

Penerapan Konsep MVC Pada Aplikasi Web Menggunakan Framework Laravel

Hasyrif SY, Rismayani

STMIK Dipanegara Makassar

Jl. Perintis Kemerdekaan Km 9 Makassar, telp 0411-587194 fax. 0411-588284

e-mail: hasyrif@gmail.com, maya_setya@ymail.com

Abstrak

Berbicara mengenai aplikasi berbasis web biasanya terbagi menjadi dua bagian, yaitu bagian yang berhubungan dengan logika (back-end) dan bagian yang berhubungan dengan tampilan (front-end). Namun aplikasi yang dibuat pada umumnya tidak memisahkan kedua bagian tersebut. Masalah muncul ketika aplikasi berkembang menjadi lebih besar, kondisi ini dapat menyebabkan programmer sulit untuk melakukan modifikasi / pengembangan program, kode yang bercampur tersebut (logika dan tampilan) menuntut programmer untuk bekerja lebih serius. Dampak lainnya yaitu sulitnya para programmer (jika aplikasi dikerjakan lebih dari satu programmer) untuk saling berkolaborasi. Metode yang digunakan untuk menyelesaikan solusi ini yaitu dengan menggunakan konsep MVC. Dengan konsep ini aplikasi dibagi menjadi tiga bagian yaitu Model, View dan Controller sehingga struktur aplikasi menjadi lebih jelas. Dengan penerapan konsep ini diharapkan proses pengerjaan aplikasi bisa menjadi lebih cepat dan memudahkan programmer untuk bekerja sama.

Kata kunci: *Framework, Laravel, MVC.*

Abstract

Talking about web-based applications are usually divided into two parts, namely the part that relates to logic (back-end) and the part associated with the display (front-end). However, the application made generally does not separate the two sections. Problems arise when applications become larger, this condition can lead programmer is difficult to do the modification / development program, the mixed code (logic and display) require programmers to work more seriously. Another effect is the difficulty of the programmer (if the application is done more than one programmer) to collaborate. The method used to as completing this solution is by using the concept of MVC. With this concept the application is divided into three parts: Model, View and Controller so that the structure becomes more obvious applications. With the implementation of this concept is expected to process the application processing can be faster and easier for programmers to work together.

Keywords: *Framework, Laravel, MVC.*

1. Pendahuluan

Membangun aplikasi komputer (program) bukan merupakan hal yang sederhana, karena didalamnya terdapat beberapa aspek yang perlu diperhatikan. Aplikasi dapat di ibaratkan sebuah bangunan, rumah tentunya merupakan sebuah bangunan besar yang di bentuk oleh potongan-potongan yang lebih kecil seperti pondasi, dinding dan atap begitu pula dengan aplikasi, ia terbentuk juga dari sesuatu yang lebih kecil dan saling berhubungan sehingga membentuk sebuah aplikasi lengkap, didalamnya dibangun dari sekumpulan kode-kode dengan tugas dan fungsi tertentu, kode ini bahkan bisa sampai beribu-ribu baris. Karena kerumitan dari aplikasi tersebut tentunya proses pembuatannya harus di rencanakan dengan matang, aspek ini harus diperhatikan dengan serius karena merubah aplikasi yang sudah selesai membutuhkan proses yang rumit dan membutuhkan waktu yang lama, karena komponen diantara software tersebut saling berkaitan, sedikit perubahan bisa saja mempengaruhi mempengaruhi sebagian atau bahkan seluruh komponen-komponen lainnya.

Jika mengambil kasus dari sebuah program yang berbasis web dari sisi kode program kita bisa melihat bahwa sebuah aplikasi terdiri dari tiga bagian yaitu kode yang berfungsi mengatur

antarmuka/interface program atau dikenal dengan istilah front-end, kode yang mengatur interaksi database dan kode yang berhubungan dengan logika atau di kenal dengan istilah back-end. Dalam pengembangan aplikasi tersebut biasanya ketiga bagian tergabung menjadi satu bagian namun semakin banyak modul aplikasi yang akan dibuat maka model ini kurang baik diterapkan karena dapat membingungkan programmer dalam melakukan pengembangan aplikasi hal ini disebabkan karena aplikasi harus lebih teliti dalam melihat kode program yang sedang dibuat, selain itu sudah lazim diketahui bahwa aplikasi web ideanya minimal dikerjakan oleh dua orang programmer, satu programmer menyelesaikan logika aplikasi (back-end) dan programmer lainnya mengerjakan antarmuka dari aplikasi (front-end). Model yang sama juga akan menimbulkan masalah karena sulit rasanya dua orang programmer mengerjakan aplikasi secara bersama karena ketika programmer mengubah kode yang berurusan dengan aplikasi biasanya mempengaruhi kode-kode yang berhubungan dengan antarmuka program, ini menyulitkan terjadinya koodinasi yang baik diantara kedua programmer tersebut.

Salah satu konsep yang dibuat oleh pakar pemrograman adalah memecah sebuah aplikasi web menjadi tiga bagian yaitu model yang berkaitan dengan operasi yang berhubungan dengan basis data, view yang berhubungan dengan antarmuka aplikasi dan terakhir controller yang berhubungan dengan logika aplikasi dan mengendalikan alur data antara view dan controller konsep ini merupakan pengembangan dari konsep pemrograman berorientasi objek dan dalam pemrograman dikenal dengan MVC. Penerapan konsep ini akan coba di jabarkan oleh penulis untuk memberikan gambaran tentang bagaimana model kerja dari konsep MVC ini. Harapannya dengan tulisan ini akan memberi gambaran dan lebih memudahkan para pengembang aplikasi web dalam membangun aplikasinya kedepan.

Untuk memberi contoh praktis tentang konsep ini maka penulis memilih menggunakan framework laravel sebagai kerangka kerja yang menerapkan konsep MVC tersebut, pemilihan framework ini adalah karena dua tahun terakhir framework yang paling sering digunakan adalah framework laravel.

2. Metode Penelitian

2.1. Metode

Adapun metode yang digunakan adalah menggunakan metode MVC (Model, View dan Controller), MVC merupakan salah satu konsep pemrograman yang membagi aplikasi menjadi tiga bagian, yaitu model yang mengurus interaksi antara aplikasi dan database, view yang mengurus urusan logika pemrograman dan controller yang mengatur interaksi antara view dan model.

3. Hasil dan Pembahasan

3.1. Hasil

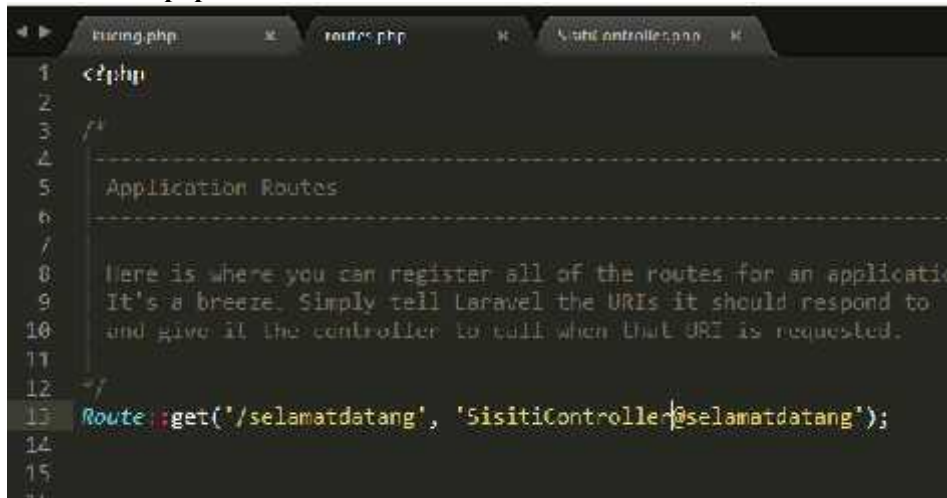


Tampilan Awal Framework Laravel

Gambar 1. Tampilan awal framework laravel

Tampilan default dari framework laravel.

Skrip dari file Routes.php



```
1 <?php
2
3 /*
4  * Application Routes
5  *
6  * Here is where you can register all of the routes for an application.
7  * It's a breeze: Simply tell Laravel the URIs it should respond to,
8  * and give it the controller to call when that URI is requested.
9  *
10  */
11
12
13 Route::get('/selamatdatang', 'SisitiController@selamatdatang');
```

Gambar 2. Skrip dari file routes.php

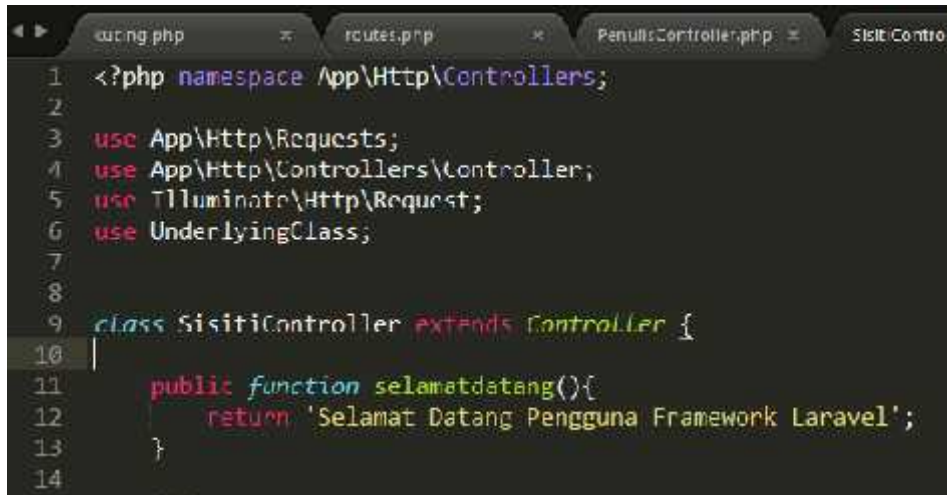
Router merupakan bagian yang mengurus pemetaan / mapping antara url dengan controller, fungsi ini dituliskan dalam sebuah file yang bernama router.php. Ada beberapa model penulisan terhadap file ini, salah satunya mengikuti konvensi berikut :

Route::method('link', namacontroller@methodcontroller, contohnya yaitu :

Route::get('/selamatdatang', 'PenulisController@selamatdatang');

Gambar diatas memperlihatkan contoh penulisan router.php pada laravel

Skrip dari controller



```
1 <?php namespace App\Http\Controllers;
2
3 use App\Http\Requests;
4 use App\Http\Controllers\Controller;
5 use Illuminate\Http\Request;
6 use UnderlyingClass;
7
8
9 class SisitiController extends Controller {
10
11     public function selamatdatang(){
12         return 'Selamat Datang Pengguna Framework Laravel';
13     }
14 }
```

Gambar 3. Skrip Dari Controller

Merujuk pada contoh penulisan router diatas “Route::get('/selamatdatang', 'PenulisController@selamatdatang');” maka penulisan link 'selamatdatang' pada url di browser akan mengarah pada controller yang bernama SisitiController dan methodnya yaitu selamatdatang().



Gambar 4. Tampilan Output Browser

Tampilan Output Browser

Merupakan tampilan output dari browser, string “Selamat Datang Pengguna Framework Laravel” di kembalikan oleh method selamat datang.

Contoh Lain Penulisan Routes.php

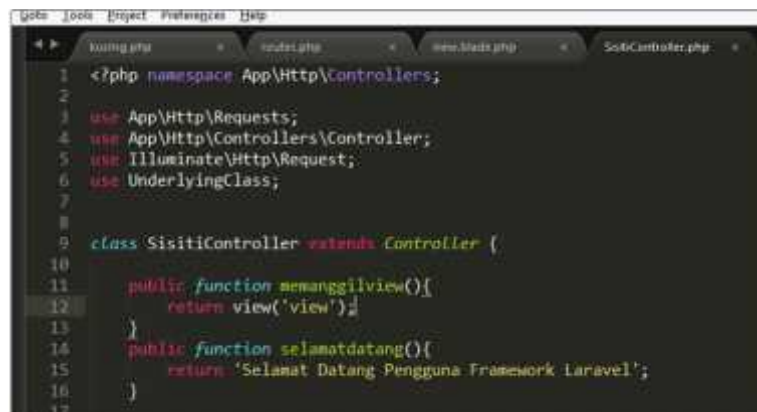


```
1 <?php
2 Route::get('/memanggilview', 'Sisiticontroller@memanggilview');
3 /
4
```

Gambar 5. Contoh Lain Penulisan Routes.php

Skrip diatas akan melakukan mapping terhadap link ‘memanggilview’ pada browser dengan controller yang bernama Sisiticontroller pada method memanggilview().

Controller yang akan menangani request dari routes.php



```
1 <?php namespace App\Http\Controllers;
2
3 use App\Http\Requests;
4 use App\Http\Controllers\Controller;
5 use Illuminate\Http\Request;
6 use UnderlyingClass;
7
8
9 class SisitiController extends Controller {
10
11     public function memanggilview(){
12         return view('view');
13     }
14     public function selamatdatang(){
15         return 'Selamat Datang Pengguna Framework Laravel';
16     }
17 }
```

Gambar 6. Controller yang akan menangani request dari routes.php

Berdasarkan skrip dari routes.php maka method yang akan dijalankan adalah memanggilview(), didalam method ini terdapat perintah untuk memanggil file view. View merupakan skrip yang bertanggung jawab dalam urusan tampilan yang akan dilihat oleh user.

Tampilan output pemanggilan view melalui controller



Gambar 7. Tampilan output pemanggilan view melalui controller

Isi dari file view (yang dipanggil oleh controller)



```
1 <?php
2     echo 'Ini contoh output view dari laravel';
3 ?>
```

Gambar 8. Tampilan output pemanggilan view melalui controller

File diatas merupakan view, bagian ini berisi tampilan / antarmuka yang akan dilihat oleh user, file ini ditampilkan melalui pemanggilan melalui controller. File ini akan menampilkan tulisan 'Ini contoh output view dari laravel'.

Contoh model pada laravel



```
1 <?php namespace App;
2
3 use Illuminate\Database\Eloquent\Model;
4
5 class Buku extends Model {
6
7     protected $table = 'buku';
8     //public $primaryKey = 'id_buku';
9     public $timestamps = false;
10
11     protected $fillable=[
12         'id_buku',
13         'judul_buku',
14         'penulis_id'
15     ];
16
```

Gambar 9. Contoh Model Laravel

Pembuatan model pada laravel, model pada skrip diatas akan digunakan untuk mengakses tabel, dalam kasus ini digunakan untuk mengakses tabel buku.

Routes untuk menghubungkan url dan controller



```
1 <?php
2 Route::get('/ambilisitabelbuku', 'SisitController@ambilisitabelbuku');
3
```

Gambar 10. Tampilan output pemanggilan view melalui controller

Skrip ini akan menghubungkan link pada browser dan controller. Link 'ambilisitabelbuku' akan mengarah pada controller 'sisitcontroller' dan method 'ambilisitabelbuku'.

Controller untuk melakukan load model buku

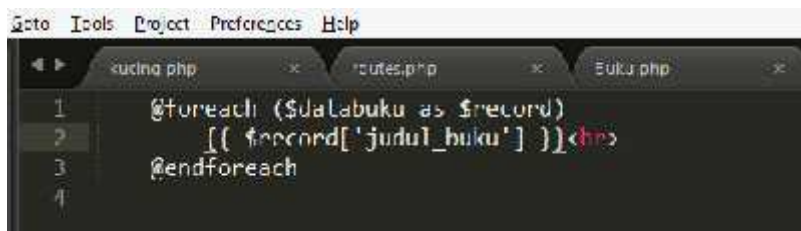


```
1 <?php namespace App\Http\Controllers;
2
3 use App\Http\Requests;
4 use App\Http\Controllers\Controller;
5 use Illuminate\Http\Request;
6 use UnderlyingClass;
7
8 use App\Buku;
9
10
11 class Sisiticontroller extends Controller {
12     public function ambilisitabelbuku(){
13         $databuku = Buku::All();
14         return view('view_isibuku')->withdatabuku($databuku);
15     }
16 }
```

Gambar 11. Controller Untuk Melakukan Load Model Buku

Method “ambilisitabelbuku” secara umum berfungsi meload model buku yang telah dibuat sebelumnya. Model memiliki method yang bernama “All()”, method ini berfungsi untuk mengambil isi dari tabel buku. Pada baris ke 13 terdapat variabel \$databuku yang akan menyimpan record dari tabel buku.

Skrip pada file view_buku.blade.php untuk menampilkan output

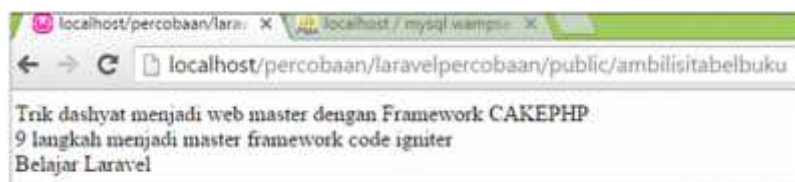


```
1 @foreach ($databuku as $record)
2     [ [ $record['judul_buku'] ] ]<br>
3 @endforeach
4
```

Gambar 12. Skrip Pada File View

Controller akan mengirim data yang disimpan pada variabel \$record variabel ini berisi record-record dari tabel buku. Untuk kali ini hanya akan ditampilkan record dari tabel (hanya mengambil field “judul_buku” saja).

Tampilan Output Browser



localhost/percobaan/laravelpercobaan/public/ambilisitabelbuku

Trik dashyat menjadi web master dengan Framework CAKEPHP
9 langkah menjadi master framework code igniter
Belajar Laravel

Gambar 13. Tampilan Output Browser

3.2. Pembahasan

Teknik pemrograman OOP Dalam PHP

Pemrograman berorientasi Objek (OOP) adalah sebuah teknik pemrograman yang dalam proses pengembangannya menggunakan terminologi objek, dimana setiap objek memiliki atribut beserta dengan

fungsi yang dapat saling berinteraksi satu dengan yang lain seperti halnya objek[2]. Teknik pemrograman OOP bisa disebut juga sebagai teknik pemrograman modern.

Berikut istilah penting yang sering digunakan dalam konsep OOP[2] :

1. Objek, merupakan kumpulan variabel dan fungsi yang saling berkaitan.
2. Clas, merupakan suatu rancangan dari objek itu sendiri. Jadi satu class dapat berisi banyak objek.
3. Inheritance, merupakan kemampuan untuk mengextends class yang akan mewarisi semua karakter dari kelas induk
4. Polimorfisme, merupakan kemampuan objek-objek yang berbeda kelas, tetapi terkait dalam pewarisan untuk merespons secara berbeda terhadap suatu pesan yang sama.

Keuntungan dari teknik OOP[1] yaitu :

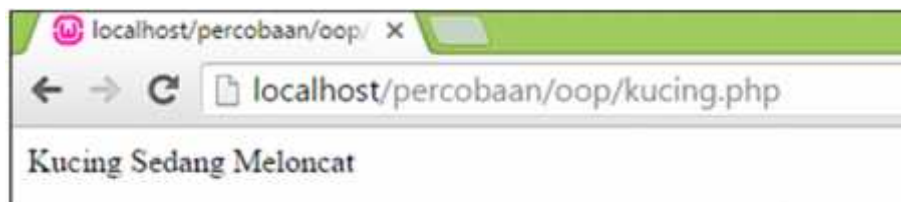
1. Reusability, artinya kemampuan untuk dapat membentuk banyak objek dengan hanya menggunakan satu class dan dapat digunakan kembali jika dibutuhkan
2. Maintability, artinya teknik OOP mempunyai kemampuan lebih mudah dikelola karena pengembang bisa langsung fokus terhadap permasalahan yang ada tanpa merusak fungsi lain.
3. Extensibilit, artinya kemampuannya dapat diperluas, setiap objek dapat ditambahkan kemampuannya tanpa merusak objek lainnya.

Berikut merupakan contoh penerapan konsep OOP dalam bahasa pemrograman PHP

```
kucing.php
1 <?PHP
2     class Kucing{
3         public function meloncat(){
4             echo 'Kucing Sedang Meloncat';
5         }
6     }
7
8     //instansiasi objek dari kelas Kucing
9     $objek_kucing=new Kucing();
10    $objek_kucing->meloncat();
11 ?>
```

Gambar 14. Skrip Sederhana Kelas dalam PHP

Skrip diatas merupakan skrip sederhana mengenai konsep OOP dalam bahasa OOP, skrip ini hanya membuat deklarasi class (pada baris 1 s/d 9) dan melakukan instansiasi dari class kucing menjadi objek kucing dan menjalankan method yang ada pada class (baris 9 s/d 10).



Gambar 15. Tampilan Output Skrip Sederhana Kelas PHP

Framework

framework adalah suatu kerangka kerja atau juga dapat di artikan sebagai kumpulan script yang dapat membantu pengembangan aplikasi dalam menangani berbagai masalah pemrograman seperti koneksi ke database, pemanggilan variable, dan file, sehingga developer lebih cepat membangun aplikasi[5]. Dalam konteks pemrograman web kita dapat memaknai framework dengan kumpulan sebuah maupun banyak modul-modul dalam bentuk class library yang dapat kita gunakan lagi untuk membentuk

sebuah aplikasi web yang lebih besar dengan memanfaatkan modul-modul class library tersebut. Contoh kasusnya adalah pada umumnya framework menyediakan library yang berfungsi dalam melakukan proses upload file maka ketika membuat aplikasi yang didalamnya ada proses untuk melakukan upload file maka kita cukup menginclude dan menggunakan library ini kedalam halaman aplikasi kita, ini dapat menghemat waktu kita dalam membangun sebuah aplikasi.

Keuntungan yang diperoleh dengan menggunakan framework yaitu :

1. Proses pembuatan aplikasi web jauh lebih cepat dibanding tidak menggunakan framework
2. Aplikasi web lebih mudah dikembangkan, maksudnya semua kode yang dibuat dengan menggunakan framework lebih mudah dibaca karena skrip yang dibuat cenderung lebih sedikit dibanding menggunakan tanpa menggunakan framework
3. Pengembang dapat fokus kepada permasalahan yang perlu ditangani, artinya jika sewaktu-waktu web mengalami perubahan, ia dapat melakukan perbaikan pada hal yang bermasalah saja tanpa perlu merombak skrip yang sudah ada.
4. Tingkat keamanan juga akan lebih terjaga.
5. Memudahkan kerjasama team dalam membangun sebuah program besar. Framework “memaksa” programmer untuk mengikuti aturan penulisan dari framework tersebut sehingga standarisasi kode program terjaga secara ketat, artinya sesama programmer akan lebih mudah membaca kode yang dibuat oleh masing-masing programmer.

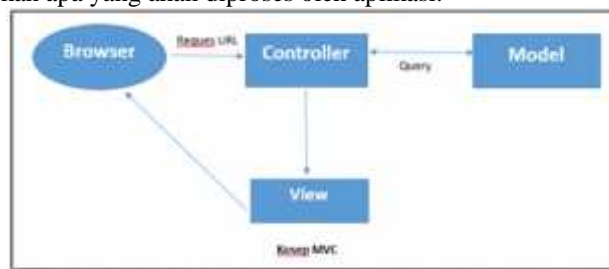
Macam-macam framework PHP yang digunakan saat ini :

1. Laravel
2. Yii
3. CakePhp
4. CodeIgniter, Dll

Konsep MVC

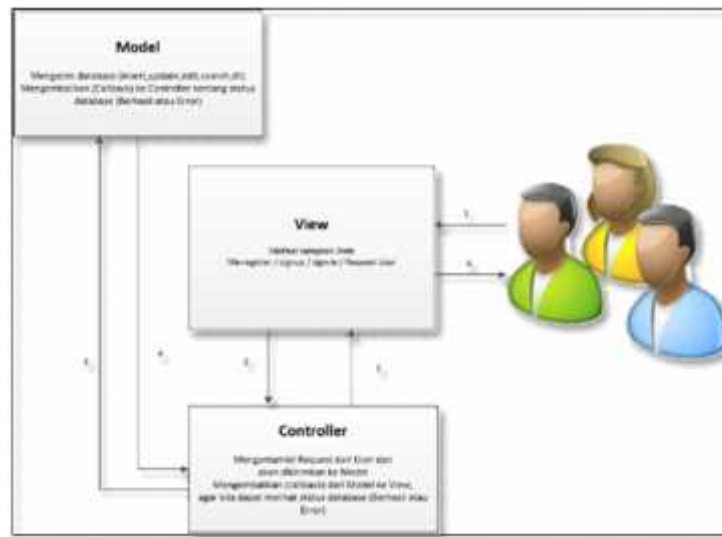
MVC merupakan singkatan dari Model, View dan Controller, istilah ini merupakan sebuah konsep pemrograman dimana menurut paradigma mvc aplikasi dibagi menjadi bahwa aplikasi seharusnya dipisah menjadi dua bagian, yaitu kode yang mengurus logika program dan bagian lain yaitu kode yang mengurus presentasi atau tampilan program. Dalam konsep MVC ada tiga komponen yang saling berinteraksi didalamnya yaitu :

1. View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.
3. Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.



Gambar 16. Konsep OOP

Agar lebih jelas kami akan memberikan penjelasan sederhana alur aplikasi yang mengadopsi konsep MVC



Gambar 17. Alur Aplikasi (Konsep MVC)

1. User melihat tampilan web, halaman web merupakan sarana yang menghubungkan user dan aplikasi.
2. User berinteraksi dengan aplikasi melalui tombol, link dan lain-lain, operasi ini akan menghasilkan request-request tertentu terhadap sumber daya dari sebuah aplikasi.
3. Dari request tersebut user bisa saja melakukan bermacam-macam request, request tersebut salah satunya adalah permintaan sumber daya aplikasi berupa record dari suatu tabel (misal dalam kasus pencarian data tertentu). Request tersebut akan ditangani oleh controller.
4. Controller mengambil data melalui model, model ini berperan sebagai penengah antara controller dan database. Model merupakan struktur data. Secara spesifik class model akan mengandung fungsi kode yang akan membantu dalam segala proses yang berhubungan dengan database seperti memasukkan, mengedit, mendapatkan dan menghapus data dalam sebuah database. Biasanya keluaran dari sebuah model dalam bentuk array.
5. Output dari model akan diterima oleh controller dan controller mengirimkan data dari model ke view atau bagian yang berurusan dengan tampilan program.
6. User melihat hasil dari request yang dilakukan sebelumnya.

Framework Laravel

Laravel adalah Framework pengembangan web MVC yang ditulis dalam bahasa PHP[3]. Framework ini di buat oleh Taylor Otwell, Laravel merupakan salah satu dari sekian framework yang terbanyak digunakan. Framework ini juga mengadopsi konsep Model-View-Controller, selain itu framework ini juga menyediakan librari-librari umum yang biasa diperlukan pada saat pengerjaan aplikasi. Laravel berada dibawah lisensi MIT.

Framework Laravel dibuat oleh Taylor Otwell, proyek Laravel dimulai pada April 2011. Awal mula proyek ini dibuat karena Otwell sendiri tidak menemukan framework yang up-to-date dengan versi PHP. Mengembangkan framework yang sudah ada juga bukan merupakan ide yang bagus karena keterbatasan sumber daya. Dikarenakan beberapa keterbatasan tersebut, Otwell membuat sendiri framework dengan nama Laravel. Oleh karena itu Laravel mensyaratkan PHP versi 5.3 keatas.

Beberapa fitur yang dimiliki framework Laravel :

1. Bundles yaitu sebuah fitur dengan system pengemasan modular dan berbagai bundle telah tersedia untuk di gunakan dalam aplikasi Anda.
2. Eloquent ORM merupakan penerapan PHP lanjutan dari pola "active record", menyediakan metode internal untuk mengatasi kendala hubungan antara objek database. Pembangun query Laravel's Fluent ini didukung oleh Eloquent.
3. Application logic merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan Controllers atau sebagai bagian dari deklarasi Route. Sintak yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh framework Sinatra.

4. Reverse routing, mendefinisikan hubungan antara Link dan Route, sehingga jika suatu saat ada perubahan pada route secara otomatis akan tersambung dengan link yang relevan. Ketika Link yang dibuat dengan menggunakan nama-nama dari Route yang ada, secara otomatis Laravel akan membuat URI yang sesuai.
5. Restful controllers, memberikan sebuah option (pilihan) untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.

4. Kesimpulan

Kesimpulan dari penelitian ini yaitu penggunaan konsep MVC akan membuat kode aplikasi menjadi lebih terstruktur. Pembagian / pemisahan aplikasi menjadi 3 bagian (Model, View dan Controller) membuat programmer lebih mudah melakukan proses pengembangan program karena setiap kode memiliki posisi masing-masing sehingga proses analisa aplikasi menjadi lebih mudah. Pemisahan ini juga memungkinkan aplikasi dikerjakan oleh orang yang berbeda, misal satu orang mengurus logika dan yang lainnya menangani tampilan aplikasi, hal ini dapat terjadi karena dalam konsep ini kode yang berhubungan dengan logika dan kode yang berhubungan dengan tampilan aplikasi terpisah.

Referensi

- [1] Basuki. Awan Pribadi, 2011, Membangun Web Berbasis PHP dengan Framework Codeigniter.
- [2] Saputra. Agus, 2012, Trik Dashyat menjadi web master dengan Framework CakePHP. Andi Publisher Lokomedia.
- [3] McCool. Shawn, 2012, "Laravel Starter The definitive introduction to the Laravel PHP web development framework". Packtpub
- [4] Rosmala. Dewi, Ichwan. Muhammad, Gandlisha. M Irzan, (2013) "Komparasi Framework MVC (Codeigniter, dan CakePHP) Pada Aplikasi Berbasis Web". Jurnal Informatika No.2, Vol. 2, Mei-Agustus 2011.
- [5] Sakur. Stendy B, 2011, PHP 5 Pemrograman Berorientasi Objek Konsep & Implementasi. Andi Publisher
- [6] Wiswakarma Komang, 2010. 9 Langkah Menjadi Master Framework Codeigniter. Lokomedia