

## **Penggunaan kerangka kerja CodeIgniter sebagai platform utama dalam pembangunan sistem aplikasi MARDI**

(The use of CodeIgniter framework as the main platform in the development of the MARDI application system)

Mohamad Zulkifly Zakaria @ Mustafa dan Mohd Fauzi Mohd Nazeri

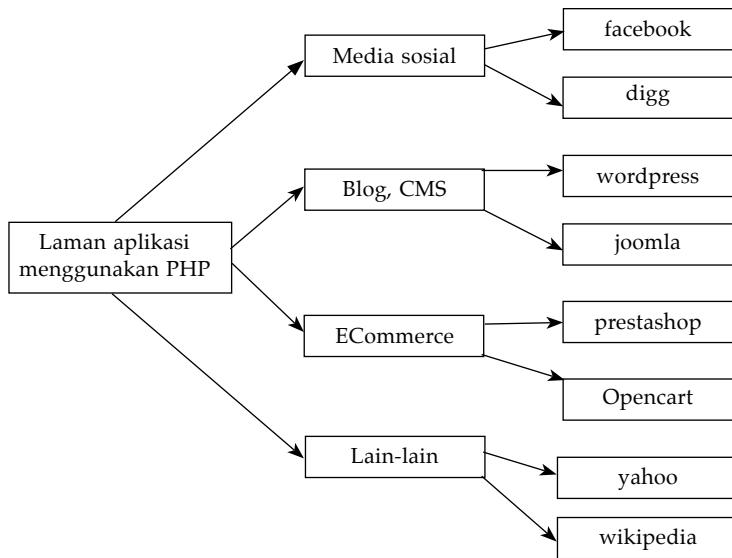
### **Pengenalan**

Pusat Pengurusan ICT MARDI adalah merupakan entiti yang dipertanggungjawabkan untuk memberi perkhidmatan berkaitan ICT. Salah satu cabang perkhidmatan yang diberikan adalah membangunkan sistem aplikasi bagi menyokong aktiviti operasi dan pengurusan teknologi MARDI. Sistem aplikasi yang dibangunkan bukan sahaja untuk menyokong perkhidmatan dalaman, tetapi juga untuk menyokong kehendak pemegang taruh MARDI khususnya petani dan usahawan tani. Artikel ini ditulis bagi memberi gambaran teknikal berkaitan penggunaan kerangka kerja CodeIgniter (CI) sebagai platform utama dalam pembangunan sistem aplikasi MARDI.

Terdapat pelbagai bahasa pengaturcaraan yang boleh digunakan untuk membangunkan sesebuah sistem aplikasi berasaskan *web* seperti JAVA, ASP, Ruby, PHP dan lain-lain. Di MARDI, bahasa pengaturcaraan utama yang digunakan bagi pembangunan sistem aplikasi berasaskan *web* ialah PHP yang telah digunakan bermula dari tahun 2007. PHP merupakan singkatan daripada perkataan *Personal Home Page*, namun kini maksud rasmi PHP ialah akronim rekursif bagi *PHP Hypertext Preprocessor*. PHP merupakan bahasa pengaturcaraan sumber terbuka bagi *server-side scripting* yang boleh digunakan bersama kod HTML, CSS dan Javascript bagi menghasilkan sistem aplikasi berasaskan *web* yang bersifat dinamik. Sebanyak 78.1% laman *web* yang wujud ketika ini menggunakan PHP sebagai bahasa pengaturcaraan utama laman *web*. Dengan berkembangnya komuniti sumber terbuka, PHP menjadi bahasa pengaturcaraan paling popular dalam membangunkan aplikasi laman *web* dan telah digunakan sebagai salah satu bahasa pengaturcaraan dalam membangunkan kerangka kerja (Rajah 1). Terdapat pelbagai kerangka kerja berasaskan PHP yang telah dibangunkan antaranya ialah Yii, Laravel, Symfony, CakePHP, CodeIgniter (CI) dan lain-lain.

### **Kelebihan kerangka kerja**

Kerangka kerja merupakan sekumpulan fungsi/prosedur/*class* yang telah tersedia dan disusun secara berstruktur untuk digunakan oleh pembangun sistem. Kerangka kerja dapat membantu pembangun sistem mempercepatkan proses



Rajah 1. Laman web yang dibangunkan menggunakan pengaturcaraan PHP

pembangunan dan pengembangan sesebuah sistem aplikasi tanpa perlu membangunkan fungsi atau *class* tersebut. Kerangka kerja juga kebiasaannya telah tersedia dengan komponen-komponen umum yang biasa digunakan seperti *validasi*, *ORM*, *pagination*, *database*, *scaffolding*, penentuan sesi, muat naik fail, penghantaran e-mel dan *error handling*. Kelebihan lain penggunaan kerangka kerja adalah mempunyai fungsi keselamatan yang dapat melindungi sistem daripada serangan dan juga memudahkan proses pengujian. Kesemua komponen ini sangat membantu dalam menghasilkan sistem aplikasi yang lebih bermutu serta cekap dan paling penting sekali kerangka kerja lebih dinamik dalam aktiviti penambahbaikan dan pengembangan sistem berbanding dengan *Content Management System* (CMS).

### Penggunaan PHP di MARDI

Sebelum tahun 2015, tradisional PHP (*plain PHP*) digunakan dalam pembangunan sistem aplikasi berasaskan *web* di MARDI tanpa menggunakan sebarang kerangka kerja. Oleh yang demikian, telah berlaku beberapa kejadian pencerobohan dan pelanggaran sistem keselamatan ke atas sistem aplikasi MARDI. Contohnya pada tahun 2016, melalui ujian penembusan didapati sistem mudah untuk dicerobohi oleh pihak tidak bertanggungjawab dan memberi masalah kepada fungsi sistem. Manakala pada tahun 2019 telah berlaku kes percubaan menceroboh data oleh pihak yang tidak bertanggungjawab yang mana sistem tidak dapat melindungi data-data yang terdapat dalam pangkalan data daripada diceroboh. Hasil semakan, mendapati ia berpunca daripada kelemahan dalam pengaturcaraan serta platform yang digunakan. Penggunaan tradisional PHP tanpa penggunaan kerangka kerja juga menyebabkan kesukaran kepada pembangun sistem MARDI untuk menaik taraf sistem aplikasi sedia ada. Banyak lagi isu-

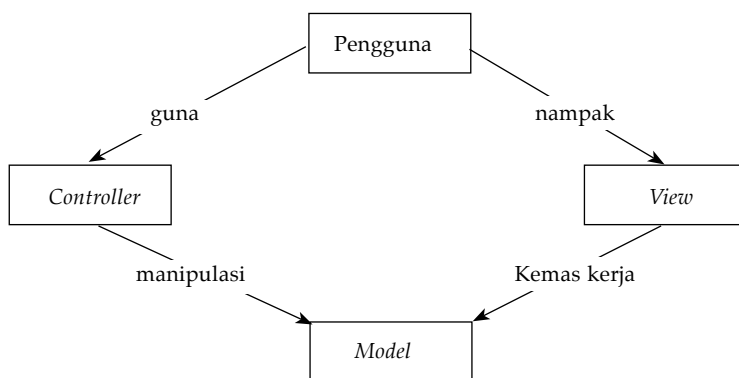
isu lain yang telah muncul seperti isu ketidakseragaman dalam pembangunan serta struktur pengaturcaraan yang tidak teratur dan membantutkan kecekapan pembangunan sistem di MARDI.

### CodeIgniter (CI)

CodeIgniter (CI) adalah kerangka kerja yang bersifat sumber terbuka yang dibangunkan menggunakan konsep *Model-View-Controller* (MVC CI) berupaya memendekkan masa pembangunan sistem, membantu mengurus kod atur cara dan menyusun struktur kod fail atur cara agar lebih tersusun dan seragam. Kod atur cara yang perlu ditulis adalah lebih ringkas, mudah diselenggara dan mudah membuat peningkatan aplikasi. CI menggunakan reka bentuk seni bina MVC yang menjadi amalan standard industri apabila membangunkan aplikasi *web*. MVC merupakan sebuah konsep dalam pembangunan sistem yang memisahkan kod atur cara kepada tiga komponen iaitu data (*model*), paparan (*view*) dan logik (*controller*) untuk mereka bentuk antara muka pengguna aplikasi *web* (Rajah 2).

Di samping itu, antara kelebihan lain yang terdapat pada CI ialah ia mempunyai rujukan dokumentasi yang lengkap, mempunyai saiz kerangka kerja yang kecil untuk dimuat turun, mendukung pelbagai fungsi umum bagi membantu memudahkan pembangunan seperti fungsi *Query Builder*, *Form Validation*, *Session Management*, *Cross Site Scripting*, *File Uploading* dan lain-lain. Ia juga mudah untuk dikonfigurasi, dapat berfungsi dengan baik di pelbagai komputer pelayan, mendukung pelbagai koneksi *database* yang umum serta mudah diintegrasikan dan dihubungkan kepada *third party library*.

Berdasarkan kepada kelebihan tersebut, serta keperluan pembangunan sistem di MARDI, CI telah dipilih untuk digunakan sebagai kerangka kerja dalam pembangunan sistem. Berikut diterangkan secara ringkas beberapa dapatan yang dapat diperoleh sepanjang penggunaan CI dalam pembangunan sistem di MARDI.



Rajah 2. Reka bentuk Model-View-Controller (MVC)

### Atur cara yang lebih ringkas

Kod pengaturcaraan yang lebih ringkas dan mudah ditulis sudah pasti akan menjimatkan masa dalam pembangunan, mengurangkan kesilapan dalam penulisan kod dan mengurangkan ralat. Contohnya kod bagi pertanyaan pangkalan data MySQL (*db query*), kebiasaannya dalam sintak yang normal ia ditulis seperti berikut:

```
$connection = mysql_connect("localhost","zulfly","pwd123");
mysql_select_db("websites",$connection);
$result = mysql_query("SELECT * FROM sites",$connection);
while ($row = mysql_fetch_array($result, MYSQL_NUM))
{
    foreach ($row as $attribute)
        print "{$attribute[1]}";
}
```

Sementara kod atur cara yang ditulis menggunakan rangka kerja CI adalah seperti yang berikut:

```
$this->load->database('websites');
$query = $this->db->get('sites');
foreach ($query->result() as $row)
{
    print $row->url;
}
```

Perbezaan ketara yang dapat dilihat daripada pengaturcaraan di atas adalah penulisan kod yang lebih ringkas dan mudah untuk difahami. Jelas sekali penulisan kod atur cara menggunakan CI lebih pendek dan ringkas untuk ditulis berbanding daripada sintak biasa.

### Kemas kini pautan secara automatik

Dalam pembangunan sebuah sistem aplikasi, di laman utama kebiasaannya akan mempunyai banyak menu dan submenu yang akan menghubungkan kepada laman lain. Kod atur cara dalam sintak tradisional yang biasa digunakan adalah seperti yang berikut:

```
<a href="http://www.mysite.com/index.php/start/hello/zulfly"
">Selamat datang ke laman Zulfly</a>
```

Masalah bagi kaedah di atas adalah apabila kita ingin pindahkan laman tersebut kepada URL atau *Domain Name System* (DNS) yang lain. Semakan satu demi satu URL yang sedia ada perlu dibuat dan mengubahnya kepada URL yang terkini. Ia akan meremehkan aktiviti pengemaskinian dan penyelenggaraan.

Manakala dalam CI akan membantu pembangun sistem untuk tidak perlu menyemak baris demi baris URL yang ada dan mengubahnya satu demi satu. Fungsi dan atur caranya adalah seperti yang berikut:

```
echo anchor('start/hello/Zulfly', 'Selamat datang ke laman Zulfly');
```

Fungsi *anchor* dalam CI akan merujuk kepada fail konfigurasi bagi mendapatkan *Base* URL. Kaedah yang digunakan dalam kerangka kerja CI adalah dengan meletakkan URL dalam satu fail konfigurasi yang boleh dikongsi dan dicapai oleh fail-fail lain. Ini akan membantu pembangun sistem untuk tidak perlu menulis satu demi satu jika ada perubahan pada *Base* URL. Hanya perlu menukar sekali sahaja pada fail konfigurasi dan semua fail lain yang berkaitan akan secara automatik turut berubah.

### **Mencegah serangan *SQL injection***

Kemasukan data maklumat ke dalam pangkalan data adalah bahagian yang sangat kritikal dan sering menghadapi masalah. Ia mudah terdedah kepada serangan pihak luar seperti serangan *SQL injection* yang digunakan penceroboh untuk melihat dan mencuri data lantas dapat mengubah atau menghapus data dalam pangkalan data aplikasi. Ia boleh terjadi sekiranya atur cara yang ditulis tidak melakukan *filter* terhadap kemasukan data oleh pengguna yang mengakibatkan manipulasi boleh dilakukan pada arahan *SQL*. Contoh atur cara yang terdedah terhadap serangan *SQL injection* diserang adalah seperti yang berikut:

Katakan kita ingin membuat *SQL query* seperti berikut;

```
SELECT id, name FROM users WHERE user = '{$user}' AND password = '{$password}';
```

Nilai bagi pemboleh ubah yang digunakan di atas ialah:

```
$user = "Zulfly";  
$password = "pword123";
```

Apabila nilai dimasukkan ke dalam *SQL query*, maka ia akan menjadi seperti yang berikut:

```
SELECT id, name FROM users WHERE user = 'Zulfly' AND password = 'pword123';
```

Jika dilihat kepada atur cara di atas, didapati tiada sebarang masalah. Atur cara akan berfungsi dengan baik. Namun demikian, sekiranya nilai yang diberikan diubah kepada nilai seperti di bawah, maka atur cara *SQL query* juga akan berubah. Mari lihat contoh atur cara tersebut.

Nilai pemboleh ubah baharu:

```
$user = "Zulfly";  
$password = "pword123' OR '1' = '1";
```

Maka pernyataan *SQL query* akan berubah kepada seperti yang berikut:

```
SELECT id, name FROM users WHERE user = 'Zulfly' AND password = 'pword123'  
OR '1' = '1';
```

Dapat dilihat daripada pernyataan di atas, klausa WHERE telah mempunyai nilai atau keadaan sentiasa TRUE. Penggodam boleh memasukkan karakter tertentu seperti " ' " untuk menjadikan *SQL query* yang kita lakukan memberi hasil yang tidak diinginkan kepada pengguna. Keadaan yang lebih buruk juga boleh berlaku di mana penggodam boleh menghapus jadual data yang terdapat dalam sistem aplikasi. Masalah ini kebiasaannya bukan dalam bentuk *SQL injection* sahaja, tetapi dalam keadaan tidak menyediakan/ menghantar data dengan cara yang betul juga boleh menyebabkan pengguna diberi data yang salah. Contohnya:

```
$user = "Zulfly";  
$password = "L09in'gm";
```

Jika dilihat katalaluan yang digunakan seolah-olah ia sangat selamat, tetapi sebenarnya ia akan memberi masalah dalam *query* yang digunakan. Sistem akan memberi maklum balas ralat apabila pernyataan di bawah dilaksanakan.

```
SELECT id, name FROM users WHERE user = 'Zulfly' AND password =  
'L09in'gm';
```

Kaedah untuk mengatasi ralat ini adalah dengan melaksanakan penyediaan data atau dikenali sebagai "prep" dalam borang kemasukan data, sebelum data tersebut dihantar ke dalam pangkalan data. Aktiviti ini memerlukan pembangun sistem menulis atur cara yang lebih daripada kebiasaannya.

Tetapi dalam rangka kerja CI, isu berkenaan dapat diatasi dengan lebih mudah dan menjimatkan masa. Apabila kita membangunkan kotak input dengan menulis atur cara seperti yang berikut:

```
echo form_input('user', 'Zulfly');
```

Secara tidak langsung CI akan turut menyediakan satu set kod tersembunyi secara automatik yang akan dapat membantu kita menguruskan karakter khas seperti " ' " atau "&" yang boleh menghasilkan kekeliruan dan ralat semasa data dihantar.

```

function form_prep($str = '')
{
    // ulang sekiranya nama field adalah array

    if (is_array($str))
    {
        foreach ($str as $key => $val)
        {
            $str[$key] = form_prep($val);
        }
        return $str;
    }
    if ($str === '')
    {
        return '';
    }
    $temp = '__TEMP_AMPERSANDS__';
    // Gantikan entiti kepada penanda sementara
    // htmlspecialchars tak akan mengganggu dan memberi hasil error
    $str = preg_replace("/&#(\d+);/", "$temp\\1;", $str);
    $str = preg_replace("/&(\w+);/", "$temp\\1;", $str);
    $str = htmlspecialchars($str);
    // Tambahan htmlspecialchars.
    $str = str_replace(array("'", '"'), array("&#39;", "&quot;"),
    $str);
    // Decode penanda sementara tadi kembali kepada entiti
    $str = preg_replace("/$temp(\d+);/", "&#\\1;", $str);
    $str = preg_replace("/$temp(\w+);/", "&\\1;", $str);
    return $str;
}

```

### Melindungi daripada *Cross Site Scripting* (XSS)

XSS adalah merupakan kelemahan dalam penulisan kod atur cara yang memberi ruang aturcara tertentu dilaksanakan dalam sistem aplikasi bagi tujuan memancing dan mencuri data serta tujuan lain yang tidak baik. Oleh yang demikian, CI telah menyediakan fungsi tersedia yang mampu untuk membantu mengelak daripada serangan XSS berlaku iaitu dengan cara mengkonfigurasi tetapan pada XSS *global filter* kepada TRUE bagi tujuan pengesahan.

```
$data = $this->input->xss_clean($data);
```

Sekiranya data yang digunakan adalah dalam format bergambar, kod berikut boleh digunakan

```
$this->input->xss_clean($file, TRUE);
```

Kod di atas akan memberitahu CI, fail yang digunakan memerlukan pengesahan terlebih dahulu.

### Status penggunaan CI di MARDI

Berdasarkan kepada perkembangan teknologi dan isu pencerobohan yang pernah berlaku di MARDI dan juga di tempat lain, pada tahun 2015 Pusat Pengurusan ICT pada ketika itu telah membuat keputusan untuk menggunakan kerangka kerja CI dalam setiap pembangunan sistem aplikasi baharu MARDI. Keputusan ini dibuat selepas melakukan kajian kepada kelebihan

yang ada pada kerangka kerja CI dan juga keupayaan tenaga kerja serta kebolehpayaan mengadaptasi pengetahuan dan kemahiran sedia ada kepada kaedah pembangunan menggunakan kerangka kerja CI. Pada tahun 2020, Pusat Pengurusan ICT MARDI telah berjaya menambah baik Sistem Pemantauan Rancangan Operasi Korporat MARDI (PROPER) menggunakan kerangka kerja CI. Terkini, Pusat Pengurusan ICT juga sedang bekerjasama dengan Pusat Pengurusan Sumber Manusia dalam menambah baik Sistem Pengurusan Sumber Manusia (INTEAM) menggunakan kerangka kerja CI. Penambahbaikan ini penting dilaksanakan segera kerana INTEAM merupakan sistem utama yang menyimpan semua maklumat sumber manusia MARDI yang berisiko tinggi untuk mengalami kegagalan dan menghadapi isu-isu keselamatan data andai kata tidak ditambah baik kepada penggunaan kerangka kerja yang lebih cekap dan selamat. Dalam perancangan, sistem yang turut akan ditambah baik ialah sistem e-Jawatan MARDI yang dijadualkan akan melalui proses penambahbaikan pada tahun 2022.

### **Bibliografi**

- Malaysian Agriculture Research and Development Institute (2009).  
Sistem e-Jawatan. Diperoleh dari <http://ejawatan.mardi.gov.my>.
- Malaysian Agriculture Research and Development Institute (2020).  
Sistem PROPER MARDI. Diperoleh dari <http://proper.mardi.gov.my>
- Hustinawati, Albert, Latifah (2014). Performance Analysis Framework Codeigniter and CakePHP in Website Creation. *International Journal of Computer Applications* (0975 – 8887)
- Blanco, Jose A, dan David Upton. Codeigniter 1.7: Improve Your Php Coding Productivity with the Free Compact Open-Source MVC Codeigniter Framework 2009.
- SocialCompare. PHP frameworks comparison, 2021. Diperoleh dari <http://socialcompare.com/en/comparison/php-frameworkscomparison>. W3Techs. Usage of server-side programming languages. Diperoleh dari [http://w3techs.com/technologies/overview/programming\\_language/](http://w3techs.com/technologies/overview/programming_language/).
- Samra J. Comparing Performance of Plain PHP and Four of Its Popular Frameworks. Thesis Project. Linnaeus University, Sweden. Diperoleh dari <https://www.diva-portal.org/smash/get/diva2:846121/FULLTEXT01.pdf>.
- P. Candra Adi, PHP vs CodeIgniter vs Yii Framework. [Ebook]  
Diperoleh dari <http://candra.web.id>.
- John Deacon. Model-View-Controller (MVC) Architecture, 2009.  
Diperoleh dari <https://www.rareparts.com/pdf/MVC.pdf>



## **Ringkasan**

Walaupun MARDI telah beralih daripada kaedah pengaturcaraan tradisional kepada menggunakan rangka kerja CI, dari semasa ke semasa kajian semula penggunaan teknologi terhadap keberkesanan dari aspek keselamatan dan kesesuaian perlu sentiasa dilaksanakan. Ini kerana dunia teknologi sentiasa berubah dan dinamik yang mana pada suatu masa nanti perubahan kepada bentuk membangunkan sistem juga akan berubah. Buat masa sekarang CI adalah kerangka kerja yang sesuai untuk diadaptasi dalam aktiviti pembangunan aplikasi di MARDI. Walau bagaimanapun, penambahbaikan dan peningkatan aspek teknikal pengaturcaraan perlu sentiasa dipertingkatkan bagi mengoptimumkan penggunaan terhadap kelebihan yang ada pada CI.

## **Summary**

Despite MARDI has shifted from traditional programming to CI framework, from time to time a review of the use of technology towards the effectiveness in terms of safety and suitability should always put into a consideration. This is because the world of technology is constantly changing and evolving. The techniques and form of developing systems will also change. As of now, CI is a compatible framework to MARDI in assisting MARDI teams on development and maintenance of application. However, the improvement and enhancement of the technical aspects of programming must be constantly improved to optimise the use of the advantages of CI. CI helps us to make it an easy task by giving us the basics, but we can extend it with our own coding skills.

## **Pengarang**

Mohamad Zulkifly Zakaria @ Mustafa  
Pusat Pengurusan ICT, Ibu Pejabat MARDI, Persiaran MARDI-UPM  
43400 Serdang, Selangor  
E-mel: zulfly@mardi.gov.my

Mohd Fauzi Mohd Nazeri  
Pusat Pengurusan ICT, Ibu Pejabat MARDI, Persiaran MARDI-UPM  
43400 Serdang, Selangor