

## PENGEMBANGAN SISTEM INFORMASI PELATIHAN BERBASIS WEB MENGUNAKAN TEKNOLOGI *WEB SERVICE* DAN *FRAMEWORK LARAVEL*

Ramos Somya<sup>1</sup>; Tan Michelle Esmeralda Nathanael<sup>2</sup>

Program Studi S1 Teknik Informatika  
Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana  
www.fti.uksw.edu

<sup>1</sup>ramos.somya@uksw.edu, <sup>2</sup>672015069@student.uksw.edu



Ciptaan disebarluaskan di bawah Lisensi Creative Commons Atribusi-NonKomersial 4.0 Internasional.

**Abstract**—*Pincher ID* is a local startup that engaged in the education sector especially for providing training classes. For now, *Pincher ID* based on a mobile application that causes only a third party (Google Play store/ App Store) who can access it. Furthermore, mobile application *Pincher ID* is not indexed by search engine so it makes difficult for people to find it. In this research, training information system web-based with web service technology and framework *Laravel* will be built. Web service technology applied to the system for data integration so data can be stored centrally. The architecture of web service used in the system is *REST* because accessing *REST* with URL address on browser application, give easy access to the application. In this research, *prototype* method used to develop an information system based on a website that can be used by *Pincher ID*. The test result showed that the training information system worked well and in accordance with the user's wishes.

**Keywords:** *Web Service, RESTful, Laravel, Training Information System*

**Intisari**— *Pincher ID* merupakan *startup* lokal yang bergerak di bidang pendidikan khususnya dalam mengadakan kelas – kelas pelatihan. Untuk saat ini, *Pincher ID* masih berbasis aplikasi *mobile* yang menyebabkan *Pincher ID* hanya bisa diakses melalui perangkat *smartphone* atau *tablet* dengan bantuan pihak ketiga (Google Playstore/ App Store). Selain itu, aplikasi *mobile* *Pincher ID* tidak terindeks oleh mesin pencari sehingga menyulitkan masyarakat dalam menemukan *Pincher ID*. Pada penelitian ini dilakukan pembuatan sistem informasi pelatihan berbasis *website* dengan menggunakan teknologi *web service* dan *Laravel* sebagai *framework*-nya. Teknologi *web service* diterapkan pada sistem tersebut agar adanya integrasi data sehingga data dapat disimpan secara terpusat. Arsitektur *web*

*service* yang digunakan adalah *REST* karena memberikan kemudahan dalam pengaksesan melalui URL pada *browser*. Penelitian ini menggunakan metode *prototype* untuk metode pengembangan aplikasi sehingga menghasilkan sistem informasi berbasis *web* yang dapat digunakan oleh *Pincher ID*. Hasil pengujian sistem menunjukkan bahwa sistem dapat berjalan dengan baik dan sesuai dengan keinginan dari *user*.

**Kata Kunci:** *Web Service, RESTful, Framework Laravel, Sistem Informasi Pelatihan*

### PENDAHULUAN

*Pincher ID* merupakan *startup* lokal yang bertujuan dalam memudahkan pengajar dalam mempromosikan kelas pelatihan serta membuat *online registration* khususnya dalam pelatihan di bidang teknologi. Selain itu, *Pincher ID* memudahkan calon peserta dalam menemukan pelatihan yang menarik. Saat ini *Pincher ID* masih berbentuk aplikasi berbasis *mobile* dengan fitur – fitur yang masih terbatas seperti melihat *list* pelatihan, penjualan tiket pelatihan, dan lain - lain. Dikarenakan *Pincher ID* masih berbentuk aplikasi *mobile* menyebabkan *Pincher ID* hanya bisa diakses melalui perangkat *smartphone* atau *tablet* dengan bantuan pihak ketiga (Google Playstore/ App Store). Aplikasi *mobile* *Pincher ID* juga tidak dapat diindeks oleh mesin pencari sehingga menyulitkan masyarakat dalam menemukan aplikasi *Pincher ID*.

Solusi dari permasalahan tersebut adalah dilakukannya pengembangan aplikasi *Pincher ID* yang semula berbasis *mobile* menjadi *website* dengan menggunakan *web service* dalam pengintegrasian data. Teknologi *web service* digunakan sebagai suatu fasilitas yang menyediakan layanan (dalam bentuk informasi atau data) kepada sistem lain, sehingga dapat berinteraksi dengan sistem tersebut melalui

layanan-layanan yang disediakan walaupun berbeda *platform*, sistem operasi, dan bahasa pemrograman. Selain itu, proses *loading* sebuah *website* juga perlu diperhatikan. Proses *loading* yang lambat akan membuat pengguna meninggalkan *website*. Laravel merupakan *developments tool framework* yang cocok karena Laravel memiliki waktu eksekusi yang sedikit sehingga *website* memiliki proses *loading* yang cepat (Das & Saikia, 2016).

Berdasarkan latar belakang yang telah dijelaskan, didapatkan rumusan masalah dari penelitian ini yaitu bagaimana pengembangan sistem informasi pelatihan berbasis *web* yang sesuai dengan proses bisnis yang ada dalam Pincher ID dengan menggunakan teknologi *web service* dan *framework* Laravel. Adapun batasan – batasan masalah dalam penelitian ini antara lain, mengembangkan sistem informasi pelatihan dengan mengambil studi kasus aplikasi Pincher ID, menggunakan data – data kelas pelatihan yang ada di Pincher ID sampai pada Juni 2018, tidak membahas mengenai keamanan *website* yang dibuat, menggunakan teknologi *Service Oriented Architecture* (SOA) atau lebih dikenal dengan sebutan *Web Service* dengan metode REST/RESTfull dan dibuat menggunakan *framework* Laravel.

## BAHAN DAN METODE

Arianto dalam penelitiannya mengembangkan sistem informasi akademik STT Terpadu Nurul Fikri yang saat ini menggunakan pola *client-server* (*two-tier system*). Tetapi kesulitan timbul pada saat akan dikembangkan, yakni belum tersedianya layanan (*web service*) yang mampu mengintegrasikan sistem tersebut dengan sistem lain sehingga mempersulit pengembangan terutama dalam hal pertukaran, integrasi dan pengelolaan data. Penelitian menggunakan teknologi REST dengan perancangan dua model API untuk dua modul yakni modul mahasiswa dan modul dosen. Pada modul ini akan menghasilkan data dengan format JSON. Teknik pengujian API menggunakan teknik *blackbox testing* dengan *tools* aplikasi pengujian Postman. Pengujian API dilakukan pada *prototype* aplikasi *web service*. Metode penelitian yang digunakan adalah *Unified Process* yang menggunakan kerangka kerja *Yii Framework 2.0* (Arianto, Munir, & Khotimah, 2016).

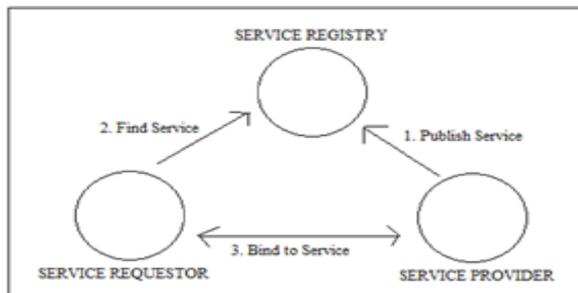
Sedangkan penelitian yang dilakukan Rianto bertujuan untuk mempermudah dan memperlancar pengolahan data pelayanan farmasi di RSUD Banyumas. Kendala yang terdapat di Farmasi RSUD Banyumas, yakni untuk pelayanan resep pasien dengan debitur petugas Farmasi yang

masih harus melakukan entri data pada Sistem Informasi Apotek dan Sistem Informasi Rumah Sakit. Oleh sebab itu, perlu dilakukan perbaikan dalam pengelolaan sebuah sistem informasi dengan menggunakan *web services* sebagai integrasi data. Pemanfaatan teknologi internet telah mewarnai perkembangan sistem *distributed computing* dengan munculnya konsep mengenai *web services*. *Web services* merupakan paradigma baru dalam mengimplementasikan sistem terdistribusi melalui web yang menggunakan basis teknologi XML, dengan *standard* protokol HTTP dan SOAP. Mengingat *standard* terbuka yang digunakan dalam mendukung teknologi *web services*, maka implementasi *web services* menjanjikan kemudahan dan perbaikan dalam mendukung integrasi berbagai *platform* sistem dan aplikasi, baik melalui infrastruktur intranet maupun internet atau ekstranet. Tahapan yang dilakukan mulai dengan *planning, analyst, design, implementation, dan testing and evaluation*, sehingga menghasilkan rancangan model analisis *services* berbasis obyek dan aplikasi *web services* sebagai integrasi data Farmasi di Rumah Sakit Umum Banyumas sehingga tidak diperlukan penggantian sistem yang lama (Riyanto, Utami, & Amborowaty, 2013).

Berdasarkan tinjauan pustaka yang telah diuraikan menunjukkan bahwa *web service* memiliki metode arsitektur berupa REST dan SOAP dengan format data berupa XML dan JSON. Atas pertimbangan dari masing – masing metode berdasarkan penelitian sebelumnya maka pada penelitian ini digunakan arsitektur REST dengan format data JSON. Perbedaan penelitian ini dengan penelitian sebelumnya adalah menggunakan *framework* Laravel dalam pembuatan satu API (*Application Programming Interface*) dan *frontend*. Selain itu, tujuan dan fokus utama dari penelitian ini adalah mengembangkan sistem informasi pelatihan berbasis *web* menggunakan teknologi *web service* untuk integrasi data. Hasil yang diharapkan dari penelitian ini adalah sistem informasi pelatihan yang memudahkan masyarakat dalam mengakses situs dengan performa yang ringan dan cepat serta menggunakan teknologi *web service* untuk integrasi data.

*Web Service* adalah suatu sistem perangkat lunak yang dirancang untuk mendukung *interaction and interoperability* antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang menyediakan layanan (dalam bentuk informasi atau data) kepada sistem lain, sehingga dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan. *Web service* menyimpan data informasi dalam format JSON atau XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda

platform, sistem operasi, dan bahasa pemrograman. Secara umum, arsitektur *web service* memiliki tiga komponen yang terlibat di dalamnya, yaitu *service provider*, *service registry* dan *service requestor*. (1) *Service Provider* berfungsi untuk menyediakan layanan/*service* dan mengolah sebuah *registry* agar layanan – layanan tersebut dapat tersedia; (2) *Service Registry* berfungsi sebagai lokal sentral yang mendeskripsikan semua layanan / *service* yang telah didaftarkan; (3) *Service Requestor* berfungsi untuk meminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut (Tidwell, 2000).



Sumber: (Tidwell, 2000)  
Gambar 1. Arsitektur Web Service

REST adalah gaya arsitektural yang memiliki aturan seperti antar muka yang seragam, sehingga jika aturan tersebut diterapkan pada *web services* akan dapat memaksimalkan kinerja *web services* terutama pada performa, skalabilitas, dan kemudahan untuk dimodifikasi. Pada arsitektur REST data dan fungsi dianggap sebagai sumber daya yang dapat diakses lewat *Uniform Resource Identifier* (URI), biasanya berupa tautan pada *web*. REST menggunakan protokol HTTP yang bersifat *stateless*. Perintah HTTP yang bisa digunakan adalah fungsi *GET*, *POST*, *PUT* atau *DELETE*. Hasil yang dikirimkan dari *server* biasanya dalam bentuk format XML atau JSON sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing di sisi *client*. Dalam penerapannya, REST lebih digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada sumber daya adalah orientasi yang menyediakan sumber daya sebagai layanannya dan bukan kumpulan-kumpulan dari aktifitas yang mengolah sumber daya itu (Kurniawan, 2014).

JavaScript Object Notation (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis, serta mudah diterjemahkan dan dibuat (*generate*). Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript. JSON merupakan format teks yang tidak tergantung pada bahasa pemrograman apapun karena menggunakan bahasa yang umum digunakan oleh

*programmer*. Oleh karena itu sifat-sifat tersebut, JSON ideal sebagai bahasa pertukaran data. JSON memiliki beberapa kelebihan dibandingkan dengan XML, kelebihan-kelebihan antara lain: (1) Format penulisan JSON relatif lebih terstruktur dan mudah; (2) Ukuran karakter yang dibutuhkan JSON lebih kecil dibandingkan XML untuk data yang sama sehingga kecepatan pertukaran data JSON lebih unggul dibandingkan XML; (3) Proses *browser parsing* JSON lebih sederhana dibandingkan pada XML karena JSON menggunakan *function* JavaScript *eval()* untuk melakukan parsing sedangkan dokumen XML diparsing oleh XML *HttpRequest* (Yusrizal, Dawood & Roslidar, 2017).

Laravel adalah *web framework* PHP yang bersifat *open source* dan gratis yang dibuat oleh Taylor Otweel yang dapat digunakan dalam mengembangkan *web applications* dengan menggunakan arsitektur MVC (*Model-View-Controller*). *Framework* Laravel mudah dipahami dan memudahkan dalam hal *authentication*, *routing*, *session manager*, *caching*, dan beberapa kegunaan lain dari komponen – komponen di Laravel. Laravel juga menyediakan fitur seperti *database migration* dan integrasi *unit testing support* yang memudahkan *developer* untuk membangun aplikasi yang kompleks (Chen et al, 2017).

**METODE PENELITIAN**

Penelitian ini dilakukan dan diselesaikan melalui 5 tahapan penelitian yaitu : (1) pengumpulan data dan analisis kebutuhan sistem; (2) desain perancangan sistem; (3) implementasi rancangan sistem; (4) pengujian sistem dan analisis hasil pengujian; dan (5) penulisan laporan hasil penelitian (Hasibuan, 2007). Tahapan - tahapan yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 2.



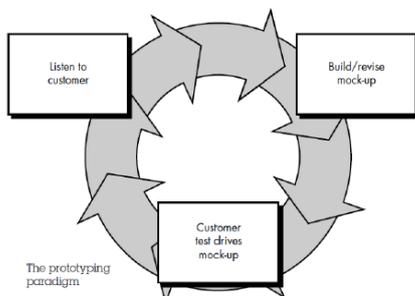
Sumber: (Somya dan Nathanael, 2019)  
Gambar 2. Tahapan Penelitian

Tahap pertama yang dilakukan adalah melakukan identifikasi dan mengumpulkan data –

data yang dibutuhkan. Pada tahap ini metode yang dilakukan yaitu wawancara dan studi literatur. Wawancara dilakukan dengan kepala tim Pincher ID. Data yang didapatkan pada penelitian ini berupa data mengenai kelas – kelas pelatihan yang diambil dari Pincher ID sebagai data primer dan data yang diambil melalui studi literatur maupun dokumen yang membahas mengenai *web service*. Data – data yang didapat diolah dan dianalisis sehingga ditemukan beberapa kelemahan pada sistem yang sedang berjalan.

Tahap kedua dilakukan desain perancangan sistem berdasarkan hasil analisis kebutuhan menggunakan UML (*Unified Modelling Language*) yang terdiri dari *use case diagram* dan *activity diagram*. *Unified Modeling Language* (UML) merupakan sebuah standar dalam industri dengan tujuan untuk merancang, visualisasi dan mendokumentasikan suatu sistem perangkat lunak. UML memberikan tawaran akan sebuah standar untuk merancang model sebuah sistem. UML dapat digunakan untuk pemodelan pada semua jenis aplikasi piranti lunak, di mana aplikasi tersebut dapat berjalan pada piranti keras, dalam jaringan dan sistem operasi apapun, serta dapat ditulis dalam bahasa pemrograman apapun (Sulistyorini, 2009). Tambahan perancangan lainnya berupa perancangan database dan perancangan *interface*.

Tahap ketiga dilakukan implementasi rancangan sistem menggunakan *framework* Laravel baik *frontend* maupun *backend*, Bootstrap sebagai desain tampilan, dan JQuery sebagai komponen. Metode yang digunakan dalam pengembangan aplikasi adalah metode *prototype*. *Prototype* merupakan metode yang dikembangkan dengan berkomunikasi kepada *client* agar aplikasi yang dihasilkan sesuai dengan keinginan *client*. Metode *prototype* juga mempermudah pengembang perangkat lunak dalam membentuk model dari perangkat lunak yang harus dibuat.



Sumber: (Pressman, 2010)

Gambar 3. Metode *prototype*

Tahap keempat dilakukan pengujian aplikasi dengan menggunakan pengujian beta serta analisis hasil pengujian sistem. Pengujian alfa merupakan

pengujian yang dilakukan pada sisi pengembang (Pressman, 2010). Dalam kasus ini pengujian alfa dilakukan menggunakan teknik *black box testing* yang dapat didefinisikan suatu pengujian untuk mengetahui apakah semua perangkat lunak telah berjalan dengan semestinya sesuai dengan kebutuhan fungsional yang didefinisikan. Pengujian beta adalah pengujian yang dilakukan oleh *user* untuk menguji apakah aplikasi sudah sesuai atau belum terhadap keinginan *user* (Purnomo dan Beeh, 2016).

Tahap kelima dilakukan penulisan laporan hasil penelitian. Tahap ini dilakukan dokumentasi proses dari tahap awal sampai tahap akhir. Laporan hasil penelitian ditulis dalam bentuk tulisan ilmiah.

Analisis kebutuhan perangkat keras dan spesifikasi perangkat lunak yang digunakan dalam membangun penelitian ini yaitu: analisis perangkat keras yang akan digunakan adalah prosesor komputer MacOS, 2 GB RAM, 2 GB VGA serta Hard Disk 250GB. Perangkat lunak yang digunakan adalah sistem operasi Linux, MySQL, XAMPP Server, Sublime Text 3, Google Chrome, Postman, serta Microsoft Visio 2010 untuk membuat UML.



Sumber: (Somya dan Nathanael, 2019)

Gambar 4. Arsitektur Sistem Pincher ID

Arsitektur sistem mendefinisikan dan menggambarkan komponen – komponen yang lebih spesifik secara terstruktur dengan tujuan agar struktur yang dirancang sesuai dengan kebutuhan dan tujuan sistem (Widiono dan Pujiyanta, 2014). Gambar 4 merupakan perancangan arsitektur sistem Pincher ID. Secara umum, arsitektur sistem terbagi menjadi 3 bagian utama yaitu : *client*, *web service*, dan *database*. *Web service* menyediakan fungsi – fungsi yang dapat diakses oleh *client* sedangkan *client* melakukan *request* kepada *web service* untuk mendapatkan data yang dibutuhkan. Database merupakan tempat penyimpanan data di mana pada penelitian ini baik *client* maupun *web service* menyimpan data di *database* yang sama.

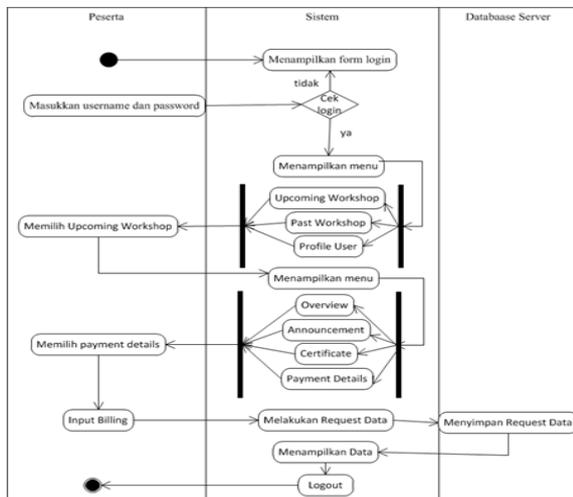
Perancangan sistem dalam penelitian ini dibuat dalam bentuk *diagram* UML. Diagram UML meliputi *use case diagram*, dan *activity diagram*.

Berikut ini akan dijelaskan masing-masing diagram yang telah dibuat.



Sumber: (Somya dan Nathanael, 2019)  
Gambar 5. Use Case Diagram Sistem

Pada Gambar 5 merupakan use case yang menjelaskan bagaimana sistem ini berjalan. Sistem ini memiliki 3 aktor yang berperan dalam sistem yang dibuat yakni peserta, educator, dan admin. Peserta merupakan orang yang telah mendaftarkan diri dalam sebuah kelas pelatihan yang tersedia di sistem. Peserta memiliki hak akses seperti melihat kelas pelatihan, mendaftarkan diri, membayar kelas pelatihan, mengunduh sertifikat, dan melihat informasi terbaru mengenai kelas pelatihan yang telah didaftarkan. Educator merupakan member dari Pincher ID yang telah terdaftar sebagai tenaga pengajar. Educator mengatur hal-hal yang berkaitan dengan pembuatan kelas pelatihan seperti tiket, pengumuman, sertifikat, data educator, foto yang berkaitan dengan kelas pelatihan, dan jadwal pelatihan. Admin merupakan staff IT dari Pincher ID yang bertanggung jawab penuh pada keseluruhan kerja sistem.



Sumber: (Somya dan Nathanael, 2019)  
Gambar 6. Activity Diagram Pembayaran Tiket Peserta

Gambar 6 merupakan activity diagram pembayaran tiket oleh peserta yang dapat dijelaskan bahwa peserta harus terlebih dahulu melakukan login. Apabila login berhasil maka sistem akan menampilkan menu yang terdiri dari *Upcoming Workshop*, *Past Workshop*, dan *Profile User*. Dari ketiga menu tersebut, peserta memilih menu *Upcoming Workshop*. Sistem menampilkan 4 menu yakni *overview*, *announcement*, *certificate*, dan *payment details*. Peserta memilih menu *payment details* dan menginputkan *billing* pembayaran. Sistem mengirimkan data ke *database* dan *database* menyimpannya. Sistem menampilkan data sesuai dengan data yang telah diinputkan. Dalam menu utama terdapat menu *logout* yang digunakan untuk *destroy session* apabila peserta selesai melakukan aktivitas dan akan kembali ke halaman *login*.

### HASIL DAN PEMBAHASAN

Dalam penelitian ini menggunakan data kelas pelatihan dari Pincher ID sampai pada Juni 2018. Struktur aplikasi yang dibuat terdiri dari dua bagian yaitu sisi client dan sisi *web service*. Aplikasi dibangun menggunakan bahasa pemrograman PHP dengan *framework* Laravel versi 5.5 dengan pendekatan arsitektur REST (*Representational State Transfer*). Langkah pertama yang dilakukan dalam pembuatan *web service* adalah membangun *service provider* pada *function viewListEvent*.

```

1. public function viewListEvent() {
2.     $array_upcoming = array();
3.     //get all event yang telah dipublish
4.     $listallevent = Event::where([
5.         ['deleted_at', null],
6.         ['is_publish', 1],
7.     ])->get();
8.     //looping hasil query
9.     foreach ($listallevent as $event) {
10.         if ($event['type'] == 'Public' && $event['is_verify'] > 0) {
11.             //show upcoming event
12.             $upcoming = EventDetail::where([
13.                 ['event_id', $event['event_id']],
14.                 ['date_start', '>=', Carbon::parse(now())->format('Y-m-d')],
15.             ])->first();
16.             if (isset($upcoming)) {
17.                 array_push($array_upcoming, $event);
18.             }
19.         }
20.     }
21.     return Response::json(array(array(
22.         'res' => true,
23.         'data' => array('upcoming' => $array_upcoming, true)),
24.     )), 200);
25. }

```

Sumber: (Somya dan Nathanael, 2019)  
Gambar 7. Fungsi Web Service viewListEvent

Gambar 7 merupakan fungsi yang digunakan untuk menampilkan list event. Hasil *query* dimasukkan dalam *looping* dan dilakukan pengecekan apakah tanggal dimulainya acara telah lewat atau belum, jika belum maka data akan ditampung dalam array. Pada pembuatan REST dengan menggunakan *framework* Laravel diperlukan penambahan library bernama

GuzzleHttp yang digunakan untuk membuat memanggil fungsi Client() sehingga *web service* dapat memberi respon dengan mengirim data ke dalam format JSON.

```

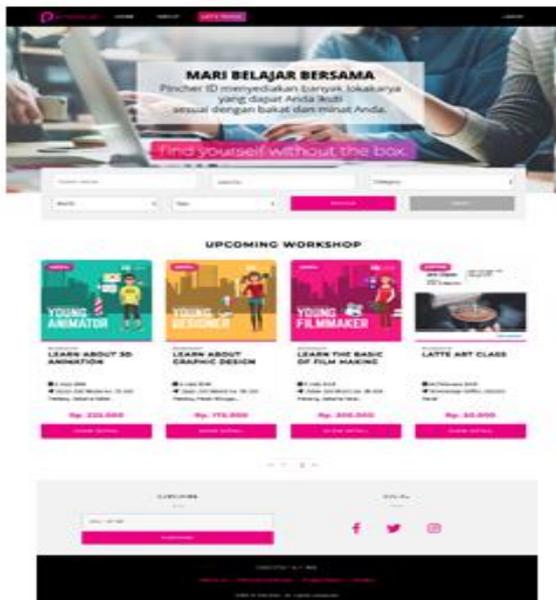
1. $client      = new Client();
2. $urlactivity = new UrlActivity();
3. $new_url     = $urlactivity->getUrl();
4. $url        = $new_url.'viewlistevent';
5. $response   = $client->get($url);
6. $event      = json_decode($response->getBody());

```

Sumber: (Somya dan Nathanael, 2019)

Gambar 8. Request Data Event

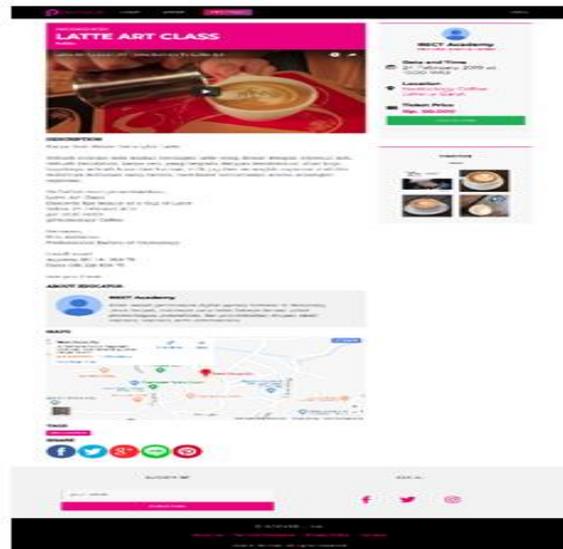
Gambar 8 merupakan potongan program pada sisi client yang bertujuan untuk melakukan *request data* ke *web service* dengan memasukkan parameter URL yang mengacu pada URL lokasi server dengan menambahkan parameter *viewlistevent* menggunakan method GET. Penelitian ini menggunakan *server local* sehingga pemanggilan dari *request data* tersebut menggunakan URL `http://localhost/event-pincher/event-pincher-  
ws/public/api/viewlistevent`. Data yang didapat dari pemanggilan URL dalam format JSON sehingga perlu di-*decode* dengan fungsi *json\_decode* dan ditampilkan pada sisi *client* seperti Gambar 9.



Sumber: (Somya dan Nathanael, 2019)

Gambar 9. Halaman Utama Aplikasi

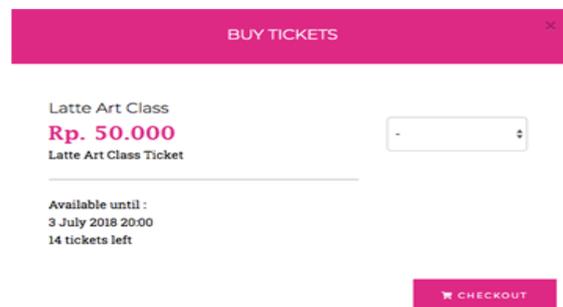
Gambar 9 merupakan halaman utama yang akan tampil ketika peserta mengakses sistem. Halaman ini berisi mengenai event – event terbaru yang ada di sistem seperti poster promosi, nama *event*, tanggal & tempat pelaksanaan serta harga tiket. Halaman ini juga memuat kolom pencarian yang memudahkan peserta dalam mencari *event*.



Sumber: (Somya dan Nathanael, 2019)

Gambar 10. Halaman Detail Event

Gambar 10 merupakan halaman detail *event* yang berisi informasi mengenai nama *event*, deskripsi acara, tempat & waktu pelaksanaan, informasi tiket & pembuat kelas, peta lokasi, dan foto promosi event. Apabila peserta ingin membeli tiket maka, peserta dapat menekan tombol *order now* yang akan menampilkan *pop up* seperti Gambar 11.



Sumber: (Somya dan Nathanael, 2019)

Gambar 11. Tampilan Pembelian Tiket

Gambar 11 merupakan *pop up* berisi informasi spesifik mengenai tiket *event* seperti nama tiket, harga, deskripsi tiket, tanggal tersedia, dan sisa tiket. Apabila peserta ingin membeli tiket, peserta dapat memasukkan jumlah tiket yang akan dibeli dengan menggunakan *drop down* menu. Selanjutnya, peserta menekan tombol *checkout* dan akan masuk ke halaman *login* apabila peserta belum melakukan *login* seperti Gambar 12. Apabila sebelumnya peserta telah melakukan *login* maka peserta akan langsung dialihkan ke halaman *checkout* seperti Gambar 13.



Sumber: (Somya dan Nathanael, 2019)  
Gambar 12. Tampilan Login

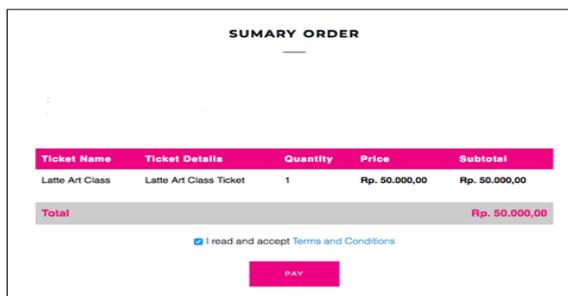
Gambar 12 merupakan halaman *login* di mana peserta diminta untuk memasukkan *username* dan *password*. Sistem akan memproses masukan dari peserta, apabila benar maka sistem akan mengaktifkan *session* dan mengalihkan ke halaman *checkout* seperti Gambar 14.

```

1. public function checkoutTicket(Request $request)
2. {
3.     $request->session()->put('ticket_event', array($array));
4.     $client = new Client();
5.     $urlActivity = new UrlActivity();
6.     $new_url = $urlActivity->getUrl();
7.     $url = $new_url . 'ticketcheckout';
8.     $response = $client->post($url($url), [
9.         'form_params' => ['ticketid_quantity' => $array],
10.    ]);
11.    $ticket = json decode($response->getBody());
12.    return view('ticket', array('ticket' => $ticket[0]->data, 27.
13.    'event' => $event[0]->data[0], 'subtotal' => $subtotal,
14.    'total' => $total));
14. }
    
```

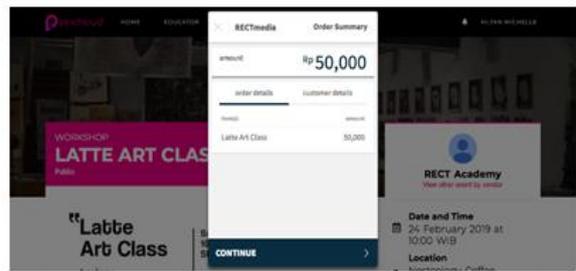
Sumber: (Somya dan Nathanael, 2019)  
Gambar 13. Perintah Pemesanan Tiket Peserta

Gambar 13 berisi barisan program yang digunakan untuk memanggil *web service* dengan mengirimkan array yang telah diolah tersebut ke *web service*. *Web service* akan memproses ke *database* dan mengembalikan hasil data. Hasil data kemudian ditampilkan dalam *view ticket* seperti Gambar 14.



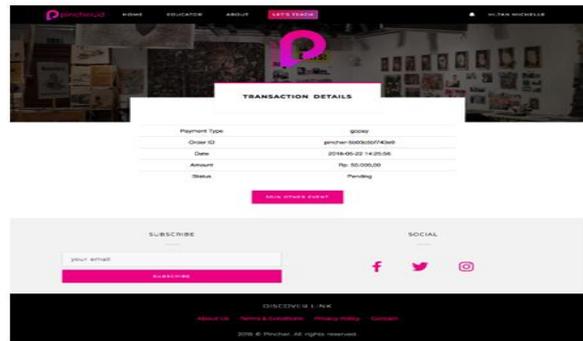
Sumber: (Somya dan Nathanael, 2019)  
Gambar 14. Tampilan Checkout Tiket

Gambar 14 merupakan halaman yang memuat rincian pembayaran yang akan dibayar oleh peserta seperti nama tiket, rincian tiket, jumlah, harga, subtotal, dan total pembayaran. Apabila peserta sudah yakin untuk membeli tiket tersebut, maka peserta dapat menekan tombol *pay* dan sistem akan menampilkan *pop up* pembayaran seperti Gambar 15.



Sumber: (Somya dan Nathanael, 2019)  
Gambar 15. Tampilan Pembayaran

Gambar 15 merupakan halaman yang digunakan peserta untuk melakukan pembayaran dengan pilihan pembayaran yang ada seperti VISA, Master Card, BCA, BNI, Mandiri, Alto, Prima, Gopay, dan lain-lain. Pembayaran dapat dilakukan sesuai dengan petunjuk yang telah tertera di masing – masing metode pembayaran. Peserta dapat menyelesaikan pembayaran paling lama 6 jam setelah melakukan pemesanan tiket, apabila lebih dari itu maka tiket yang telah dipesan akan hangus dan peserta harus mengulang kembali dari awal. Apabila peserta sudah menentukan metode pembayaran, maka akan dialihkan ke halaman *finish* seperti Gambar 16.



Sumber: (Somya dan Nathanael, 2019)  
Gambar 16. Tampilan Hasil Proses Pembayaran

Gambar 16 merupakan halaman yang memuat rincian hasil pembayaran yang telah dibayar oleh peserta. Halaman ini berisi informasi mengenai metode pembayaran, *order id* yang didapat, tanggal pembayaran, total pembayaran, dan status pembayaran. Status pembayaran akan tertulis *completed* apabila peserta telah benar – benar selesai melakukan pembayaran. Peserta dapat melakukan penyelesaian pembayaran sesuai dengan ketentuan dan langkah - langkah yang ada.

*User Acceptance Testing* adalah pengujian yang dilakukan oleh *user* untuk menguji apakah aplikasi sudah sesuai atau belum terhadap keinginan *user* (Purnomo dan Beeh, 2016). Pengujian ini dilakukan terhadap pengguna melalui wawancara dimana pada tahap ini dilakukan wawancara oleh RECT Academy selaku *user* dari

Pincher ID. Berdasarkan wawancara yang dilakukan disimpulkan bahwa sistem yang telah dibuat mudah digunakan (*user friendly*), ringan dan cepat, serta sesuai dengan tujuan dan manfaat dari penelitian yang dilakukan. Kemudahan yang ditawarkan pada sistem ini terlihat dari tampilan awal sistem yang menampilkan poster kelas pelatihan terbaru dan navigasi yang mudah ditemukan dan digunakan oleh *user*. Selain itu, sistem yang dibuat dapat dijalankan disemua perangkat dengan berbagai macam ukuran layar. Sistem ini membantu Pincher ID dalam menyelenggarakan kelas pelatihan.

### KESIMPULAN

Berdasarkan penelitian dan pengujian yang dilakukan maka dapat disimpulkan bahwa Sistem Informasi Pelatihan dengan menggunakan teknologi *web service* berbasis *website* pada Pincher ID dapat berjalan dengan baik dengan hasil pengujian dari sistem dinyatakan layak dan secara fungsional sistem dapat menghasilkan *output* yang diharapkan. Penerapan teknologi *web service* pada fungsi *viewlistevent* dan *ticketcheckout* pada sistem ini mempermudah dalam proses integrasi data yang dapat dilakukan secara terpusat. Selain itu, penggunaan *framework* Laravel bermanfaat karena membuat sistem memiliki performa yang cepat dalam memproses data serta ringan untuk dijalankan.

### REFERENSI

- Arianto, M. A., Munir, S., dan Khotimah, K. (2016). Analisis Dan Perancangan Representational State Transfer (Rest) Web Service Sistem Informasi Akademik Stt Terpadu Nurul Fikri Menggunakan Yii Framework. *Jurnal Teknologi Terpadu*, 2(2). Retrieved from <http://journal.nurulfikri.ac.id/index.php/JTT/article/view/57>
- Chen, X., Ji, Z., Fan, Y., dan Zhan, Y. (2017). Restful API Architecture Based on Laravel Framework. In *IOP Conf. Series: Journal of Physics: Conf. Series 910 (2017)* (pp. 1-7). <https://doi.org/10.1088/1742-6596/910/1/012016>
- Das, R., dan Saikia, L. P. (2016). NComparison of Procedural PHP with Codeigniter and Laravel Frameworko Title. *International Journal of Current Trends in Engineering & Research (IJCTER)*, 2(6), 42-48. Retrieved from <https://www.ijcter.com/papers/volume-2/issue-6/comparison-of-procedural-php-with-codeigniter-and-laravel-framework.pdf>
- Depkes. (2018). Pemagangan, Cara Pemerintah Kebut Peningkatan Daya Saing Tenaga Kerja. Retrieved from [www.depkes.go.id/article/view/16122300001/pemagangan-cara-pemerintah-kebut-peningkatan-daya-saing-tenaga-kerja.html](http://www.depkes.go.id/article/view/16122300001/pemagangan-cara-pemerintah-kebut-peningkatan-daya-saing-tenaga-kerja.html)
- Erick Kurniawan. (2014). IMPLEMENTASI REST WEB SERVICE UNTUK SALES ORDER DAN SALES TRACKING BERBASIS MOBILE. *Jurnal EKSIS*, 7(1), 1-12. Retrieved from <https://media.neliti.com/media/publications/78853-ID-implementasi-rest-web-service-untuk-sale.pdf>
- Hasibuan, Z. A. (2007). *Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi*. Pressman, R. (2010). *Software Engineering: A Practitioner's Approach, Seventh Edition*.
- Purnomo, R., dan Beeh, Y. R. (2016). *Perancangan Aplikasi Emergency Button berbasis. Android*.
- Riyanto, Utami, E., dan Amborowaty, A. (2013). PEMANFAATAN WEB SERVICES PADA INTEGRASI DATA FARMASI DI RSU BANYUMAS. In *Seminar Nasional Informatika 2013 (semnasIF 2013)* (pp. 276-283). Retrieved from <http://jurnal.upnyk.ac.id/index.php/semnasif/article/view/881>
- Somya, R. dan Nathanael, T. M. E. (2019). Pengembangan Sistem Informasi Pelatihan Berbasis Web Menggunakan Teknologi Web Service Dan Framework Laravel.
- Sulistiyorini, P. (2009). Pemodelan Visual dengan Menggunakan UML dan Rational Rose. *Jurnal Teknologi Informasi DINAMIK*, 14(1), 23-29.
- Tidwell, D. (2000). *Web Services: The Web's next Revolution*.
- Widiono, R., dan Pujiyanta, A. (2014). Sistem Informasi Rute Terpendek Tour Travelling Daerah Istimewa Yogyakarta Dengan Algoritma Genetika. *Jurnal Sarjana Teknik Informatika*, 2(3), 181-192. Retrieved from <http://journal.uad.ac.id/index.php/JSTIF/article/view/2840>
- Yoga Kresna Permana, Ni Nyoman Yuliantini, I. W. B. (2016). Analisis Dampak Pelatihan Terhadap Prestasi Kerja Karyawan. *Jurusan Manajemen*, 4. Retrieved from <https://ejournal.undiksha.ac.id/index.php/JJM/article/view/6733>
- Yusrizal, Dawood, R., dan R. (2017). Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter. *Jurnal Online Teknik Elektro*, 2(1), 1-8. Retrieved from <http://www.jurnal.unsyiah.ac.id/kitekro/article/view/6803/5571>