

Penerapan Web Service JSON pada Backend-Developer Summary Report Executive Menggunakan Arsitektur MVC CodeIgniter

Robertus Laipaka

STMIK Pontianak

Jalan Merdeka No 372, 0561-735555, Jurusan Sistem Informasi

e-mail: asinfo.cims@gmail.com

Abstrak

Informasi Penerimaan dan Pendapatan menjadi kebutuhan yang sangat penting bagi Executive dalam pengambilan keputusan. Seperti melihat informasi target dan realisasi penerimaan pendapatan, Penerimaan berdasarkan harian, Bulanan dan lokasi unit Layanan. Dengan informasi yang selalu tersedia secara realtime, maka pihak executive tidak akan mengalami kendala dalam mendapatkan informasi yang cepat yang tidak tergantung dengan staf administrasi, bahkan mesti menunggu waktu untuk disiapkan sehingga mengakibatkan pihak executive tidak bisa mengambil keputusan dengan cepat. Atas dasar permasalahan tersebut dapat diatasi dengan membangun aplikasi khusus untuk report executive. Metode yang digunakan dalam pengaksesan data menggunakan webservice dengan JSON sedangkan pada pengembangan sistem ini adalah model waterfall yang terdiri dari fase Communication, Planning, Modeling, Construction, Development. Penelitian ini menghasilkan sebuah sistem informasi yang dapat membantu level manajemen executive untuk mendapatkan informasi penerimaan secara Realtime, karena setelah diuji ketersediaan informasi berdasarkan periode kebutuhan informasi pada level executive menunjukkan bahwa informasi yang dibutuhkan harus tersaji secara cepat dan akurat sehingga dapat diambil suatu keputusan strategis terhadap informasi yang di hasilkan. Selain itu juga penelitian ini dapat memberi acuan dalam proses pembangunan sistem informasi yang terintegrasi menjadi lebih mudah dan efisien

Kata kunci: SR, MVC, CI, JSON, Webservice.

1. Pendahuluan

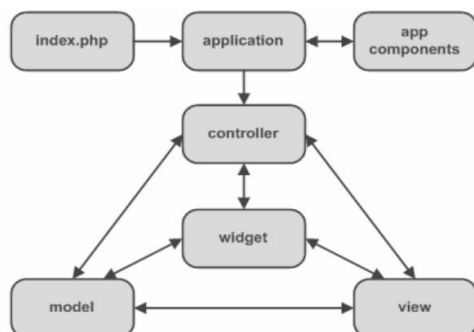
Perkembangan teknologi informasi sudah besinergi baik dilingkungan instansi pemerintahan maupun swasta perlu penanganan yang lebih serius karena pengaruh teknologi informasi begitu besar dalam berbagai aspek kehidupan. Pada level executive perkembangan teknologi informasi tidak hanya berpengaruh pada kinerja suatu instansi namun juga pada proses pengambilan keputusan melalui sistem *computing* dan kecepatan akses data. Bagi executive teknologi informasi di dimanfaatkan untuk mengetahui informasi secara cepat dan akurat. Namun

teknologi ini masih memiliki keterbatasan, yaitu adanya keterbatasan akses informasi dari berbagai platform yang berbeda-beda sehingga sulit untuk melakukan pertukaran data dan informasi. Hal tersebut harus membutuhkan sebuah service yang dapat melakukan sebuah tugas yang spesifik dalam menangani pertukaran data dan informasi yang dikenal dengan web-service. Konsep ini merupakan pengembangan teknologi web yang saat ini banyak digunakan adalah salah satu penerapannya dengan menggunakan framework. Dimana terdapat kumpulan script (terutama class dan function) yang dapat membantu developer dalam menangani berbagai masalah dalam pemrograman[1]. Berdasarkan alasan tersebut penerapan teknologi web-service yang memungkinkan perpaduan fungsi-fungsi untuk membangun sebuah progra aplikasi tanpa bergantung lagi pada ressource sistem operasi maupun bahasa pemrograman yang digunakan, maka dalam sistem ini akan menerapkan web-service untuk membuat backend-Developer summary report executive menggunakan Arsitektur MVC Codeigniter yang ada pada BPKPD Prov.Kal-Bar service tersebut digunakan untuk melihat integrasi antar web-service pada sistem operasi yang berbeda sehingga sistem informasi summary report executive dapat di sajikan melalui sistem terintegrasi. Pertukaran data dan informasi tersebut dilakukan dengan menggunakan dokumen JavaScript Object Notation (JSON). Selain itu beberapa aplikasi berbasis web juga membutuhkan lebih banyak sumber daya untuk membangun sebuah antarmuka yang efektif dan terstruktur secara modular dan itu lebih baik dari pada hanya sekedar logika proses[2]. Sistem berbasis web konvensional, masih mencampur kode program antara logika proses dan antarmuka. Antarmuka dalam sistem berbasis web konvensional hanya dapat digunakan dalam satu proses logika, hal ini akan mengurangi modularitas aplikasi, membuat pemeliharaan sistem lebih sulit, dan juga membuat antarmuka sulit untuk dimodifikasi ketika akan digunakan untuk aplikasi lain[3]. Pada tahun 1970-an penemu smalltalk, Trygve Reenskaug, mendefinisikan sebuah arsitektur untuk menyelesaikan masalah tersebut, yang disebut arsitektur model-view-controller (MVC) yang memisahkan logika bisnis dan presentasi (tampilan) aplikasi. Beberapa penelitian mengenai penerapan web-service diantaranya implementasi web-service pada sistem pelayanan

perijinan terpadu satu atap dipemerintah kota palu [4]. Kemudian penerapan web-service pada aplikasi sistem akademika pada platform sistem operasi mobile android[5]. Teknologi web-service juga diterapkan untuk integrasi data layanan puskesmas dan rumah sakit[6]. Oleh sebab itu, sistem ini akan menerapkan web-service json pada back-end developer summary report pada Arsitektur MVC Menggunakan CodeIgniter. Sistem ini diharapkan dapat menyajikan informasi yang dibutuhkan executive secara cepat dan akurat.

Adapun data yang menjadi objek dalam penelitian ini terdiri dari data dari penerimaan harian, pertanggal, perbulan yang didapat dengan menggunakan web-service yaitu dengan menggunakan JSON. Melakukan wawancara dengan level executive tentang informasi yang dibutuhkan untuk ditampilkan pada aplikasi yang dibangun. Untuk mendukung referensi pada penelitian ini dilakukan studi pustaka yang meliputi teori pendukung tentang objek penelitian baik dalam bentuk buku maupun dalam bentuk jurnal ilmiah.

Sedangkan metode pengembangan sistem yang digunakan adalah model *waterfal l* [8]. Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Fase-fase dalam model *waterfall*, yaitu: *Communication, Planning, Modeling, Construction, Deployment*. *Communication*: tahapan pertama ini dilakukan pengumpulan kebutuhan elemen-elemen *software*. Baik kebutuhan fungsional maupun kebutuhan non-fungsional. Hasil akhir dari tahap ini adalah spesifikasi kebutuhan perangkat lunak. Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya dapat dilakukan oleh sistem. Sedangkan kebutuhan nonfungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem. pada penelitian ini menggunakan arsitektur MVC pada framework codeigniter. Aplikasi mengumpulkan beberapa informasi mengenai request pengguna, kemudian mengirimnya ke controller yang sesuai untuk penanganan selanjutnya[7]. Struktur statis sebuah aplikasi codeigniter seperti dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur MVC pada Codeigniter [7]

Model-View-Controller (MVC) membagi aplikasi menjadi tiga modul asosiasi yang terdiri dari *model*, *view*, dan *controler*. Pada modul *model* digunakan untuk

menelola informasi dan memberitahukan pengamat ketika ada perubahan informasi mengenai *model* yang mengandung data dan fungsi yang berhubungan dengan pemrosesan data. Modul *view* merupakan tampilan antar muka atau interface dari controller untuk merespon event user.

Sedangkan modul Controller menerima input dari user dan menginstruksikan model dan view untuk melakukan aksi berdasarkan event tersebut. *Planning*: Pada tahapan ini akan menghasilkan dokumen user *requirement* atau data yang berhubungan dengan keinginan user dalam pembuatan *software*, termasuk rencana yang akan dilakukan. Untuk mempermudah memahami ruanglingkup sistem yang akan dibangun pada back-end summary report penerimaan pajak kendaraan bermotor untuk executive dapat diuraikan sebagai berikut berdasarkan unit layanan dan periode penerimaan, pihak bpkpd prov.kalbar khususnya *executive summary report* dapat melihat informasi mengenai tingkat penerimaan pajak kendaraan bermotor secara realtime, baik *report* harian, pertanggal maupun berbulan dapat di monitoring secara realtime kapan saja dibutuhkan. Hal ini dapat mempermudah pihak executive dalam mengambil keputusan strategis terhadap kebijakan maupun keputusan yang akan di lakukan maupun yang akan direncanakan untuk menyusun target dan realisasi penerimaan.

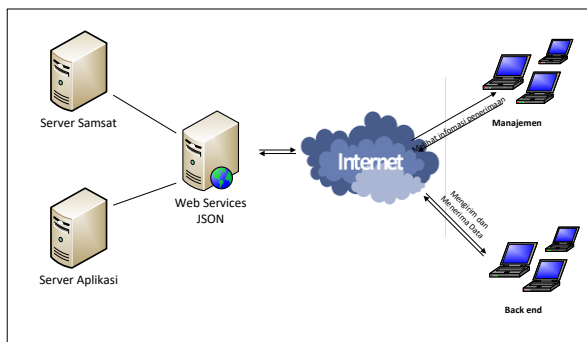
Modeling: Proses ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat coding. Perancangan *software* terdiri atas beberapa langkah, yaitu: desain struktur data, desain arsitektur *software* dan juga desain antar muka. Pada tahapan perancangan, kebutuhan-kebutuhan atau spesifikasi perangkat lunak yang dihasilkan pada tahap analisis ditransformasikan ke dalam bentuk arsitektur perangkat lunak yang memiliki karakteristik mudah dimengerti dan tidak sulit untuk diimplementasikan. Perancangan sistem dalam penelitian ini dibangun dengan pemrograman berbasis objek menggunakan Diagram UML sebagai media visualisasi dan komunikasi antar berbagai pihak yang terlibat dalam pengembangan sistem atau perangkat lunak aplikasi[8].

Web services adalah suatu system perangkat lunak yang didisain untuk mendukung interaksi mesin ke mesin pada suatu jaringan. Ia mempunyai suatu interface yang diuraikan dalam suatu format machine-processible seperti WSDL (Web Service Description Language). Sistem lain yang berinteraksi dengan Web service dilakukan melalui interface/antar muka menggunakan pesan seperti pada SOAP. Pada umumnya pesan ini melalui HTTP dan XML yang merupakan salah satu standard web. [9](W3C, 2004). Konsep arsitektur yang mendasari teknologi Web service adalah Service Oriented Architecture (SOA), SOA mendefinisikan 3 peran berbeda yang menunjukkan peran dari masing-masing komponen dalam system, yaitu[10]:

- Service provider, yaitu suatu entitas yang menyediakan interface terhadap sistem yang menjalankan suatu sekumpulan tugas tertentu.

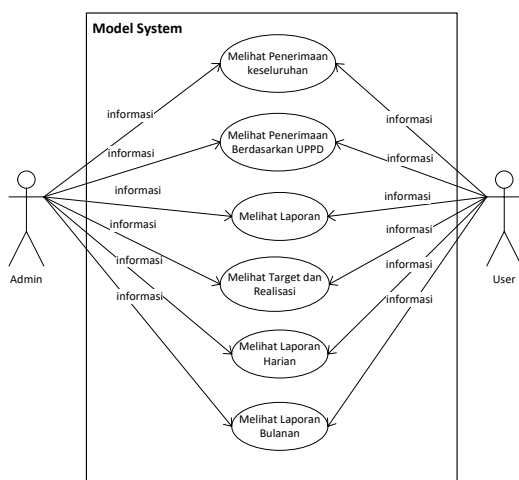
terhindar dari kesalahan yang non teknis. Berikut ini akan diuraikan bagaimana sistem bekerja dalam menyajikan *report* untuk *executive*.

Aplikasi mengumpulkan beberapa informasi mengenai request user melalui server aplikasi menggunakan layanan webservice json kepada server database, kemudian mengirimnya ke user yang sesuai untuk penanganan selanjutnya dapat menggunakan smartphone maupun pc dan laptop. Adapun arsitektur sistem dapat dilihat pada gambar 7.



Gambar 7. Arsitektur sistem

Diagram UML yang digunakan adalah use case diagram . Use case diagram digunakan untuk menunjukkan fungsionalitas sistem maupun sebagai media visualisasi dan komunikasi antar berbagai pihak yang terlibat dalam pengembangan sistem. Pada gambar 8. Pada sistem ini terdapat 2 aktor dan 6 use case diagram. Aktor admin dan user pada usecase diagram spesifikasi terdiri dari proses melihat penerimaan keseluruhan lokasi, melihat detail penerimaan peruppd, melihat target dan realisasi penerimaan, melihat laporan harian, melihat laporan bulanan berdasarkan unit layanan. Adapun yang membedakan antara aktor admin dan user adalah hak akses terhadap sistem dimana hak akses admin lebih lengkap dari pada hak user yang hanya sebagai view informasi dari sistem saja maupun melihat grafik tingkat penerimaan pada tampilan informasi .



Gambar 8. Diagram UseCase

Berikut ini akan ditampilkan *Model-View-Controller* (MVC) yang membagi aplikasi menjadi tiga modul asosiasi yang terdiri dari *model*, *view*, dan *controller*.

```

1 <?php
2
3 class Map_model extends CI_Model{
4
5     function __construct()
6     {
7         parent::__construct();
8     }
9
10    function get_coordinates()
11    {
12        $return = array();
13        $this->db->select("lat,lng");
14        $this->db->from("maps");
15        $query = $this->db->get();
16        if ($query->num_rows()>0) {
17            foreach ($query->result() as $row) {
18                array_push($return, $row);
19            }
20        }
21        return $return;
22    }
23 }
24 }
25 }
26 >>
    
```

Gambar 9. Tampilan Model

Pada bagian model ini berfungsi untuk menangkap keadaan proses ketika ada perubahan informasi pada sistem sehingga *developer* dapat menentukan model yang menghubungkan *beck-end* dengan *front-end* yang di presentasikan pada sebuah user interface. Pada gambar 8 merupakan contoh pendekatan model yang digunakan dari proses untuk mengkap lokasi dan unit layanan.

```

1 <!DOCTYPE html>
2 <head>
3     <meta charset="utf-8">
4     <title>Samsat Kalbar</title>
5
6     <meta name="viewport"
7         content="maximum-scale=1.0,
8             initial-scale=1.0, width=device-width">
9
10    <link rel="icon" type="image/x-icon"
11        href="<?php echo
12            base_url();?>favicon.ico" />
13    <?php echo $map['js'];?>
14 </head>
15 <body>
16    <?php echo $map['html'];?>
17 </body>
18 </html>
    
```

Gambar 10. Tampilan view

Pada bagian view ini berfungsi sebagai penanggung jawab untuk pemetaan grafis ke sebuah perangkat dimana ketika model berubah view secara otomatis menunjukkan perubahan tersebut meskipun pada tampilan permukaan yg berbeda.

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Map extends CI_Controller {
5
6     public function __construct() {
7
8         parent::__construct();
9         $this->load->library(array('session','form_validation',
10             'ion_auth'));
11         $this->load->helper(array('url','form'));
12         $this->load->model('mcrud');
13
14         if (!$this->ion_auth->logged_in()) {
15             redirect('auth','refresh');
16         }
17     }
18
19     function index()
20     {
21
22
23         $this->load->library('googlemaps');
24
25         $this->load->model('map_model', '', TRUE);
26
27         $config=array();
28         $config['center']="37.4419, -122.1419";
29         $config['zoom']=17;
30         $config['map_height']="400px";
31         $this->googlemaps->initialize($config);
32
33         $marker=array();
34         $marker['position']="37.4419, -122.1419";
35
36         $this->googlemaps->add_marker($marker);
37         $data['map']=$this->googlemaps->create_map();
38         $this->load->view('map_view', $data);
39     }

```

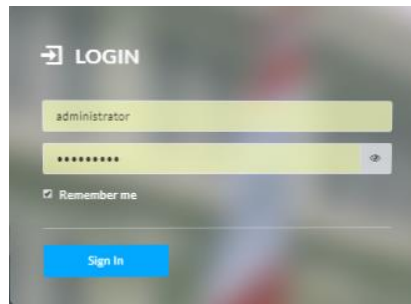
Gambar 9. Tampilan acontroller

Pada bagian Controller ini berfungsi sebagai penerima input dari pengguna dan menginstruksikan model dan view untuk melakukan aksi berdasarkan action user. Sehingga controller bertanggung jawab untuk pemetaan *action and user* terhadap respon aplikasi. Contoh ketika user ingin melihat lokasi layanan controller langsung bertanggung jawab untuk menentukan bagaimana aplikasi seharusnya merespon terhadap permintaan user.

Disisi server aplikasi ini bisa beroperasi pada sistem operasi window dan Android minimal spesifikasi CPU Quad-core 1.2 GHz, *WampServer*, *Xampp* yang digunakan untuk mendeskripsikan web server sedangkan disisi user untuk mendapatkan hasil yang optimal dapat menggunakan akses data internet yang bagus serta memilih browser yang selalu update.

2. Pembahasan

Pengujian penerapan web-service JSON pada back-end summary report executive menggunakan arsitektur *mode-view-control* (MVC) pada framework *Codeigniter* pada sistem ini dapat menyajikan kebutuhan informasi bagi *executive* untuk mengambil suatu keputusan baik informasi secara *Realtime Report* maupun *Bacth Report*. Informasi mengenai tingkat penerimaan dan pendapatan digunakan oleh *executive* untuk melihat target dan realisasi yang di peroleh di setiap lokasi layanan maupun pertanggal layanan terlebih bila ada pihak ketiga dalam hal ini yang ingin mengetahui informasi secara cepat pihak executive dapat memaparkan langsung pada saat dilakukan permintaan informasi tersebut. Adapun meliputi tampilan informasi mengenai Penerimaan harian yang dimulai dari tampilan autentikasi yang memungkinkan user dapat mengakses sistem berdasarkan hak akses yang di miliki. Berikut gambar 11. Halaman login Sistem.



Gambar 11. Tampilan halaman Login

Pada halaman login ini akan berfungsi sesuai dengan yang ada pada diagram usecase yang menjelaskan peranan user dalam mengakes sistem. Bila user salah atau bukan orang yang berwenang dalam mengakses sistem maka sistem akan memberikan *feed back* berupa notifikasi pesan kepada user. Bila user sebagai pemilik hak akses terhadap sistem, maka user akan di arahkan pada tampilan menu *dashboard* seperti pada gambar 12 yang menyajikan widget informasi yang tersaji seperti Widget Penerimaan, Widget Target dan Realisasi, widget Report yang meliputi report pertanggal dan report perbulan.



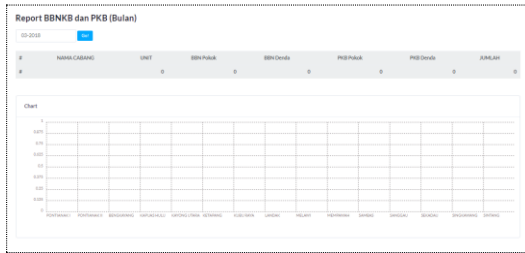
Gambar 12. View Dashboard

Pada tampilan *view* ini berisi kode yang bertugas untuk menampilkan *user interface* sistem.

Target dan Realisasi		01-05-2018		
UNIT	TREND	01-05-2018	02-05-2018	03-05-2018
BBN POKOK		0	0	0
BBN DENDA		0	0	0
PKB POKOK		0	0	0
PKB DENDA		0	0	0
JUMLAH		0	0	0

Gambar 13. View Target dan Realisasi

View informasi mengenai target dan realiasi secara keseluruhan maupun sampai dengan transaksi sampai dengan saat ini yang meliputi item Unit, BBN Pokok, BBN Denda, PKB Pokok, PKB Denda dapat di sajikan kapan saja pihak *executive* membutuhkan



Gambar 14. View Report Bulanan

Pada *view report* bulanan disajikan nama cabang layanan, jumlah unit, BBN Pokok, BBN Denda, PKB Pokok, PKB Denda dan Jumlah. Informasi ini dapat digunakan oleh *executive* dalam mengambil langkah-langkah strategis terhadap trend pembayaran dan penerimaan pendapatan pajak kendaraan bermotor berdasarkan Bulanan. Dengan di suport menggunakan system secara realtime kebijakan strategis dapat di ambil secara cepat tanpa harus melalui *batch report* karena akan terkendala dalam mengambil kebijakan maupun keputusan.

Berikut ini akan dilakukan analisis *realtime Report* dengan *Batch Report* dari hasil pada penerapan web-service json pada back-end summary report executive menggunakan arsitektur *Codeigniter* pada tabel 1. *Realtime Report VS Batch Report*.

Tabel 1. *Realtime VS Batch Report*

No	Informasi yang butuhkan	<i>Realtime</i>	<i>Batch</i>
1	Penerimaan Harian	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Penerimaan Bulanan	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Target dan Realisasi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Grafik Penerimaan	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Grafik Realisasi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Dari penerapan web-service pada back-end summary report executive menggunakan arsitektur MVC codeigniter dapat dilihat pada tabel penyajian informasi untuk *executive* di seluruh lingkungan BPKPD prov. Kal-Bar berdasarkan jenis report yang dibutuhkan dapat diuraikan bahwa level *executive* lebih senang mendapatkan dan menyajikan informasi secara realtime yang dapat di buktikan bila ada informasi penerimaan maupun target pihak *executive* dapat langsung menyebarkan informasi di grup seluruh unit Layanan penerimaan pendapatan yang tersebar di Provinsi Kalimantan Barat. Dengan adanya sistem yang real time khususnya untuk pihak *executive* sangat membantu dan menjadi sumber motivasi bagi unit layanan yang belum terealisasi berdasarkan target yang telah di sepakati dengan pihak *executive*. bukan berarti mengabaikan *Batch Report*, *Batch report* biasanya digunakan untuk merekap secara berkala yang di lakukan oleh petugas yang telah di tunjuk dengan tujuan untuk memastikan penerimaan sikron terhadap realtime yang di sajikan, antar realtime report dan batch report saling ketergantungan.

3. Kesimpulan

Dari hasil penelitian pada penerapan web-service json pada back-end summary report executive menggunakan arsitektur *Mode-View-Controller* (MVC) codeigniter dapat diambil kesimpulan bahwa sistem yang dihasilkan dapat digunakan untuk mendukung *executive* dalam mengambil langkah-langkah dan keputusan yang strategis dapat dilakukan dengan cepat tanpa menunggu dari petugas maupun staf mengumpulkan berbagai sumber informasi untuk disajikan pada periode tertentu dengan menggunakan model *Batch Report*. Sistem ini dapat menyajikan informasi penerimaan secara keseluruhan unit layanan dan lokasi cabang layanan, sistem juga dapat menyajikan target dan realisasi secara realtime untuk mempermudah pihak *executive*.

Agar Pengembangan lebih lanjut diharapkan dapat membuat berbagai versi aplikasi dengan *framework multiplatform* dan penambahan berbagai fitur lainnya seperti masalah keamanan sistem, melakukan integrasi kepada pihak-pihak lain yang membutuhkan informasi sayang disajikan.

Daftar Pustaka

- [1]. A. Ashari, R. Hidayat, "Penerapan Teknologi Web-service untuk integrasi layanan puskesmas dan rumah sakit", 2013 Berkala MIPA, (23)1
- [2]. I.G. Wijaya, "Penerapan Web-Service pada aplikasi sistem akademik pada platform sistem operasi mobile android", 2012 seminar.
- [3]. M. Yazdi, "Implementasi Web-service pada sistem pelayanan perijinan terpadu satu atap di pemerintahan kota palu", 2012 Semantik, ISBN 979 -26-0255-0.
- [4]. A. Hidayat, B. Surarso, "Penerapan arsitektur model view controller (mvc) dalam rancang bangun sistem kuis online adaptif. 2012. In seminar nasional teknologi informasi dan komunikasi.
- [5]. P. Simajuntak, "Analisis model view controller (mvc) pada bahasa php. 2016. Journal information system development (isd), 1(2).
- [6]. J.S. Pasaribu, "Penerapan framework yii pada pembangunan sistem ppdb smp bppi baleendah kabupaten bandung". 2017. Jurnal ilmiah teknologi informasi terapan, 3(2).
- [7]. S. Sumiyatun, F.D. Astuti, S. Windarti, "Penerapan framework dengan arsitektur model-view-controller pada pengelolaan data penelitian dan pengabdian kepada masyarakat". 2015. Jurnal teknologi informasi respati, 10(30).
- [8]. E.N. Jannah, M. Masrur, S. Asiyah, "Penerapan framework yii dalam pembangunan sistem informasi asrama santri pondok pesantren sebagai media pencarian asrama berbasis web". 2015. Journal of information systems engineering and business intelligence, 1(2), 49-58.
- [9]. E.W. Hidayat, "penerapan pola hierarchical model-view-controller pada rekayasa sistem berbasis web framework". 2011. jurnal teknologi technoscintia, 3(1), 169-178.
- [10]. S. Ahmad, "Implementasi Framework Codeigniter dengan metode Model View Controller (MVC) pada pembuatan website Radio SSFM Semarang". 2017. Skripsi, Fakultas Ilmu Komputer.
- [11]. R. Majid, N. Rijati. Implementasi Web Service Pada Aplikasi Jateng. News Dengan Menggunakan Metode REST Implementation of Web Service on Jateng. News App Using REST Method.
- [12]. C. Jones, Integrate, Automate, and Regulate your business process with the best of pact's SOA. Birmingham-Mumbai, PACKT Publishing, 2011