

PENERAPAN RESTFUL WEB SERVICE DENGAN FRAMEWORK LARAVEL UNTUK PEMBANGUNAN SISTEM INFORMASI MANAJEMEN SUMBER DAYA MANUSIA

Okky Dwi Arianto¹⁾, Yerymia Alfa Susetyo²⁾

^{1, 2)}Fakultas Teknik Informatika, Universitas Kristen Satya Wacana
Jl. Diponegoro 52-60, Salatiga 50771, Indonesia
e-mail: 672018321@student.uksw.edu¹⁾, yerymia.alfa@uksw.edu²⁾

ABSTRAK

Manajemen sumber daya manusia sebagai penunjang pengelola data ketenagakerjaan perlu memiliki inovasi teknologi untuk pengelolaannya. Suatu organisasi yang mengimplementasikan sistem informasi menggunakan aplikasi perangkat lunak akan dapat meningkatkan efektifitas dan efisiensi. Dengan bantuan sistem informasi maka kinerja manajemen akan optimal. Pengelolaan data organisasi akan terpusat sehingga kemungkinan akan terjadi redundancy data relatif kecil. Selain itu pengolahan dan penyajian data tidak membutuhkan waktu yang lama. Tujuan dari penelitian ini, untuk menerapkan RESTful web service pada sistem informasi manajemen sumber daya manusia. RESTful web service merupakan teknologi yang dapat digunakan untuk melakukan integrasi data. RESTful web service dibangun menggunakan arsitektur REST dengan menggunakan framework Laravel. Arsitektur REST pada penelitian ini mendukung keamanan pertukaran informasi yang mengkolaborasikan Library Laravel Passport. Sebagai hasilnya, berbagai fungsi pengolahan data sumber daya manusia dalam bentuk RESTful web service dengan return value dalam format JSON. JSON diterapkan pada sistem informasi manajemen sumber daya manusia berbasis framework Laravel yang dapat digunakan berbagai jenis user berbeda.

Kata Kunci: Framework, Laravel, Restful web service, Sumber Daya Manusia

ABSTRACT

Human resource management as a supporter of labor data management needs to have technological innovations for its management. An organization that implements information systems using software applications will be able to increase effectiveness and efficiency. With the help of information systems, management performance will be optimal. The organization's data management will be centralized so that the possibility of data redundancy is relatively small. In addition, processing and presenting data does not require a long time. The purpose of this research is to implement a RESTful web service on a human resource management information system. RESTful web service is a technology that can be used to perform data integration. RESTful web service is built using the REST architecture using the Laravel framework. The REST architecture in this study supports the security of information exchange in collaboration with the Laravel Passport Library. As a result, various human resource data processing functions are in the form of RESTful web services with return values in JSON format. JSON is applied to a human resource management information system based on the Laravel framework that can be used by different types of users.

Keywords: Framework, Human resource, Laravel, RESTful web service,

I. PENDAHULUAN

Sumber daya manusia memiliki peran penting dalam sebuah organisasi salah satunya menunjang aktivitas organisasi untuk mencapai tujuan yang telah ditetapkan. Dalam bidang manajemen, dijelaskan bahwa sumber daya manusia merupakan salah satu unsur dalam organisasi. SDM adalah sumber daya manusia yang perlu penanganan manajemen secara efisien dan efektif. Pada sumber daya manusia, jumlah informasi yang perlu dikelola menimbulkan tingkat kompleksitas dan memerlukan integrasi data yang tinggi [1].

Pesatnya perkembangan teknologi informasi menawarkan solusi pada pengelolaan organisasi yang dapat melakukan proses simpan, ubah, hapus dan mengirimkan informasi secara digital. Sistem informasi bekerja dengan sangat teratur didasarkan pada pola pengelolaan yang telah ditentukan guna memberikan informasi yang dibutuhkan [2].

Sistem informasi manajemen sumber daya manusia disebut sebagai sistem yang bermanfaat dalam membuat serta menyebarkan data tentang sumber daya manusia yang mempunyai keselarasan dengan arah strategis pada suatu organisasi. Sistem informasi dibuat untuk membantu proses-proses didalam organisasi, dimana mencakup proses pendataan, pengelolaan dan laporan [3]. Berbagai jenis unit dan layanan dalam organisasi yang ada pada saat ini, masih dikelola secara konvensional sehingga pengolahan data dan informasi masih dilakukan secara desentralisasi data. Hal tersebut tentunya membuat pengelolaan dan pengaksesan data menjadi sulit [4].

Desentralisasi mengakibatkan sistem informasi tersebut berjalan sendiri-sendiri menggunakan data yang terpisah antar unit atau layanan dalam organisasi [5]. Pengelolaan data dan informasi sangat penting dalam suatu sistem informasi yang kompleks dengan berbagai jenis unit dan layanan. Semakin banyak jenis unit dan layanan maka membutuhkan sentralisasi data yang baik [6].

Integrasi data menjadi salah satu cara untuk menangani masalah diatas. Integrasi data dapat direalisasikan dengan banyak cara. Salah satu cara yang bisa digunakan untuk merealisasikan yaitu membuat sebuah *web service*. Layanan *web service* memiliki fitur pengumpulan dan integrasi data yang dapat berinteraksi antar perangkat lunak. Dengan penerapan arsitektur REST (*Representational State Transfer*) pada layanan *website* berbasis *web service* memberi kemudahan untuk mengembangkan sistem informasi di unit dan departemen lainnya [7].

REST adalah arsitektur layanan web dimana *client* membuat permintaan ke *server*, *server* memproses permintaan dan mengembalikan *response*. Saat menggunakan teknologi REST, sistem informasi didukung oleh API (*Application Programming Interface*) berbasis *website*. *RESTful web service* merupakan nama lain untuk sistem informasi yang menerapkan arsitektur REST [8].

Dari permasalahan yang ada, maka menerapkan sistem informasi manajemen sumber daya manusia menggunakan *RESTful web service* dapat memberi kemudahan dalam integrasi data. Melalui *request* yang dilakukan *client* ke *server* dengan protokol HTTP dan *response* yang diberikan oleh *server* dalam bentuk JSON. Sehingga data dan informasi akan lebih terintegrasi dan tidak tersimpan di masing-masing sistem informasi unit dan departemen yang berbeda. Oleh karena itu, perlu dilakukan penerapan *RESTful web service* pada sistem informasi manajemen sumber daya manusia.

Terdapat beberapa penelitian terdahulu dibidang ini. Penelitian dengan judul Penerapan *Framework Laravel* Pada Aplikasi HRIS (*Human Resource Information System*) membahas tentang aplikasi HRIS yang menangani masalah berupa pengelolaan data, absensi dan laporan keuangan. Pada penelitian terdahulu, aplikasi HRIS sangat membantu PT Octagon Studio Ltd dalam proses pelaporan keuangan dan pengelolaan data karyawan. Dari keberhasilan penelitian terdahulu, maka dibuat penelitian ini yang nantinya tidak hanya membantu pada pengelolaan data pegawai, namun juga membantu dalam penanganan integrasi data antar sistem informasi. Terdapat kesamaan penelitian ini dengan penelitian terdahulu yaitu sama-sama menggunakan *framework Laravel* dan *MYSQL* sebagai database. penelitian terdahulu belum menggunakan teknologi *RESTful web service* sedangkan penelitian ini dibuat dengan menggunakan teknologi *RESTful web service* [1].

Penelitian lain dengan judul Implementasi Teknologi *RESTful web service* dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya) membahas tentang perangkat lunak untuk merekam prestasi mahasiswa di Fakultas Teknologi Pertanian Universitas Brawijaya yang dibangun berbasis *website* dengan menerapkan *RESTful web service*. Penelitian ini memiliki kesamaan teknologi dengan penelitian terdahulu yaitu sama-sama menggunakan teknologi *RESTful web service*. Pada penelitian terdahulu penerapan aplikasi REST yang bersifat *stateless* memberi konsep yang ringan dan sederhana. Penelitian ini berhasil menerapkan teknologi *RESTful web service* menggunakan *framework Laravel* dan bahasa pemrograman PHP. Perbedaan penelitian terdahulu dengan penelitian ini yaitu kasus yang berbeda, yaitu sistem informasi manajemen sumber daya manusia [8].

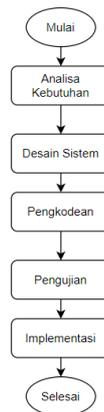
Selanjutnya penelitian yang berjudul *RESTful web service* Untuk Sistem Pencatatan transaksi Studi Kasus PT. XYZ membahas tentang sistem pencatatan penjualan pada lingkungan bisnis yang terdapat beberapa unit dalam perusahaan. Dalam seharinya, terdapat transaksi-transaksi yang banyak yang mempengaruhi jumlah stok, pencatatan dan laporan transaksi. Proses tersebut berdampak pada penyesuaian antara beberapa unit. Pada penelitian terdahulu sistem pencatatan dibuat menggunakan teknologi *RESTful web service* ini sangat membantu mengatasi masalah. Dari keberhasilan penelitian terdahulu, maka dibuat penelitian ini yang dapat mengatasi beberapa unit dan departemen dalam kasus sistem informasi manajemen sumber daya manusia dengan menerapkan *RESTful web service* [9].

II. METODE PENELITIAN

Sistem informasi manajemen sumber daya manusia akan mengelola atribut-atribut data di setiap bagian jenis unit dan departemen yang berbeda. Tantangan dalam implementasi sistem ini adalah bagaimana mengelola data agar mudah di akses oleh berbagai jenis layanan dan user di sistem informasi manajemen sumber daya manusia. Penerapan melalui *web service* menggunakan metode REST API merupakan solusi untuk membagi jenis layanan menjadi bagian yang lebih kecil dan mengintegrasikan data menjadi satu *service*.

Metode penelitian pada penelitian ini menggunakan metode *Waterfall*. Metode *waterfall* adalah model penelitian yang sistematis dan sekuensial dimulai dari tahapan pertama secara berurutan sampai dengan tahap akhir. Tahapan tidak bisa diulang dan tahapan selanjutnya akan dilakukan jika tahapan sebelumnya telah selesai dilakukan [10].

Tahapan-tahapan yang dilakukan pada penelitian ini dapat di gambarkan pada Gambar 1.

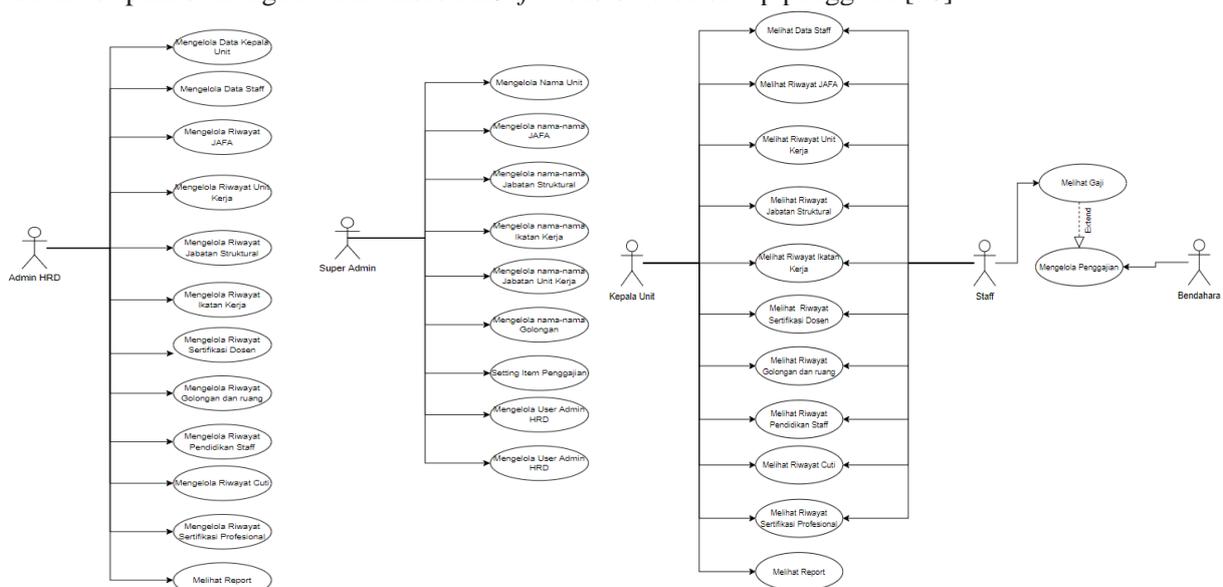


Gambar 1. Tahapan penelitian

Dari Gambar 1 dapat dijelaskan sebagai berikut, Tahapan pertama adalah melakukan analisa kebutuhan terkait pengumpulan data yang digunakan sistem informasi manajemen sumber daya manusia dan perancangan sistem. Tahap kedua adalah melakukan desain perancangan sistem menggunakan *Unified Modeling Language (UML)*. Dimulai dari penentuan *actor-actor*, fungsi-fungsi pada sistem, *class-class* dan *method* yang akan digunakan. Tahap ketiga adalah melakukan pengkodean berdasarkan desain rancangan sistem yang telah dibuat sebelumnya. Pada tahap ini juga akan dibangun *RESTful web service* dengan menggunakan *framework Laravel*. Tahap keempat adalah pengujian terhadap RESTful yang akan di implementasi ke sistem menggunakan Postman. Tahap terakhir adalah melakukan implementasi dan pengujian menggunakan metode *black box* pada sistem informasi manajemen sumber daya manusia. *User Interface (UI)* sistem informasi di implementasi menggunakan *Framework Bootstrap*. Selain itu, *Framework Laravel* dan *Javascript* juga digunakan untuk pemanggilan *RESTful web service*.

III. HASIL DAN PEMBAHASAN

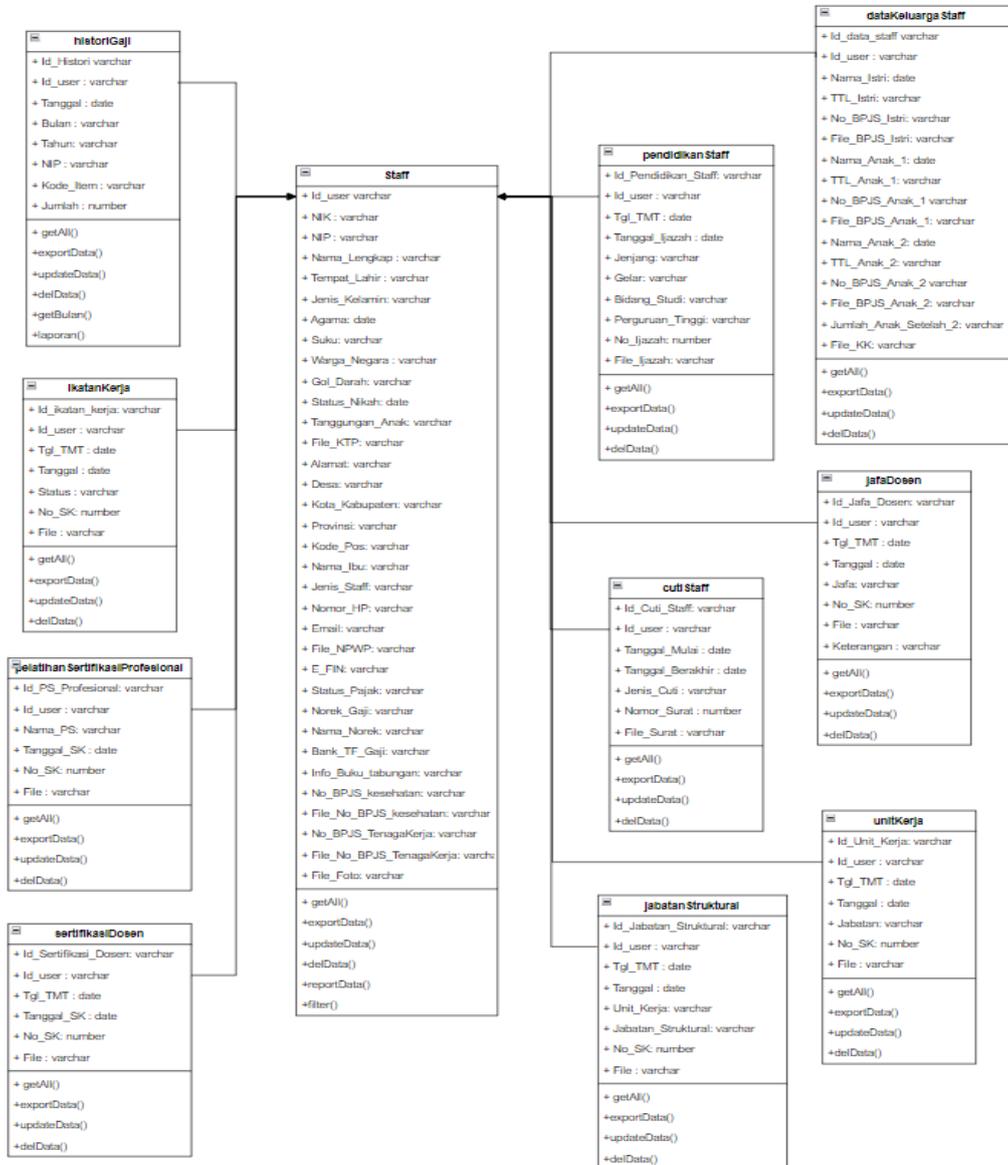
Hasil dari penelitian ini adalah mengimplementasikan teknologi *RESTful web service* menggunakan *framework Laravel* pada sistem informasi manajemen sumber daya manusia. Tahap pertama di mulai dari pembuatan rancangan *Use Case Diagram*, *Class Diagram* dan *Activity Diagram*. Maka yang harus dilakukan pada tahap ini yaitu pembuatan *use case diagram*, digunakan untuk menggambarkan hubungan pengguna dengan sistem dan fungsionalitas aplikasi mengenai hak akses *RESTful web service* setiap pengguna [10].



Gambar 2. Use Case Diagram

Use Case Diagram pada Gambar 3 menggambarkan lima *actor* dalam sistem, yaitu Super Admin, Admin HRD, Kepala Unit, Staff dan Bendahara beserta fungsi-fungsi apa saja yang bisa dilakukan pada sistem. Semua interaksi yang dilakukan oleh aktor harus melakukan login terlebih dahulu.

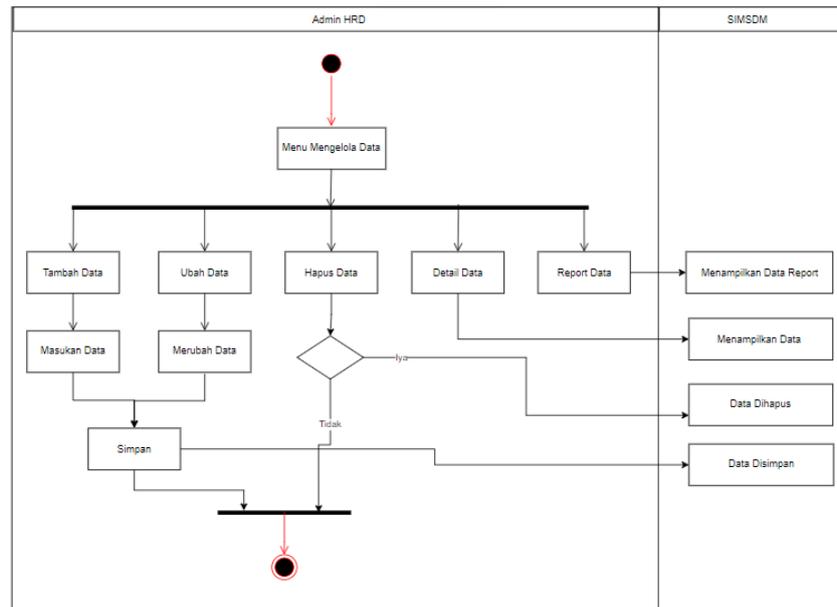
Selanjutnya, membuat *class diagram* untuk menggambarkan apa yang harus ada dalam *RESTful web service* sistem informasi manajemen sumber daya manusia yang dimodelkan dengan berbagai komponen. *Class diagram* menunjukkan sistem *class*, *attribute*, *method* dan hubungan antar objek [11].



Gambar 3. *Class diagram*

Class diagram diatas merupakan *class diagram* pada sistem informasi sumber daya manusia. Dalam *Class Diagram* terdapat sebelas *class* yang telah memiliki *attribute* dan *method* serta hubungannya masing-masing. Terakhir, tahap perancangan sistem pada penelitian ini yaitu membuat *Activity Diagram*.

Activity Diagram merupakan diagram yang memodelkan proses-proses yang terjadi pada sebuah sistem yang berisi aktivitas dan tindakan yang dapat dilakukan setiap user pada *service*. Pada Gambar 5 merupakan *Activity Diagram* yang digunakan untuk menggambarkan aktivitas Admin HRD pada sistem [12].



Gambar 4. Activity Diagram

Setelah tahap perancangan sistem dilanjutkan tahap pengkodean dan pengujian. Pada tahap ini, *Framework Laravel* digunakan untuk membangun *RESTful web service*. Arsitektur *RESTful web service* yang digunakan dalam penelitian ini menerapkan standar keamanan menggunakan *Library Laravel Passport*. Sehingga setiap komunikasi dan pertukaran informasi sistem informasi manajemen sumber daya manusia membutuhkan autentikasi. *Laravel Passport* digunakan untuk membuat *token keys* yang akan dikirimkan untuk pertukaran informasi oleh aktor.

Tahap pertama untuk membangun *RESTful web service* pada penelitian ini, terlebih dahulu menambahkan *Library Laravel passport*. *Laravel passport* akan menghasilkan file *migration* dan membuat tabel-tabel yang diperlukan. Tabel-tabel tersebut akan digunakan sebagai acuan token yang telah di *generate* saat aktor login. Selanjutnya melakukan konfigurasi *Laravel Passport* dengan cara merubah file *config/auth.php* menjadi seperti Tabel I.

TABEL I
KODE PROGRAM AUTENTIKASI PADA CONFIG/AUTH.PHP

```

1  'guards' => [
2      'web' => [
3          'driver' => 'session',
4          'provider' => 'users',
5      ],
6      'api' => [
7          'driver' => 'passport',
8          'provider' => 'users',
9          'hash' => false,
10     ],
11     'admin' => [
12         'driver' => 'passport',
13         'provider' => 'admin',
14         'hash' => false,
15     ],
16     'admin-hrd' => [
17         'driver' => 'passport',
18         'provider' => 'admin-hrd',
19         'hash' => false,
20     ],
21     'kepala-unit' => [
22         'driver' => 'passport',
23         'provider' => 'kepala-unit',
24         'hash' => false,
25     ],
26     'bendahara' => [
27         'driver' => 'passport',
28         'provider' => 'bendahara',
29         'hash' => false,
30     ]
31 ]

```

Kode program 1 bertujuan untuk menginstruksikan ke *project Laravel* yang menjadi *RESTful web service* untuk menggunakan *passport* ketika mengautentikasi *request* dari aktor. Untuk membuat hal tersebut perlu merubah *driver* pada *key api*, *admin*, *admin-hrd*, *kepala-unit*, dan *bendahara* ke *passport*. Hal ini dilakukan karena harus ada autentikasi berupa token pada saat mengakses *RESTful web service* pada penelitian ini. Oleh karena itu, konfigurasi di atas diperlukan dalam membangun sistem ini.

Framework Laravel menggunakan pola MVC (*Model-View-Controller*) dengan setiap bagian memiliki tugas masing-masing [13]. *Class*, *method* dan *attribute* yang telah dibuat di *class diagram* akan membantu dalam pembuatan *Controller* dan *Model* untuk pembuatan *RESTful web service* pada sistem. Sehingga tahap berikutnya yaitu membuat model yang digunakan untuk berkomunikasi dengan database.

TABEL II
KODE PROGRAM MODEL CUTI STAFF

```

1 class CutiStaffModel extends Model
2 {
3     protected $table = "cuti_staff";
4     protected $primaryKey = "id_cutiStaff";
5     public $timestamps = false;
6     public $incrementing = true;
7     public $fillable = [
8         'id_user', 'id_cutiStaff', 'tanggal_mulai', 'tanggal_berakhir',
9         'jenis_cuti', 'nomor_surat', 'file_surat'
10    ];
11 }

```

Tabel II merupakan Model untuk Cuti Staff, *class CutiStaffModel* adalah *Extends* dari *Class Model*. *Class Model* adalah bagian dari *library Eloquent*, yaitu *library* yang membantu dan mempermudah untuk proses pengolahan data. Pada baris tiga dan empat mendefinisikan nama tabel dan *primary key*. Adapun pada baris tujuh sampai baris sepuluh adalah *array fillable* yang digunakan untuk menampung *field* dari table Cuti Staff. Sementara itu, pengolahan data pada Cuti Staff dilakukan pada *controller* untuk membuat *service* yang berguna sebagai pengolah data *insert*, *read*, *update*, dan *delete* sesuai kebutuhan data seperti yang ditunjukkan pada Tabel III.

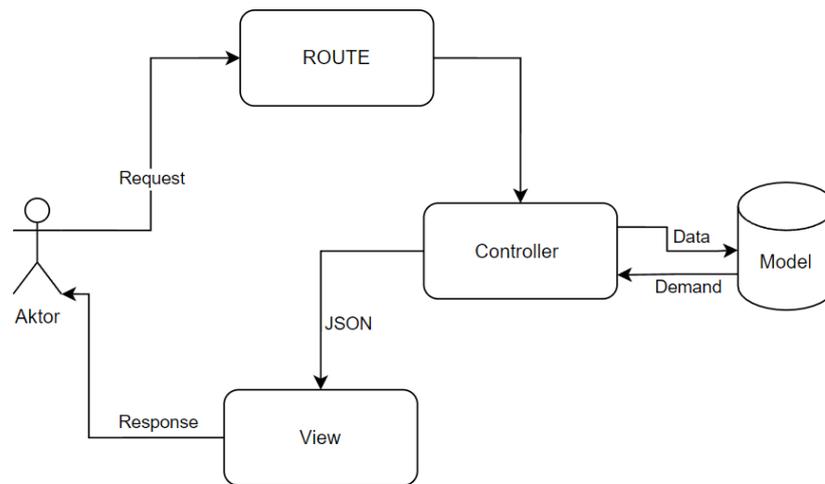
TABEL III
KODE PROGRAM CONTROLLER CUTI STAFF

```

1 use App\CutiStaffModel;
2 use DB;
3 class cutiStaffController extends Controller {
4     public function getAll(Request $request){
5         return response()->
6             json(CutiStaffModel::select-
7                 tRaw('cuti_staff.id_user,user.nip,user.nama_lengkap,cuti_staf
8                     f.*')->
9                     join('staff','staff.id_user','=', 'cuti_staff.id_user')->
10                    get(),200);
11    }
12    public function exportData(Request $request){...}
13    public function addData(Request $req){...}
14    public function updateData($id , Request $req){...}
15    public function delData($id){...}
16 }

```

Controller ini digunakan sebagai penghubung antara *request client* atau *view* ke *class model*. Pada baris satu, *CutiStaffModel* diterapkan pada *controller* ini untuk mengenali *field* yang berapa pada model. Adapun pada baris tiga ditunjukkan nama *controller* yaitu *cutiStaffController*. *Public function getAll* merupakan *method* yang digunakan untuk mengambil data berupa *String* dari hasil *join* antara tabel *cuti_staff* dan *staff*. Setelah mengeksekusi perintah, *controller* akan mengembalikan nilai dalam bentuk *JSON*. Ditandai dengan *HTTP response code 200*, menunjukkan permintaan aktor ke *service* berhasil dijalankan oleh server. Baris sebelas sampai lima belas merupakan *method* lain dari *class cutiStaffController*. Pada gambar dibawah ini dapat diperlihatkan gambaran dari pola MVC framework Laravel.

Gambar 5. Pola MVC pada *Framework Laravel*

Dari Gambar 5 dapat dijelaskan bahwa struktur pola MVC *framework Laravel* terdapat perbedaan dengan pola MVC pada umumnya. Perbedaan terletak pada *route* yang menghubungkan antara *request* dari user dan *controller*. Sehingga *request* dari user tidak diterima langsung oleh *controller* [14]. Setelah membuat model dan *controller* langkah selanjutnya untuk pengkodean *RESTful web service* agar dapat di gunakan adalah pembuatan *route*. Tabel IV adalah kode program yang menunjukkan *route* yang didefinisikan pada *framework Laravel*.

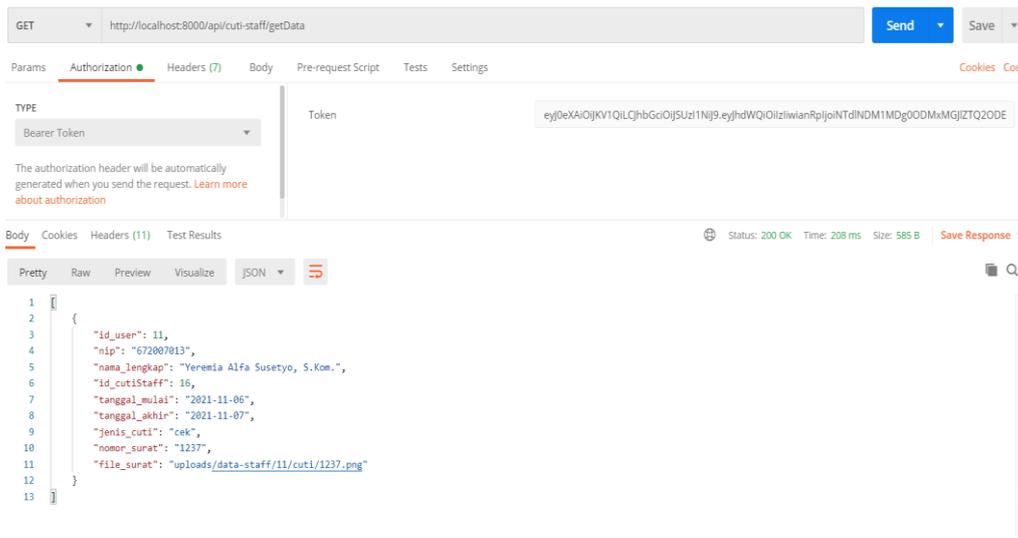
TABEL IV
KODE PROGRAM ROUTE

1	<code>Route::group(['middleware' => ['scope:admin,admin-hrd,kepala-</code>
2	<code>unit,user,bendahara']],function(){</code>
3	<code>Route::get('/cuti-staff/getData', 'cutiStaffController@getAll');</code>
4	<code>});</code>

Baris satu menunjukkan *middleware* agar sesuai dengan autentikasi hak akses dari *scopes* sesuai dengan role dari token yang dihasilkan dari *library laravel passport* yang lebih dulu didefinisikan. Pada *route* Tabel 4 menggunakan method GET dan *path* URI `/cuti-staff/getData` yang eksekusinya memanggil *cutiStaffController* dengan *method* `getAll` yang berada pada *class controller*. Terdapat beberapa *method* yang dapat digunakan pada *route* tergantung *request* yang dikirimkan oleh aktor.

Web service adalah sekumpulan layanan yang bisa diakses melalui jaringan protocol HTTP. Setiap layanan yang ada berbasis *website* dan dapat diterapkan pada *back-end* dan *front end* yang berbeda dalam suatu aplikasi. Penelitian ini menggunakan *web service* yang menggunakan model REST terdapat beberapa *method* yang disediakan seperti GET, POST, PUT dan DELETE [15]. *Method* digunakan untuk parameter pemanggilan *service*.

Tahap terakhir dari pembangunan *RESTful web service* ini adalah dengan menguji REST API yang telah dibuat menggunakan Postman. Pengujian ini dilakukan untuk memastikan response yang dihasilkan dari proses *request* ke REST API dapat diterapkan ke sistem informasi manajemen sumber daya manusia dengan berbagai jenis unit dan layanan. REST API memiliki sifat *client-server* sehingga dalam pengujian yang dilakukan menggunakan postman harus mengirimkan *request* ke sistem *RESTful web service*. *Request* dikirim dalam bentuk alamat *endpoint*, HTTP *method* (GET, POST, PUT, DELETE), *headers*, dan *body* sebagai parameter. RESTful akan menerima *request* dan memberikan *response* dalam format JSON. Gambar dibawah melihat pengujian REST API menggunakan Postman.



Gambar 6. Hasil Pengujian pada Postman

Dari hasil pengujian pemanggilan REST API data Cuti Staff didapatkan hasil bahwa *response* dalam bentuk JSON sesuai dengan hasil yang diharapkan dan dapat diterapkan pada sistem. Pada Gambar 6 diperlihatkan bahwa harus mengirimkan parameter *Authorization* berupa *Bearer Token* yang dihasilkan dari proses login. *Return value* yang dihasilkan dari REST API berisi data-data atribut atau *record* dari suatu tabel.

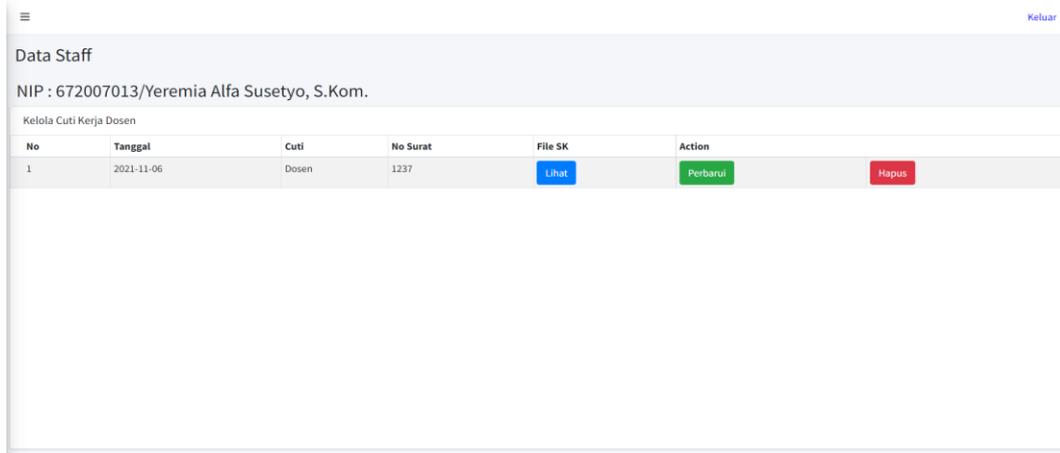
Salah satu keuntungan *RESTful web service* yang telah dibangun ini telah membagi layanan menjadi bagian lebih kecil dan terpisah. Sehingga pembagian layanan di unit-unit lain akan dengan mudah dengan adanya *RESTful web service*. Hal itu tidak mempengaruhi dari segi *back-end* dan *front end* sistem yang akan menggunakan *RESTful web service* penelitian ini.

Pada penelitian ini, *front end application* di bangun menggunakan *framework bootstrap* dan *back-end application* menggunakan *framework Laravel* dan *Javascript*. Dilakukan implementasi pada *user interface* (UI) pada gambar yang menunjukkan data Cuti Staff.

TABEL V
KODE PROGRAM PEMANGGILAN SERVICE

1	<code>url:" http://localhost:8000/api/cuti-staff/getData",</code>
2	<code>framework Laravel:"GET",</code>
3	<code>data:{</code>
4	<code> _token : "{{csrf_token()}}"</code>
5	<code>},</code>
6	<code>headers:{"Authorization": "Bearer " + getCookie</code>
7	<code> ("passport_token"),</code>
8	<code>success:function(data){</code>
9	<code> var output = "";</code>
10	<code> ... }</code>

Kode program diatas merupakan kode program yang menunjukkan cara mengakses *RESTful web service* yang menerapkan *laravel passport* pada *Javascript*. Struktur *request* dalam bentuk alamat *endpoint* dari *RESTful web service* pada baris satu. *Method* GET yang ditunjukkan pada baris kedua digunakan untuk mendapatkan data sesuai dengan path URI yang dikirimkan pada baris satu yaitu `/cuti-staff/getData`. Selanjutnya, header berupa token pada baris empat. Terakhir, pada baris delapan ini akan menginterpretasikan data dalam bentuk *user interface*. Salah satu contoh tampilan tabel pada *user interface* ditunjukkan pada gambar 7.



No	Tanggal	Cuti	No Surat	File SK	Action
1	2021-11-06	Dosen	1237	Lihat	Perbarui Hapus

Gambar 7. Hasil Implementasi Service

Setelah melakukan implementasi ke *user interface*, dilakukan pengujian terhadap sistem informasi yang pada penelitian ini. Metode Pengujian yang diterapkan pada pengujian yaitu metode *black box*. Pengujian dilakukan dengan mengetes setiap pemanggilan service yang sudah diterapkan pada tampilan *user*, pada Tabel VI merupakan hasil dari pengujian tersebut.

TABEL VI
HASIL PENGUJIAN BLACK BOX

Controller	Pengujian Route	Method	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Login		POST	Mendapatkan Autentikasi Login dengan memasukkan username dan password	Pengguna yang terdaftar mendapatkan response sukses dan mendapatkan response gagal ketika belum terdaftar	Sesuai
CutiStaff	addData	POST	Dapat menambahkan data ke tabel Cuti Staff	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel Cuti Staff	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari tabel Cuti Staff	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai
	getData	GET	Dapat mengambil data dari tabel Cuti Staff	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai
	exportData	GET	Dapat mengambil data dari tabel Cuti Staff sesuai parameter	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
PendidikanStaff	addData	POST	Dapat menambahkan data ke tabel Pendidikan Staff	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel Pendidikan Staff	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari tabel Pendidikan Staff	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai
	getData	GET	Dapat mengambil data dari tabel Pendidikan Staff	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai
	exportData	GET	Dapat mengambil data dari tabel Pendidikan Staff sesuai parameter	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
JabatanStruktural	addData	POST	Dapat menambahkan data ke tabel Jabatan Struktural	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel Jabatan Struktural	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari tabel Jabatan Struktural	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai
	getData	GET	Dapat mengambil data dari tabel Jabatan Struktural	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai
	exportData	GET	Dapat mengambil data dari tabel Jabatan Struktural	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
UnitKerja	addData	POST	Dapat menambahkan data ke tabel UnitKerja	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel UnitKerja	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari tabel UnitKerja	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai

	getData	GET	Dapat mengambil data dari tabel Unit-Kerja	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai
	exportData	GET	Dapat mengambil data dari tabel Unit-Kerja	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
DataKeluargaStaff	addData	POST	Dapat menambahkan data ke tabel Data Keluarga	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel Data Keluarga	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari tabel Data Keluarga	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai
	getData	GET	Dapat mengambil data dari tabel Data Keluarga	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai
	exportData	GET	Dapat mengambil data dari tabel Data Keluarga	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
IkatanKerja	addData	POST	Dapat menambahkan data ke tabel Ikatan Kerja	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel Ikatan Kerja	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari Ikatan Kerja	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai
	getData	GET	Dapat mengambil data dari tabel Ikatan Kerja	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai
	exportData	GET	Dapat mengambil data dari tabel Ikatan Kerja	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
Sertifikasi	addData	POST	Dapat menambahkan data ke tabel Sertifikasi	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel Sertifikasi	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari Tabel Sertifikasi	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai
	getData	GET	Dapat mengambil data dari tabel Sertifikasi	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai
	exportData	GET	Dapat mengambil data dari tabel Sertifikasi	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
Pelatihan-Profesional	addData	POST	Dapat menambahkan data ke tabel Pelatihan Sertifikasi Profesional	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel Pelatihan Sertifikasi Profesional	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari Tabel Pelatihan Sertifikasi Profesional	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai
	getData	GET	Dapat mengambil data dari tabel Pelatihan Sertifikasi Profesional	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai
	exportData	GET	Dapat mengambil data dari tabel Pelatihan Sertifikasi Profesional	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
HistoriGaji	addData	POST	Dapat menambahkan data ke tabel Histori Gaji	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel Histori Gaji	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari Tabel Histori Gaji	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai
	getData	GET	Dapat mengambil data dari tabel Histori Gaji	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai
	exportData	GET	Dapat mengambil data dari tabel Histori Gaji	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
Jafa	addData	POST	Dapat menambahkan data ke tabel Jafa	Data berhasil di input dan ditampilkan pada halaman user	Sesuai
	updateData	POST	Dapat merubah data di tabel Jafa	Data berhasil di ubah dan ditampilkan pada halaman user	Sesuai
	delData	DELETE	Dapat menghapus data dari Tabel Jafa	Data berhasil di hapus dari tabel dan ditampilkan pada halaman user	Sesuai
	getData	GET	Dapat mengambil data dari tabel Jafa	Berhasil mendapatkan semua data dari tabel dan ditampilkan pada halaman user	Sesuai

exportData	GET	Dapat mengambil data dari tabel Jafa	Berhasil mendapatkan response data sesuai parameter dan disimpan dalam bentuk excel	Sesuai
------------	-----	--------------------------------------	---	--------

Dari hasil pengujian yang telah dilakukan, didapatkan hasil bahwa setiap pengujian yang disimulasikan sudah sesuai dengan hasil yang diharapkan. Sistem informasi ini mengelola data yang diperoleh dari *RESTful web service* dan memberikan layanan ke unit dan departemen lain tentang masalah sumber daya manusia. Sistem informasi ini menerapkan otorisasi terhadap hak akses setiap *client* atau *actor*. Sehingga tidak semua *client* dapat mengakses semua REST API yang sudah dibuat. Hak akses tertinggi terdapat pada Super Admin dan dilengkapi dengan admin *dashboard* yang digunakan untuk mengontrol sistem informasi, meliputi fungsi input data, mengubah data dan menghapus data. Semua fungsi yang tersedia dalam sistem informasi ini berbasis *RESTful web service*

IV. KESIMPULAN

Setelah melakukan penelitian ini dapat disimpulkan bahwa, dengan pembuatan sistem informasi manajemen sumber daya manusia yang terintegrasi dapat membantu dalam pengelolaan data yang lebih efektif dan efisien. Integrasi data dibuat dengan menerapkan *RESTful web service* didukung oleh keamanan pertukaran sistem informasi menggunakan *library Laravel passport* dengan bahasa pemrograman PHP menggunakan *framework Laravel*. Akses *web service* dilakukan dengan mengirimkan *request* beserta token. Data *response* dari *RESTful web service* memiliki format JSON. Dilakukan juga pengujian terhadap *user interface* dan hasilnya sesuai dengan yang diharapkan.

Dengan adanya sistem informasi manajemen sumber daya manusia berbasis *website*, organisasi dapat melakukan beberapa proses pengelolaan data secara sistematis dan terkontrol pada satu pusat yaitu *RESTful web service*. Saran untuk penelitian ini dapat dikembangkan menggunakan *platform mobile* sehingga lebih fleksibel ketika digunakan.

DAFTAR PUSTAKA

- [1] M. Ma'ruf, H. Junaedi, S. Susanti, and A. Mubarak, "PENERAPAN FRAMEWORK LARAVEL PADA APLIKASI HRIS (HUMAN RESOURCE INFORMATION SYSTEM)," *JURNAL RESPONSIF*, vol. 2, no. 2, pp. 176–183, 2020, [Online]. Available: <http://ejurnal.ars.ac.id/index.php/jti>
- [2] N. K. Wardhani, ; Muhammad Thariq, and A. Aziz, "SISTEM INFORMASI MANAJEMEN SUMBER DAYA MANUSIA BERBASIS WEB (STUDI KASUS: PT. KLIK TEKNOLOGI INDONESIA)," *Jurnal TECHNO Nusa Mandiri*, vol. 15, no. 2, p. 145, 2018, [Online]. Available: www.mercubuana.ac.id/www.mercubuana.ac.id
- [3] R. Ary Putra, R. Wahu Saputra, G. Hilman Pradana, F. Sains dan Teknologi, and U. Sunan Ampel -Surabaya, "Nusantara Journal of Community Engagement IMPLEMENTASI SISTEM INFORMASI MANAJEMEN SUMBER DAYA PERUSAHAAN DI PT. DAVINTI INDONESIA," vol. 1, no. 1, 2020, [Online]. Available: <http://ejournal.kopertais4.or.id/tapalkuda/index.php/NJCE/index>
- [4] Yusrizal, R. Dawood, and Roslidar, "Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter," *Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter*, vol. 2, no. 2252–7036, pp. 1–8, 2017.
- [5] T. Andriyanto and R. Indriati, "RESTful Web service Untuk Integrasi Antar Sistem Informasi," 2017. [Online]. Available: <http://service.teguh.dosen.unpkediri.ac.id/servi>
- [6] M. Agus Arianto, S. Munir, and K. Khotimah, "ANALISIS DAN PERANCANGAN REPRESENTATIONAL STATE TRANSFER (REST) WEB SERVICE SISTEM INFORMASI AKADEMIK STT TERPADU NURUL FIKRI MENGGUNAKAN YII FRAMEWORK," *Jurnal Teknologi Terpadu*, vol. 2, no. 2, 2016.
- [7] S. P. Utomo, N. H. Alfiah, Z. A. Sani, M. Hanafi, and A. Primadewi, "Perancangan RESTful Web Service Pada Sistem Informasi Terintegrasi Menggunakan Framework CodeIgniter," 2020.
- [8] W. Galindra Wardhana, I. Arwani, and B. Rahayudi, "Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)," 2020. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [9] P. F. Tanaem, "RESTful Web Service Untuk Sistem Pencatatan Studi Kasus PT. XYZ," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 2, no. Jurnal Teknik Informatika dan Sistem Informasi, 2016.
- [10] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 18, no. 2, pp. 284–293, May 2019, doi: 10.30812/matrik.v18i2.407.
- [11] W. Chrisna, U. Dagha, and Y. A. Susetyo, "Pembangunan Aplikasi Web Event Menggunakan Framework Spring Boot di PT XYZ," 2021. [Online]. Available: <http://jurnal.mdp.ac.id>
- [12] F. Abdussalaam and S. A. Saputra, "PERANCANGAN SISTEM INFORMASI COMPLAINT MANAGEMENT DENGAN METODE RAD MENGGUNAKAN FRAMEWORK LARAVEL," 2018.
- [13] R. Renaldo Prasena and H. Sama, "STUDI KOMPARASI PENGEMBANGAN WEBSITE DENGAN FRAMEWORK CODEIGNITER DAN LARAVEL," [Online]. Available: <http://journal.uib.ac.id/index.php/cbsit>
- [14] D. Purnama Sari, R. Wijanarko, and J. X. Menoreh Tengah, "Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang)," vol. 2, no. 1, pp. 32–36, 2019.
- [15] R. Rizal and A. Rahmatulloh, "RESTful Web Service untuk Integrasi Sistem Akademik dan Perpustakaan Universitas Perjuangan," 2019.