberikan pengaruh terhadap proses rekayasa web. Perkembangan yang terjadi di sisi pemrograman adalah pola kerja membuat dan memulai *coding* dari awal ataupun pemanfaatan kode dari proyek sebelumnya telah bergeser menjadi pemanfaatan pustaka bersama, artinya untuk mengerjakan proyek aplikasi berbasis web cukup melakukan dan membuat sedikit *coding* dengan memanfaatkan pustaka bersama guna menghasilkan aplikasi web yang dinamis. Akan lebih mudah lagi apabila *web developer* bekerja dalam lingkungan sebuah *framework web* tertentu yang sekarang ini mulai banyak digunakan.

Saat ini ada kecenderungan bahwa seseorang pengembang membangun sesuatu diatas sesuatu atau dengan kata lain membangun aplikasi diatas *framework* tertentu, artinya pengembang IT sekarang ini lebih *object-oriented*, *reusable* dan *component-based* (Hidayat, 2010). Untuk mempercepat waktu pembuatan dan pengerjaan aplikasi berbasis web, telah banyak digunakan *framework-framework web*. Penggunaan *framework web* ini dalam sebuah proyek rekayasa web memberikan kemudahan bagi web developer dalam membangun sebuah proyek berbasis web karena penggunaan *framework* menawarkan kemudahan-kemudahan dalam pemanfaatan pustaka bersama. *Framework* tersebut memiliki arsitektur atau pola MVC atau *Model-View-Controller* yang memisahkan bagian logika dan presentasi. Pola MVC bagi tim pembuat atau pengembang dalam suatu proyek rekayasa web memudahkan seorang *web developer* untuk berkonsentrasi pada *core-system* sedangkan bagi *web designer* dapat berkonsentrasi di bagian *user-interface*. Selanjutnya pola HMVC atau *Hierarchical Model-View-Controller* telah banyak pula digunakan dalam rekayasa web dimana pola-

nya hampir sama dengan MVC tetapi terjadi pemisahan dimana setiap *web developer* dapat mengerjakan masing-masing modul tanpa mengganggu modul lain yang sedang dikerjakan sehingga secara keseluruhan program yang dibuat menjadi lebih modular.

Dari latar belakang tersebut diatas, dalam penelitian ini akan dibahas mengenai bagaimanakah penerapan pola HMVC dalam suatu sistem berbasis *web framework* guna menghasilkan program yang modular. Tujuan dari penelitian ini adalah untuk menjelaskan tahapan-tahapan ini membangun suatu aplikasi berbasis web dengan *framework* tertentu dan untuk menjelaskan keuntungan dalam mengembangkan aplikasi berbasis web dengan pola HMVC. Batasannya yaitu *framework* yang digunakan adalah *framework Codeigniter*, sedangkan untuk implementasinya dilakukan dengan mengembangkan *Panel Control* yaitu BackendPro sebuah *open-source* berlisensi bebas.

Rekayasa Web adalah sub-disiplin dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara, dan melibatkan aplikasi web. Rekayasa web mempunyai tujuan yang secara efektif mendukung aktifitas-aktifitas organisasi dengan aplikasi web. Dalam rekayasa web, MVC termasuk kedalam pendekatan-pendekatan pada pengembangan aplikasi web berupa *framework* (Simarmata, 2010).

MVC atau *Model-View-Controller* adalah sebuah pola pemrograman yang bertujuan memisahkan logika bisnis, logika data dan logika tampilan atau secara sederhana memisahkan antara proses, data dan tampilan (Wardana 2010). Pola MVC dimulai sejak era 70-an atas pemikiran Prof. Trgve Reenskaug yaitu seorang berkebangsaan Norwegia. Penggunaan pola ini diharapkan dapat meminimalisasi penulisan perintah, sehingga resiko terjadinya *bug* juga minimal, serta meningkatkan efisiensi pembangunan aplikasi. Pada awal penerapannya, konsep MVC digunakan Smaltalk. Seiring dengan popularitasnya yang kian menanjak, saat ini arsitektur MVC luas digunakan dalam dunia *framework*

pemrograman baik untuk aplikasi *desktop-based* maupun *web-based* (Pratama, 2010).

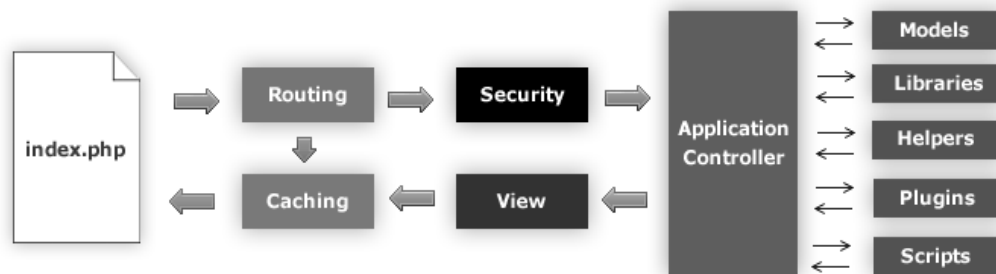
MVC terdiri dari 3 (tiga) bagian utama yaitu bagian *model* bagian *view* dan bagian *controller*. Bagian *model* adalah komponen MVC yang merepresentasikan data, mengatur respon terhadap permintaan, serta memberi hak akses untuk memanipulasi data yaitu pengambilan data dari basis data dan memasukkan data ke dalam database. Isi utama dari bagian ini berupa perintah SQL yang hasilnya dikirimkan ke bagian *controller*. Bagian *view* adalah komponen MVC yang bertugas mengatur bagaimana suatu data yang diperoleh dari *controller* ditampilkan untuk user dan mencakup semua proses yang terkait dengan *layout output*. Isi dari bagian ini bisa berupa form, tabel, gambar, animasi yang boleh dilihat oleh pengguna. Bagian *controller* adalah komponen MVC yang bertugas mengirim perintah ke bagian *model* untuk mendapatkan data yang diinginkan dan selanjutnya dikirimkan ke bagian *view* untuk ditampilkan. Bagian ini tidak mengetahui bagaimana data tersebut diambil dari database, karena bagian ini tidak berisi kode perintah SQL. Namun umumnya aplikasi yang dibangun dengan konsep MVC adalah aplikasi yang cukup besar sehingga penggunaan pola ini dapat mempercepat dan mempermudah proses pengembangan perangkat lunak.

HMVC atau *Hierarchical Model-View-Controller* adalah suatu pola MVC tetapi berupa hirarki dimana dalam implementasinya MVC tersimpan didalam modul-modul tertentu sehingga setiap modul memiliki *model*, *view*, dan *controller* sendiri.

Web framework dikenal dengan *web application framework* berupa sebuah *software framework* yang di desain untuk mendukung pengembangan web dinamis, aplikasi web dan *web service*. *Framework* adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lainnya sehingga dalam pembuatan aplikasi website harus mengikuti aturan dari *framework* tersebut (Wardana, 2010). *Framework* dapat di-

artikan sebagai koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal (Basuki, 2010). *Web framework* sangat cocok untuk pembangunan aplikasi berbasis web karena lebih fleksibel, *secure*, dan menggunakan pola MVC. Contoh dari *web framework* misalnya Akelos, Kohana, CakePHP, Symfony Project, Yii, Canvas, Zend, Zoop, Seaquill, Trax, Prado, Codeigniter dan lainnya.

ringan dan cepat, Codeigniter menggunakan konsep MVC sehingga untuk memungkinkan pemisahan antara bagian *application logic* dan *presentation*, Codeigniter tidak memerlukan *template engine* tapi menggunakan *template parser* sehingga *per-formance* sistem lebih baik, Codeigniter memiliki dokumentasi yang lengkap dan jelas sehingga mudah dipelajari dan memperjelas fungsi sebuah kode program, Codeigniter di dukung oleh komunitas sehingga ada kemudahan untuk *sharing* pengetahuan. Codeigniter hanya membutuhkan sedikit konfi-



Gambar 1. Alur Program Codeigniter (sumber: masdeka.web.id)

Framework Codeigniter adalah sebuah framework PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibandingkan jika menulis semua kode dari awal (Basuki, 2010). Codeigniter dikembangkan oleh Rick Ellis yang merupakan CEO dari Ellislab, Inc. Codeigniter sering digunakan oleh Perusahaan IT untuk pengembangan web atau web development karena framework ini memiliki beberapa kelebihan. Kelebihan dari Codeigniter adalah dapat diperoleh secara gratis karena berada dibawah Apache/BSD-style *open source license*. Sesuai dengan lisensinya dalam menggunakan Codeigniter diizinkan untuk memakai, meng-copy, melakukan modifikasi dan mendistribusikan *software*nya dan dokumentasinya dengan atau tanpa modifikasi, untuk segala macam tujuan sesuai dengan ketentuan yang ada (Noviyanto, 2010). Artinya Codeigniter dapat digunakan secara bebas. Selain itu, Codeigniter memiliki ukuran yang kecil dan bisa diatur agar sistem hanya *meload library* yang dibutuhkan saja sehingga sistem dapat berjalan

gurasu untuk setting standar dan juga memiliki keleluasaan melakukan konfigurasi untuk kepentingan dari *routing*. Codeigniter menghasilkan URL yang bersih sehingga untuk meningkatkan *web accessibility*. Codeigniter memudahkan koneksi web aplikasi dengan database, karena Codeigniter telah menyediakan fungsi dan class untuk melakukannya. Perintah yang digunakan juga sangat mudah diingat, karena mirip dengan perintah koneksi database PHP biasa (Wiswakarma, 2010).

Alur program menggunakan pola MVC dalam *web framework* Codeigniter digambarkan oleh Gambar 1. Penjelasan dari gambar tersebut adalah sebagai berikut: File *index.php* berfungsi sebagai *controller* di bagian depan saat pertama kali sistem dipanggil melalui *HTTP request*, selanjutnya menginisialisasi *basic resource* yang dibutuhkan untuk menjalankan Codeignier. *Router* menganalisa *HTTP request* untuk menentukan apa yang harus dilakukan dengan *HTTP request* itu. Jika file *cache* masih ada untuk *request* yang sama, maka *response* akan dikirim langsung ke browser

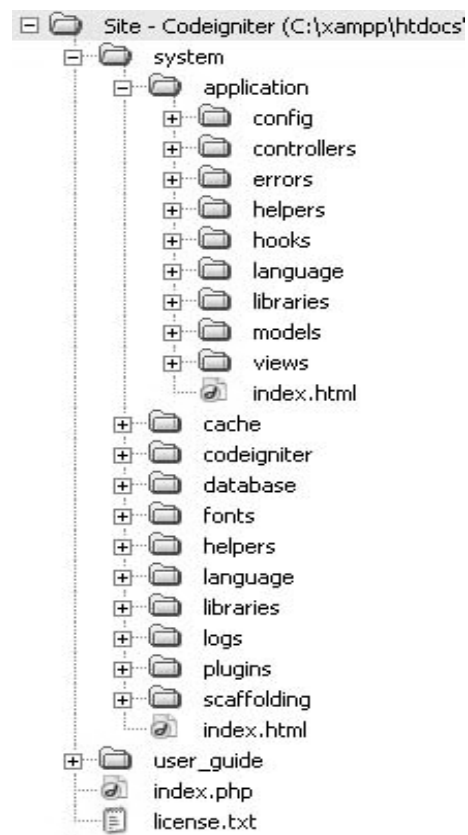
tanpa melewati eksekusi normal sistem. Pada bagian *Security*, sebelum *controller* aplikasi di panggil, *HTTP request* dan data yang dikirim user terlebih dahulu difilter untuk alasan keamanan. Pada bagian *controller* terjadi pemanggilan model, *libraries*, *helper*, *plugin*, dan resource lain yang di butuhkan untuk memproses *request* tersebut. Selanjutnya *view* yang sudah diproses, dikirim ke browser sebagai *layout output*. Jika status *caching* aktif, *view* akan disimpan di *cache* dengan maksud apabila ada *request* yang sama, *view* itu bisa langsung ditampilkan lagi.

BackendPro adalah panel kontrol untuk *web developer* yang ditulis dalam PHP untuk *framework* Codeigniter. Proyek *opensource* yang dikembangkan oleh Adam Price dengan lisensi bebas dari www.kaydoo.co.uk ini tidak hanya memberikan dasar yang solid baik untuk sebuah website dengan *auth library*, tetapi juga mencakup banyak fitur yang dibutuhkan untuk membantu pengembangan aplikasi *backend* berbasis web dengan cepat. BackendPro bukan CMS atau *Content Management System*. CMS adalah aplikasi yang bersifat *instant* dan *templating*, CMS memberikan sistem yang bekerja dengan penuh tetapi tidak memberikan bagian-bagian dari sistem untuk dikembangkan artinya CMS tidak cocok untuk membangun aplikasi web. Sedangkan BackendPro menawarkan banyak fitur CMS tanpa mengkonstruksi bagian *frontend* dan memberikan kebebasan mengembangkan sistem secara penuh. BackendPro menyediakan fungsionalitas untuk melakukan semua tugas-tugas sederhana yang biasa dikerjakan berulang-ulang dalam setiap proyek rekayasa web seperti *authentication*, *permissions*, dan *panel control*. Kelebihan lain yang menjadi alasan dalam penggunaan BackendPro adalah adanya fasilitas *user authentication*, *access control lists*, kemudahan membuat pelacakan *breadcrumb*, memungkinkan dilakukan *management asset*, adanya fasilitas konversi variable PHP ke Javascript, *site wide preferences*, adanya fasilitas *built in* untuk pengaturan anggota, memiliki *system log management*, dan adanya fitur pemeliharaan website. BackendPro

dapat digunakan sebagai dasar dalam pembangunan aplikasi berbasis web.

PEMBAHASAN

Tahapan didalam melakukan instalasi *web framework* Codeigniter paket standar yaitu hasil ekstraksi dari file *zip* yang di unduh dari situs resmi Codeigniter yaitu www.codeigniter.com dapat langsung diletakkan atau di unggah dalam direktori publikasi pada server, antara lain misalnya *xampp/htdocs* atau *var/www*.



Gambar 2. Struktur Direktori Codeigniter

Untuk konfigurasi dilakukan pada file-file *config.php*, *autoload.php*, *routes.php*, dan *database.php*. File *config.php* berisi konfigurasi dasar dari aplikasi yang akan dibangun misalnya konfigurasi pada fungsi *base_url* untuk menyesuaikan dengan alamat yang akan digunakan untuk *HTTP request*. File *autoload.php* digunakan untuk menspesifikasi resource apa saja yang akan di load oleh Codeigniter. File *routes.php* digunakan

untuk melakukan *remap URI* request agar mengarah ke suatu fungsi tertentu pada sebuah *controller*. File database.php berisi semua informasi yang digunakan untuk berkomunikasi dengan database. Struktur direktori dari Codeigniter diperlihatkan oleh Gambar 2. Dari struktur direktori Codeigniter tersebut, bagian terpenting untuk rekayasa web ada di direktori *application* yaitu: */system/application/controllers*, */system/application/models*, */system/application/views*. Misalkan disini akan dibuat suatu aplikasi Berita sederhana dengan anggapan *database* telah dibuat dan siap digunakan, maka langkah yang perlu dilakukan adalah membuat file *controller* dengan nama *berita.php* dan diletakkan di */system/application/controllers*.

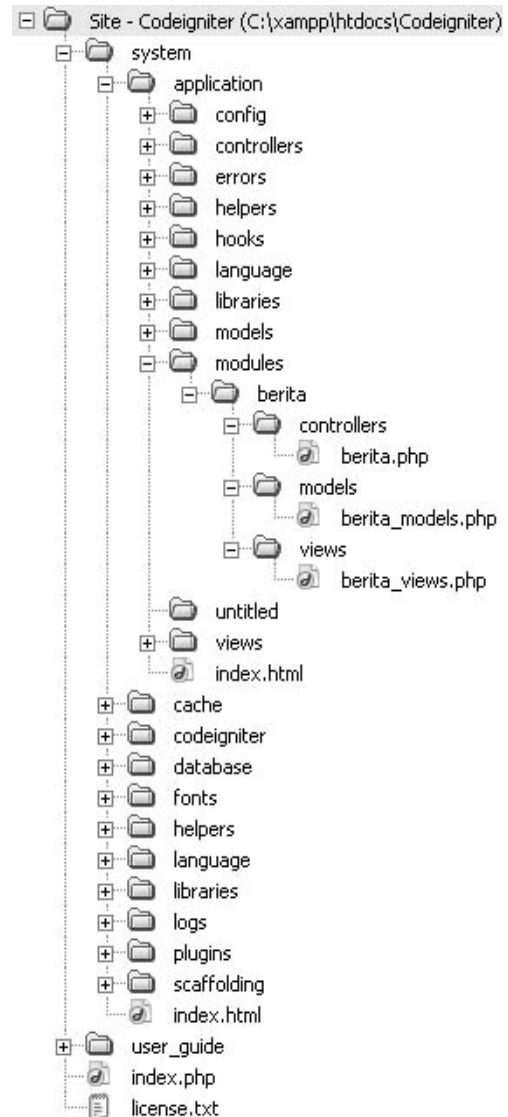
```
<?
class Berita extends Controller
{
function Berita ()
{
parent::Controller();
}
function index()
{
$this->load-
>view('berita_view')
}
//dan seterusnya
}
?>
```

Setelah itu membuat file model dengan nama *berita_model.php* dan diletakkan di direktori */system/application/models*.

```
<?
class Berita_model extends
Model {
function Berita_model ()
{
parent::Model();
}
//dan seterusnya
}
?>
```

Selanjutnya membuat file *view* yaitu *berita_view.php* yang diletakkan di dalam direktori */system/application/views*, sehingga hasil akhirnya adalah: */application/controller/berita.php*, */application/model/berita_model.php*, */application/view/berita_view.php*. Dengan cara seperti yang telah dijelaskan diatas, maka program yang

dibangun hanya berjalan pada *web framework* dengan pola MVC saja.



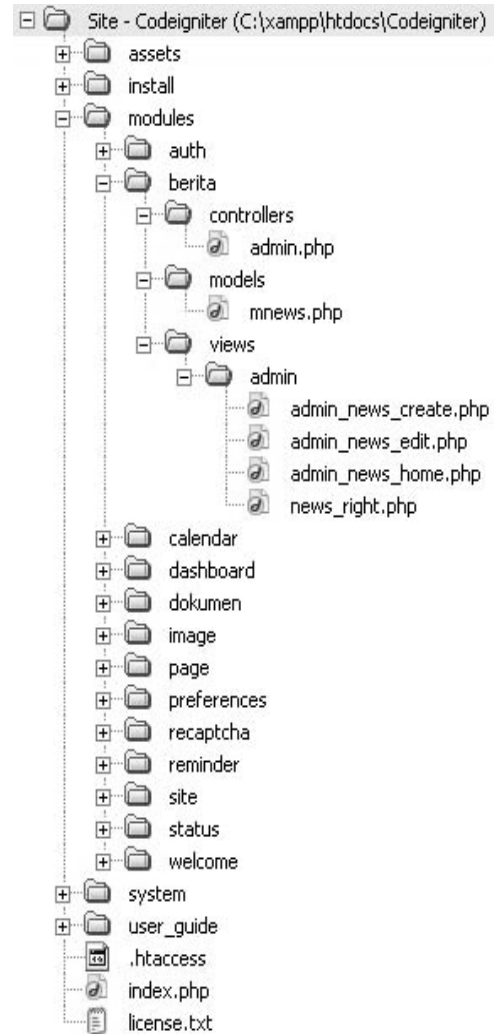
Gambar 3. Struktur Direktori Codeigniter dengan Modular Extension

Tahapan dalam melakukan instalasi *web framework* Codeigniter dengan pola *Hierarchical MVC* atau *HMVC* dilakukan dengan terlebih dahulu menyiapkan file *modular extensions* yang bisa di unduh di situs resmi Codeigniter. Hasil ekstraksi terdapat 3 buah file yaitu *MY_Router.php*, *Modules.php*, dan *Controlle.php* yang disimpan ke dalam direktori */system/application/libraries*. Selanjutnya membuat direktori baru dengan

nama *modules* di dalam direktori */system/application*, sehingga terdapat direktori baru dengan *path: /system/application/modules*. Di dalam *path* inilah kegiatan pengembangan dilakukan dimana setiap materi di simpan kedalam direktori-direktori. Misalnya untuk aplikasi Berita yang telah dibuat akan dikembangkan kedalam bentuk modular, maka didalam direktori */system/application/modules* dibuat direktori baru bernama *berita* dan di dalam direktori *berita* ditambahkan tiga subdirektori baru yaitu direktori *controllers*, direktori *models*, dan direktori *views* sehingga struktur direktori Codeigniter menjadi seperti Gambar 3, dimana masing-masing direktori tersebut untuk menampung file *berita.php*, *berita_models.php*, *ini berita_views.php*. Dengan menambahkan *modular extension* dalam web framework menjadikan Codeigniter lebih modular dan bekerja dalam pola *Hierarchical MVC* sehingga apabila akan dibuat aplikasi baru cukup dengan membuat direktori baru di dalam direktori *modules*. Dengan kedua tahapan tersebut yaitu instalasi Codeigniter dengan modular extension dan tanpa *modular extension*, pekerjaan seorang *web developer* tetap harus dilakukan dari awal untuk membangun *authentication*, *permissions*, dan *panel control*.

Tahapan untuk instalasi web framework Codeigniter dengan menggunakan BackendPro dilakukan dengan cara mengunduh file BackendPro pada situs www.kaydoo.co.uk, meng-ekstraksi file terkompresi, lalu me-nempatkan file kedalam *root server* dan selanjutnya melakukan pemanggilan melalui *browser web* untuk proses instalasi *database*. Setelah proses instalasi selesai, maka kita bisa langsung membangun modul-modul sesuai kebutuhan tanpa perlu memikirkan pembuatan bagian *authentication*, *permissions*, dan *panel control* karena bagian tersebut sudah disediakan oleh BackendPro. Ada sedikit perbedaan pada struktur direktori BackendPro, yaitu di root terdapat tiga direktori tambahan yaitu direktori *assets*, *install*, dan *modules*. Seperti yang diperlihatkan oleh Gambar 4. Jika sebelumnya untuk membuat modul baru dilakukan didalam direktori *system/application/modules*, maka untuk

BackendPro modul baru diletakkan di dalam direktori */modules/*.



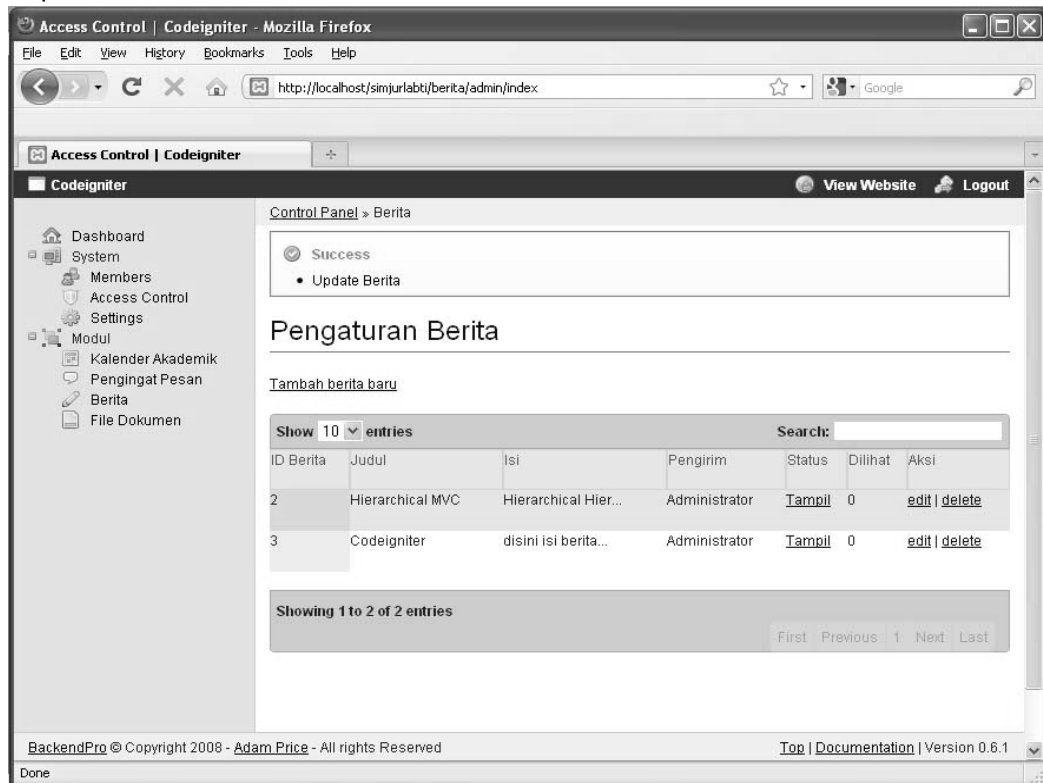
Gambar 4. Struktur Deriktori Backend Pro

Misalnya untuk membuat aplikasi Berita dalam bentuk modul yang lengkap menggunakan BackendPro sehingga modul tersebut memiliki fasilitas CRUD, maka langkah-langkahnya adalah membuat controller modul di */system/application/libraries/ MY_Controller.php*, lalu membuat direktori *berita* didalam direktori *modules* beserta subdirektori *controllers*, subdirektori *models*, dan subdirektori *views* dalam direktori *modules/berita/*. Di subdirektori *controllers* pada *modules/berita/controllers* ada file *admin.php* yang digunakan sebagai controller yang

mengacu ke controller modul di `/system/application/libraries/MY_Controller.php`. Dalam direktori `modules/berita/models` terdapat file `mnews.php` yaitu file model untuk pengolahan *database*.

Dalam direktori `modules/berita/models` terdapat subdirektori `admin` dan di dalamnya terdapat file untuk mengatur tampilan modul berita. Setelah membuat

dan membangun modul berita, langkah selanjutnya adalah mengatur keberadaan dan hak akses menu modul berita di dalam direktori `/system/application/views/admin/menu_modul.php`. Langkah terakhir adalah mendaftarkan modul berita sebagai *resources* di aplikasi BackendPro-nya.



Gambar 5. Antarmuka Modul Berita dengan BackendPro

Setelah dilakukan pembuatan modul berita yang dibangun di atas BackendPro, langkah selanjutnya adalah dilakukan pengujian. Hasil pengujian menunjukkan bahwa modul berita dapat berjalan dan terintegrasi dengan baik dengan BackendPro. Tahapan selanjutnya adalah dengan cara yang sama seperti pembuatan modul berita, dilakukan pembuatan modul kalender akademik, modul pengingat pesan, dan modul file dokumen.

Hasil akhir dari pembuatan modul berita yang dilakukan pada tahapan-tahapan tersebut diatas setelah ditambahkan pembuatan modul kalender akademik, modul pengingat pesan, dan modul file dokumen dapat dilihat pada

Gambar 5 yaitu tampilan atau antarmuka *Control Panel* atau fasilitas *Back End* program pada *browser web* dengan memanfaatkan BackendPro sebagai landasan awal aplikasi sebagaimana yang telah dibahas sebelumnya. Dalam program tersebut, pada bagian kiri terdapat 3 menu utama yaitu, Menu *Dashboard*, Menu *System* dengan sub menu *members*, *access control* dan *setting*, dan Menu Modul. Seluruh modul yang telah dibangun sebelumnya disimpan pada bagian Menu Modul. Sedangkan bagian kanan adalah bagian yang akan menampilkan isi dari menu-menu tersebut. Pada gambar diatas, di bagian kanan adalah antarmuka *Control Panel* untuk menu berita yang diambil dari modul berita.

Control Panel tersebut sudah memiliki fasilitas pengolahan basis data atau CRUD (*Create, Read, Update, Delete*).

Dari tahapan-tahapan seperti yang telah disebutkan diatas, maka akan terlihat bahwa tahapan-tahapan untuk membangun modul baru dengan menggunakan BackendPro lebih kompleks dibandingkan dengan membuat modul baru dalam Codeigniter yang ditambahkan *modular extension*. Tahapan pembangunan aplikasi berbasis web dengan menggunakan Codeigniter ditambah *modular extension* lebih sederhana, tetapi masih menyisakan pekerjaan bagi developer karena bagian sistem yang terpenting untuk aplikasi berbasis web seperti *authentication, permissions*, dan *panel control* harus dibangun dari awal. Pemanfaatan dan penggunaan BackendPro yang berlisensi bebas dalam pembangunan aplikasi berbasis web lebih ditujukan untuk pembangunan aplikasi berbasis web dengan waktu singkat, tanpa perlu memikirkan bagian *authentication, permissions*, dan *panel control*. Bagian-bagian tersebut sudah disediakan oleh BackendPro, sehingga developer dapat berkonsentrasi pada bagian-bagian lain yang lebih penting seperti bagian *Front End* jika diperlukan.

KESIMPULAN

Dari hasil analisa dapat disimpulkan beberapa hal yaitu penerapan pola *Hierachical MVC* dalam suatu sistem berbasis *web framework* menghasilkan program yang modular karena terjadi pemisahan antara bagian *logic application* dan bagian *presentation* kedalam bagian-bagian Model, View dan Controller. Dalam implementasinya terhadap *web framework* Codeigniter, untuk membangun satu aplikasi tertentu dikerjakan berupa modul dimana setiap modul memiliki subdirektori *controllers*, subdirektori *models*, dan subdirektori *views*. Masing-masing subdirektori tersebut digunakan untuk menampung file controllers, file views, dan file models. Tahapan-tahapan dalam membangun suatu aplikasi berbasis web dengan *web framework* memiliki tingkat kompleksitas yang berbeda. Dilihat dari sisi pemrograman, menggunakan *web framework*

Codeigniter yang berorientasi *Object Oriented Programming (OOP)* memiliki keunggulan tersendiri, keunggulan tersebut adalah *maintainability* yaitu mudah di kelola, *extensibility* yaitu memiliki kemampuan dapat diperluas tanpa mengganggu modul-modul yang telah dikerjakan sebelumnya, dan *reusability* yaitu memiliki kemampuan untuk dapat digunakan kembali dimana setiap objek dapat digunakan oleh *project* yang lain tanpa penyesuaian yang berarti. Selain itu web framework Codeigniter bekerja dalam pola MVC, sehingga tidak ada kesulitan untuk penyesuaian ke pola berbentuk *Hierarchical*. Karena Codeigniter mendukung konsep OOP maka dalam hal ini *web developer* dituntut untuk memiliki pengetahuan dan memahami OOP secara penuh.

Pemisahan kode program yang mengikuti konsep MVC menjadi lebih mudah di *maintenance*, mudah dikembangkan dan mempercepat proses rekayasa web atau pengembangan web sehingga *developer* dapat lebih fokus pada fitur yang dibutuhkan dengan membuat kode program seminimal mungkin.

Penggunaan pola *Hierarchical MVC* membuat program menjadi lebih modular, artinya *developer* dapat mengerjakan masing-masing modul tanpa mengganggu direktori yang sedang dikerjakan oleh *developer* lain. Saat penggabungan modul-modul menjadi lebih mudah karena tinggal mengkopi modul di master program sehingga terjadinya konflik file dapat dihindari.

Menggunakan BackendPro untuk membangun sebuah aplikasi berbasis *framework web* membantu mempercepat proses *coding* karena *web developer* dapat berkonsentrasi penuh pada aplikasi yang akan di kembangkan, bukan pada bit-bit untuk mengelola sistem.

DAFTAR PUSTAKA

- Basuki, A.P., 2010, *Membangun Web Berbasis PHP dengan Framework Codeigniter*, Lokomedia, Yogyakarta.
- Hidayat, E.W., 2010, *Pengembangan Sistem Informasi Laboratorium Teknik Informatika (SILABTI) Dengan Arsitektur Model View*

- Controller*, Jurnal Sitrotika vol. 6 no. 2, Juli 2010, 118-123.
- Noviyanto, A., 2010, *Burn Your PHP Code Using Codeigniter*, Skripta Media Creative, Yogyakarta.
- Pratama, A.N.W., 2010, *Codeigniter: Cara Mudah Membangun Aplikasi PHP*, Mediakita, Jakarta.
- Simarmata, J., 2010, *Rekayasa Web*, Penerbit Andi, Yogyakarta.
- Wardana., 2010, *Menjadi Master PHP dengan Framework Codeigniter*, PT.Elex Media Komputindo, Jakarta.
- Wiswakarma, K., 2010, *9 Langkah Menjadi Master Framework Codeigniter*, Lokomedia, Yogyakarta.