



Artikel

APLIKASI PENDAFTARAN SEMINAR MENGGUNAKAN METODE MVC BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK CODEIGNITER 3.1.10

Dermawan¹, Dicky Surya Dwi Putra², Lianny Wydiastuty Kusuma³

1, 2 Universitas Buddhi Dharma, Teknik Informatika, Banten, Indonesia

SUBMISSION TRACK

Diterima: 24 Februari 2020
Revisi Terakhir: 30 Januari 2020
Available Online:

KEYWORD

Pendaftaran Seminar, MVC, Framework, Codeigniter, Website

KORESPONDENSI

Phone:

¹E-mail: dermawansevenfold@gmail.com

A B S T R A K

Seminar adalah sebuah pertemuan yang bertujuan untuk membahas suatu masalah secara ilmiah. Perkembangan teknologi informasi dan komunikasi semakin berkembang pesat, hampir semua semua seluruh komponen dan institusi negeri maupun swasta bahkan kalangan masyarakat umum telah menggunakan teknologi informasi sebagai sarana komunikasi. Teknologi internet yang saat ini kian marak dan berkembang pesat adalah teknologi internet. Website merupakan kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait di mana masing-masing dihubungkan dengan jaringan-jaringan halaman (hyperlink). Framework adalah kerangka kerja. Framework juga dapat diartikan sebagai kumpulan scrip (terutama class dan function) yang dapat membantu developer/programmer dalam menangani berbagai masalah-masalah dalam pemrograman, seperti koneksi ke database, pemanggilan variable, file, dan lain-lain sehingga pekerjaan developer lebih focus dan lebih cepat dalam membangun aplikasi.

PENDAHULUAN

Seminar adalah sebuah pertemuan yang bertujuan untuk membahas suatu masalah secara ilmiah. Untuk menghadiri acara seminar, beberapa seminar menggunakan tiket sebagai akses masuk. Tiket adalah sebuah tanda masuk yang digunakan untuk mendatangi atau menghadiri sebuah tempat/acara tertentu. Dalam sebuah acara

seminar, peserta harus mendaftar terlebih dahulu untuk mendapatkan tiket yang digunakan untuk masuk kedalam acara seminar dan tiket tersebut masih menggunakan tiket dalam bentuk kertas atau yang biasa disebut dengan tiket konvensional [1].

Perkembangan teknologi informasi dan komunikasi semakin berkembang pesat, hampir semua semua seluruh komponen dan

institusi negeri maupun swasta bahkan kalangan masyarakat umum telah menggunakan teknologi informasi sebagai sarana komunikasi. Teknologi internet yang saat ini kian marak dan berkembang pesat adalah teknologi internet. Teknologi internet yang sering diaplikasikan dalam bentuk basis *Web* semakin lama semakin dikembangkan untuk berbagai keperluan dan kebutuhan [2].

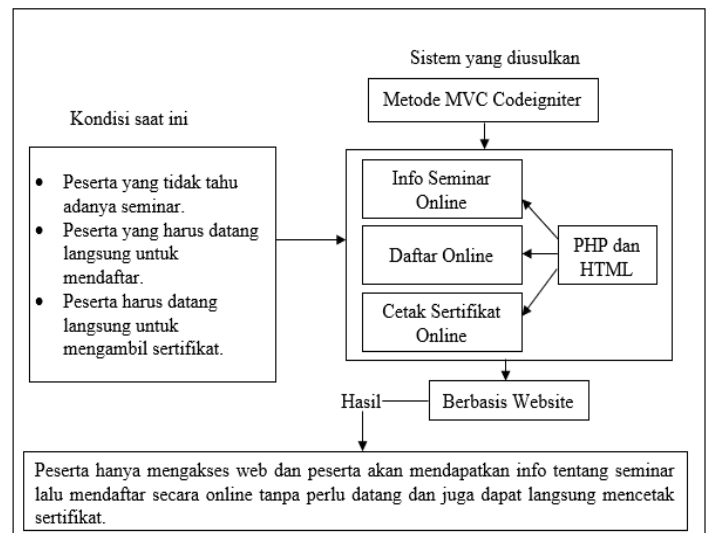
Website merupakan kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait di mana masing-masing dihubungkan dengan jaringan-jaringan halaman [3].

Pola *MVC* memecahkan sebuah aplikasi menjadi tiga modul asosiasi: *model*, *view*, dan *controller*. *Model* adalah logika bisnis dari aplikasi dan inti dari sebuah aplikasi. *View* adalah *user interface* dari *controller* yang merupakan muka umum untuk respon *event* pengguna. Komponen *controller* mengimplementasi *flow* yang mengontrol antara *view* dan *model* [4].

Codeigniter merupakan sebuah *web framework* yang dikembangkan oleh Rick Ellis dari *Ellis Lab*. *Codeigniter* dirancang untuk menjadi sebuah *web framework* yang ringan dan mudah untuk digunakan. Bahkan pengakuan dari Rasmus Lerdorf, sang pencipta Bahasa pemrograman *PHP* mengatakan bahwa *Codeigniter* merupakan *web framework* yang mudah, cepat dan handal [5]

Berdasarkan pada hal yang dijabarkan di atas, maka dibangun sebuah aplikasi pendaftaran seminar berbasis *web* dengan metode (*MVC*), sehingga seseorang akan mudah dalam pendaftaran seminar tanpa harus datang ke tempat penyelenggara seminar tersebut.

KERANGKA PEMIKIRAN



I. METODE

1.1. MVC

Model-View-Controller atau *MVC* adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*) [6].

- Model* mewakili struktur data. Biasanya *model* berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data, seperti memasukkan data ke basis data, pembaruan data, dan lain-lain.
- View* adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
- Controller* merupakan bagian yang menjembatani *model* dan *view*. *Controller* berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman *web*.

1.2. FAMEWORK

Framework adalah kerangka kerja. Framework juga dapat diartikan sebagai kumpulan scrip (terutama class dan function) yang dapat membantu developer/programmer dalam menangani berbagai masalah-masalah dalam pemrograman, seperti koneksi ke database, pemanggilan variable, file, dan lain-lain sehingga pekerjaan developer lebih focus dan lebih cepat dalam membangun aplikasi. Framework adalah komponen pemrograman yang siap digunakan kapan saja sehingga programmer tidak harus membuat script yang sama untuk tugas yang sama. [6]

1.3. CODEIGNITER



Gambar 1: Logo Codeigniter

Codeigniter adalah sebuah *framework PHP* yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi *web* berbasis *PHP* dibandingkan jika menulis semua kode program dari awal [7].

Adapun beberapa keuntungan menggunakan *Codeigniter*, diantaranya:

1. Gratis
Codeigniter berlisensi di bawah *Apache/BSD opensource*.
2. Ditulis menggunakan *PHP 4*
Meskipun *Codeigniter* dapat berjalan di *PHP 5*, namun sampai saat ini kode program *Codeigniter* masih dibuat dengan menggunakan *PHP 4*.
3. Berukuran kecil
Ukuran *Codeigniter* yang kecil merupakan keunggulan tersendiri.

Dibandingkan dengan *framework* lain yang berukuran besar.

4. Menggunakan konsep *MVC*
Codeigniter menggunakan konsep *MVC* yang memungkinkan pemisahan *layer application-logic* dan *presentation*.
5. *URL* yang sederhana
Secara default, *URL* yang dihasilkan *Codeigniter* sangat bersih dan *Search Engine Friendly(SEF)*.
6. Memiliki paket *library* yang lengkap
Codeigniter mempunyai *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis *web*, misalnya mengakses *database*, mengirim *email*, memvalidasi *form*, menangani *session* dan sebagainya.
7. *Extensible*
Sistem dapat dikembangkan dengan mudah menggunakan *plugin* dan *helper*, atau dengan menggunakan *hooks*.
8. Tidak memerlukan *template engine*
Meskipun *Codeigniter* dilengkapi dengan *template parser* sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.
9. Dokumentasi lengkap dan jelas
Dari sekian banyak *framework*, *Codeigniter* adalah satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas.

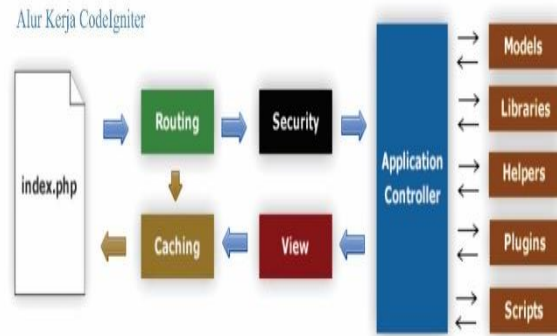
1.4. ANALISA KEBUTUHAN APLIKASI

Berdasarkan analisa kebutuhan pengguna di atas dibuatlah program aplikasi yang dapat memenuhi kebutuhan pengguna yang dijelaskan pada table di bawah ini:

Tabel 1. Analisa Kebutuhan Aplikasi

No	Kebutuhan Pemakai	Keterangan
1	Tampilan yang rapih dan dapat diakses oleh media manapun.	√
2	Memberikan data pendaftaran yang jelas.	√
3	Memiliki metode pembayaran yang mudah seperti cash, transfer bank, dll.	√
4	Memiliki fitur untuk memberikan sertifikat secara online.	√
5	Dapat mentracking tempat seminar melalui google maps.	x
6	Memberikan pesan validasi terhadap data pendaftar.	√
7	Memiliki fitur untuk mengupload bukti transfer.	√
8	User dapat dengan mudah mengakses info seminar.	√
9	Memberikan kemudahan dalam mendaftar seminar.	√
10	Sertifikat seminar tersedia diwebsite dan bisa didownload.	√
11	Dapat dengan mudah merubah data user diprofile.	x
12	Tampilan pada web menggunakan bahasa indonesia.	√
13	User dapat mengetahui status pembayaran.	√

II. HASIL



Gambar 2: Alur Kerja Codeigniter

1. *File index.php* (kadang disebut *entry scripts*) berfungsi sebagai *front controller*, menginisialisasi *resource* utama yang dibutuhkan untuk menjalankan *Codeigniter*.
2. *Router* memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan dengan itu.
3. Jika *file cache* ada, dikirim langsung ke *browser*, melewati eksekusi sistem normal.
4. *Security*, Sebelum *controller* aplikasi dimuat, *HTTP request* dan setiap data pengguna yang di-submit disaring terlebih dahulu untuk keamanan.
5. *Controller* memuat *model*, *library* utama, *helper*, dan setiap *resource* lainnya yang diperlukan untuk memproses permintaan khusus.
6. *View* di-render kemudian dikirim ke *web browser* agar dapat dilihat. Jika *caching* diaktifkan, *view* di-cache terlebih dahulu sehingga pada permintaan berikutnya dapat dilayani.

III. PEMBAHASAN

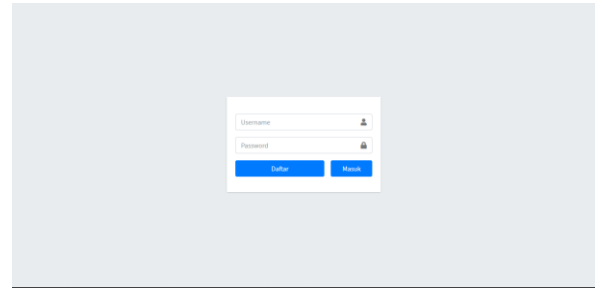
Pengujian *Blackbox* dilakukan dengan tes fungsionalitas dari aplikasi atau perangkat lunak. Tujuan pengujian ini adalah untuk menemukan kesalahan dalam aplikasi. Berikut adalah tabel hasil pengujian *blackbox*:

Tabel 2. Blackbox Testing

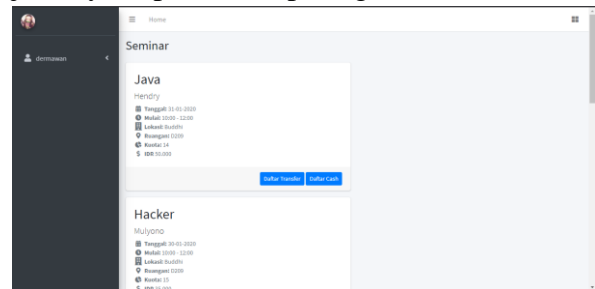
NO	Scene Pengujian	Skenario	Hasil Pengujian
1	Halaman Login	Pengguna menginput Username dan Password dan menekan tombol login untuk melakukan login atau menekan tombol daftar untuk pindah ke halaman daftar	Sesuai
2	Halaman Register	Pengguna menginput data diri lalu setelah itu menekan tombol register atau menekan tombol back	Sesuai
3	Halaman Dashboard Admin	Pengguna menekan tombol home atau seminar atau logout	Sesuai
4	Halaman Beranda User	Pengguna menekan tombol seminar untuk melihat seminar yang telah pengguna daftar tombol logout untuk keluar aplikasi dan tombol daftar untuk daftar seminar dan home untuk ke beranda	Sesuai

Setelah menyelesaikan pengerjaan aplikasi adapun tampilan akhir aplikasi ini adalah sebagai berikut:

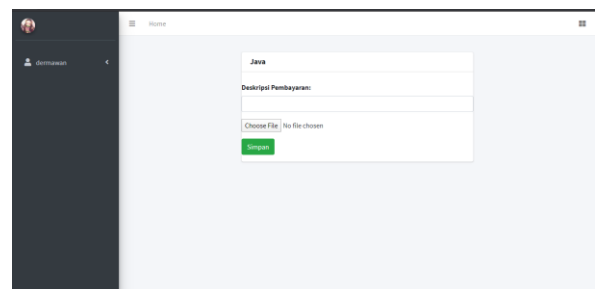
Berikut ini adalah tampilan halaman login dalam aplikasi ini, untuk lebih jelasnya dapat dilihat pada gambar berikut ini:

**Gambar 3: Tampilan Halaman Login**

Berikut ini adalah tampilan halaman beranda user dalam aplikasi ini, untuk lebih jelasnya dapat dilihat pada gambar berikut ini:

**Gambar 4: Tampilan Halaman *Homepage User***

Berikut ini adalah tampilan halaman daftar via transfer dalam aplikasi ini, untuk lebih jelasnya dapat dilihat pada gambar berikut ini:

**Gambar 5: Tampilan Halaman Pendaftaran Seminar**

Berikut ini adalah tampilan halaman cetak sertifikat dalam aplikasi ini, untuk lebih jelasnya dapat dilihat pada gambar berikut ini:



Gambar 6: Tampilan Halaman *Download* Sertifikat

IV. SIMPULAN

Setelah dilakukan perancangan, pembuatan dan pengujian maka dapat diambil kesimpulan sebagai berikut:

1. Dengan adanya aplikasi pendaftaran seminar ini, Dapat membantu dan mempermudah pengguna dalam mendapatkan informasi tentang seminar dan juga dalam melakukan pendaftaran seminar lalu dapat mencetak sertifikat secara langsung.
2. Dengan metode *mvc* proses pembuatan aplikasi pendaftaran seminar dapat diimplementasikan.
3. Dengan metode *mvc* proses pengembangan menjadi lebih mudah.
4. Dengan *framework codeigniter* aplikasi *web* pendaftaran seminar menjadi *web* dinamis.

REFERENSI

- [1] D. A. A. Nugroho dan H. Supriyono, “Sistem Informasi Pendaftaran Seminar Dengan Tiket Berbasis Qr Code,” *Jurnal Emitter*, pp. 34-40, 2019.
- [2] M. Agarina dan A. S. Karim, “Rancangan Bangun Sistem Informasi Kegiatan Seminar Nasional Berbasis Web Pada Institut Informatics dan Bisnis Darmajaya,” *Jurnal Sistem Informasi dan Telematika*, pp. 26-32, 2019.
- [3] F. Membuat Website Mudah dan Praktis dengan Weebly, Jakarta: PT Elek Media Komputindo, 2014.
- [4] P. Simanjuntak dan A. Kasnady, “Analisis Model View Controller (MVC) Pada Bahasa PHP,” *Jurnal ISD Vol.2 No.2*, pp. 56-66, 2016.
- [5] A. Subagia, *Menguasai Framework Codeigniter 3*, Cirebon: CV. ASFA Solution, 2018.
- [6] Y. Yudhanto dan H. A. Prasetyo, *Panduan Mudah Belajar Framework Laravel*, Jakarta: PT Elex Media Komputindo, 2018.
- [7] L. Hakim, *Membangun Web Berbasis PHP dengan Framework Codeigniter*, Yogyakarta: Lokomedia, 2010.

BIOGRAFI

Dermawan lahir di Tangerang pada 02 Februari 1997. Menyelesaikan pendidikan SMA Setia Bhakti pada tahun 2015, dan mendaftar di Universitas Buddhi Dharma pada tahun 2015 jurusan teknik informatika.

Dicky Surya Dwi Putra lahir di Tangerang pada 27 Juni 1987. Menyelesaikan pendidikan S1 (S.Kom.) di STMIK Buddhi dan pendidikan S2 (M.Kom.) di Eresha School of IT pada tahun 2012. Sekarang aktif sebagai dosen tetap di Universitas Buddhi Dharma sejak tahun 2009.

Lianny Wydiastuty Kusuma, adalah lulusan dari STMIK Buddhi dan Universitas Budi Luhur serta aktif sebagai dosen tetap di Universitas Buddhi Dharma hingga kini.