AINET Jurnal Informatika
e-ISSN : 2686 – 1917
p-ISSN : 2657 – 0653

Implementasi Metode *Open Authorization* (OAuth2) Untuk Pengelolaan Data Dosen di Universitas Islam Nusantara

Yenni Fatman*[1], Renova Octaviawati[2]
[1,2]Universitas Islam Nusantara
e-mail: yennifatman@gmail.com*

Abstract

Islamic Nusantara University (UNINUS) currently does not have a good system for lecturer data management in order to meet the needs of all parties within the Islamic Nusantara University (UNINUS). Work units and faculties within UNINUS cannot yet access lecturer data easily and safely because the lecturer data management system is closed. This is different from other systems in the UNINUS environment that can be accessed openly using the client server facility with login and password notifications. There is a desire for a short time at this time limited access to the lecturer management database, however problems arising related to security in authorization to access the desired source. The research was made to create an authentication module requesting access to the lecturer database, by utilizing a third-party (client) authentication process using the Open Authorization (Oauth 2.0) authorization protocol. By utilizing the Simple Mail Transfer Protocol (SMTP) protocol, tokens are sent to the destination e-mail server, and then verified using the OAuth 2.0 third-party authorization module. If the recipient of the email is valid, then the token will be sent immediately and the client can access the lecturer database. The research produced a system that can provide services with guaranteed ease and security in the form of Application Programming Interface (API) -based modules by implementing OAuth 2.0 and SMTP authorization which is expected to be an alternative solution to overcome the problem of access to resources and lecturer data transactions in UNINUS.

Keyword*: Lecturer Data Processing System; Open Authorization (OAuth2); Simple Mail Transfer Protocol (SMTP); Application Programming Interface (API)*

Abstrak

Universitas Islam Nusantara saat ini belum memiliki sistem yang baik untuk pengelolaan data dosen agar dapat memenuhi kebutuhan semua pihak di lingkungan Universitas Islam Nusantara (UNINUS). Unit kerja dan fakultas di lingkungan UNINUS belum dapat mengakses langsung data dosen dengan mudah dan aman karena sistem pengelolaan data dosen bersifat tertutup. Hal ini berbeda dengan sistem lainnya di lingkungan UNINUS yang sudah dapat diakses terbuka menggunakan fasilitas *client server* dengan notifikasi *login* dan *password*. Adanya keinginan dalam waktu singkat saat ini akses terbatas ke database pengelolaan dosen, namun demikian timbul persoalan terkait keamanan dalam otorisasi untuk mengakses sumber yang diinginkan. Penelitian dibuat untuk membuat sebuah modul otentifikasi permintaan akses ke database dosen, dengan memanfaatkan proses otentikasi pihak ketiga (*client*) menggunakan protokol otorisasi *Open Authorization* (*OAuth 2.0*). Dengan memanfaatkan protokol *Simple Mail Transfer Protocol* (SMTP), token dikirimkan ke server email tujuan, untuk kemudian dilakukan verifikasi menggunakan modul otorisasi pihak ketiga OAuth 2.0. Jika penerima email valid, maka token akan segera dikirim dan *client* dapat mengakses database dosen. Penelitian menghasilkan sistem yang dapat memberikan layanan dengan kemudahan dan keamanan yang terjamin dalam bentuk modul berbasis *Application Programming Interface* (API) dengan mengimplementasikan otorisasi OAuth 2.0 dan SMTP yang diharapkan mampu menjadi alternatif solusi untuk mengatasi permasalahan akses sumber daya dan transaksi data dosen di lingkungan UNINUS.

Kata kunci: Sistem Pengolahan Data Dosen; *Open Authorization* (OAuth2); *Simple Mail Transfer Protocol* (SMTP); *Application Programming Interface* (API)

1. Pendahuluan

Pengelolaan data dosen di sebuah perguruan tinggi merupakan sesuatu yang sangat penting, di Universitas Islam Nusantara saat ini belum memiliki sistem informasi tersendiri untuk pengelolaan data dosen tersebut. Untuk itu diperlukan sistem informasi penggelolaan data dosen yang lebih baik dan dapat memenuhi kebutuhan semua pihak di lingkup Universitas Islam Nusantara. Permasalahan yang ada adalah bagaimana mengelola hak akses untuk pihak lain di lingkungan Universitas Islam Nusantara seperti fakultas – fakultas dapat mengakses data dosen tersebut dengan mudah dan aman.

Sistem keamanan yang paling sering digunakan pada metode otentikasi adalah kata sandi (*password*). Adanya kemudahan dalam hal implementasi menjadi faktor utama dari pemanfaatan sistem berbasis *password* dan pengguna juga sudah terbiasa dengan sistem semacam ini sehingga waktu penyesuaian dapat diminimalkan. Di sisi lain, banyaknya penggunaan jaringan yang belum aman (misalnya protokol HTTP) masih menjadi ancaman bagi pengguna mengingat seringkali *password* satu-satunya mekanisme yang digunakan [1].

OAuth (*Open Authorization*) merupakan suatu protokol terbuka yang mengizinkan otorisasi secara aman dengan metode yang sederhana dan standar dari aplikasi web, mobile dan desktop [2]. Protokol otorisasi OAuth memungkinkan pihak ketiga (*third-party*) untuk dapat mengakses sumber daya dengan akses tertentu melalui protokol HTTP. OAuth dirancang khusus untuk mengatasi permasalahan klasik pada model autentikasi pada *client-server* [3].

Proses otentikasi pihak ketiga (*client*) membutuhkan bantuan otentikasi melalui email. Proses otentikasi yang diterapkan yaitu dengan bantuan protokol SMTP (*Simple Mail Transfer Protocol*). SMTP merupakan salah satu protokol email yang umum digunakan untuk pengiriman email di Internet. Protokol ini dipergunakan untuk mengirimkan data dari komputer pengirim ke server email tujuan. SMTP adalah protokol yang cukup sederhana, berbasis teks dimana protokol ini menyebutkan satu atau lebih penerima email untuk kemudian diverifikasi. Jika penerima email valid, maka email akan segera dikirim [4].

Berdasarkan permasalahan dan fakta yang telah dijelaskan sebelumnya, maka dibutuhkan sebuah sistem yang dapat memberikan layanan dengan kemudahan dan keamanan yang terjamin yaitu sistem web API data dosen dengan mengimplementasikan otorisasi OAuth 2.0 dan SMTP yang diharapkan mampu menjadi alternatif solusi untuk mengatasi permasalahan akses sumber daya dan transaksi data dosen.

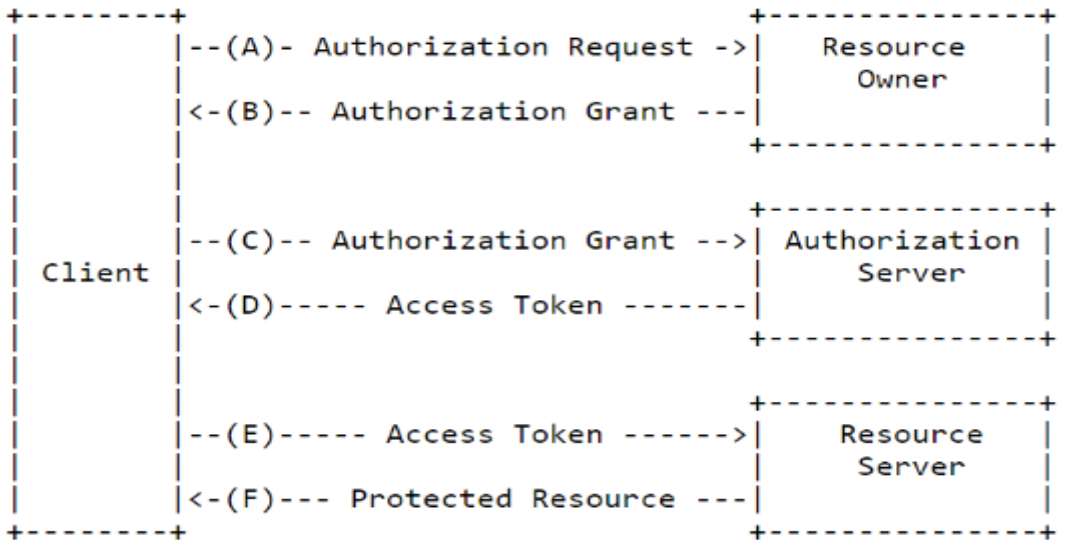
2. Metode Penelitian

2.1 Metode Pengembangan Sistem

Metode yang diterapkan pada penelitian ini adalah dengan pengembangan metode *waterfall*. Metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan [5].

2.2 Metode *Open Authorization* (OAuth 2.0)

OAuth merupakan suatu protokol terbuka yang mengizinkan otorisasi secara aman dengan metode yang sederhana dan standar dari aplikasi web, *mobile* dan *desktop*. Protokol otorisasi OAuth memungkinkan pihak ketiga *(third-party*) untuk dapat mengakses sumber daya dengan akses tertentu melalui protokol HTTP. OAuth dirancang khusus untuk mengatasi permasalahan klasik pada model autentikasi pada *client-server*[3]. Mekanisme kerja OAuth2 terbentuk dari peran aktif 4 (empat) bagian yang terdiri dari : *client, resource owner, authorization server, resource server* [6].



Gambar 1. Proses Kinerja OAuth2 [6]

Penjelasan untuk tugas dan fungsi dari komponen – komponen OAuth2 pada gambar 1 ditunjukkan pada tabel 1.

Tabel 1. Fungsi 4 Komponen OAuth2 [6]

Kode	Keterangan
A	*Client* melakukan permintaan otorisasi dari *resource owner*. Permintaan otorisasi dapat dilakukan langsung menuju *resource owner*, atau jika tidak langsung melalui perantara *authorization server*.
B	*Client* mendapatkan persetujuan otorisasi yang merupakan *credential* mewakili otorisasi kepemilikan *client*. Pemberian otorisasi ini tergantung pada metode yang digunakan oleh *client* dan jenis yang didukung oleh *authorization server*.
C	*Client* melakukan permintaan akses token dengan otentikasi kepada *authorization server*, *client* mendapatkan penyajian hibah dan bentuk otorisasi dari *authorization server* seperti pada gambar proses kinerja OAuth 2.
D	Otorisasi server (*authorization server*) melakukan otentikasi kepada *client* dan memvalidasi pemberian otorisasi kepada *client*, jika sesuai dan berlaku, otorisasi server membagikan akses token.
E	*Client* melakukan permintaan sumber daya yang sudah diproteksi dari *resource server*, melakukan tindakan otentikasi dengan menghadirkan akses token.
F	*Resource server* memvalidasi akses token, jika valid dan sesuai, akan melayani permintaan *client* untuk menggunakan aplikasi yang sudah terlindungi.

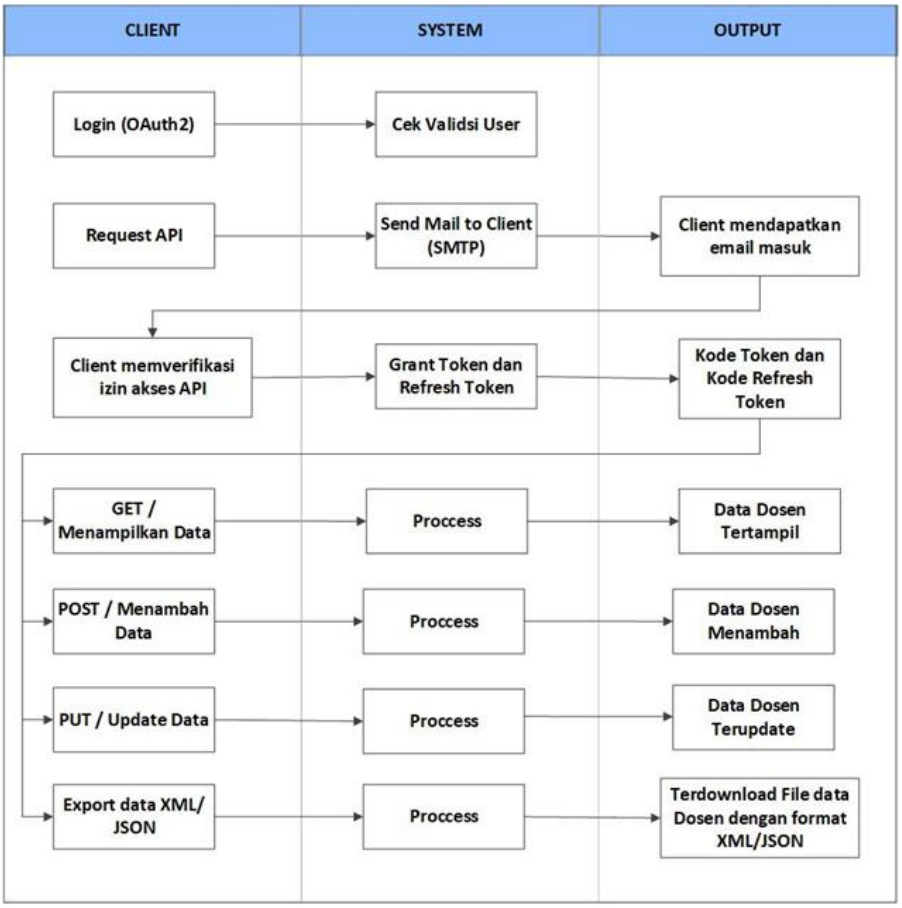
3. Hasil dan diskusi

Sistem web api data dosen ini akan menerapkan metode *Open Authorization Version2* (OAuth2) dan protokol *Simple Mail Transfer Protocol* (SMTP). Implementasi ini menggunakan sebuah *framework* PHP yang sudah teruji kemampuannya, yaitu *Framework* Laravel dengan dukungan *plugin* laravel *passport.*

3.1 Rancangan Proses Otorisasi dan Otentikasi

Rancangan proses otentikasi dan otorisasi menggunakan metode OAuth2 dengan protokol *Simple Mail Transfer Protocol* (SMTP) secara umum digambarkan sebagai berikut :

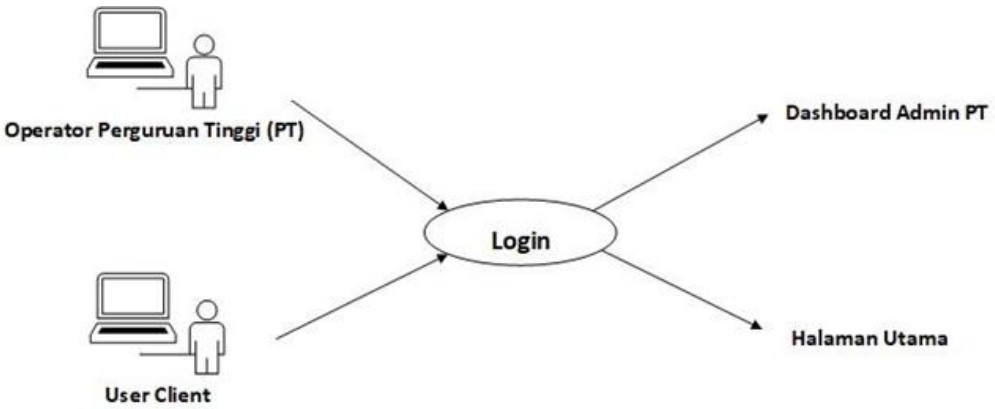
AINET Jurnal Informatika
e-ISSN : 2686 – 1917
p-ISSN : 2657 – 0653



Gambar 2. Proses Otorisasi dan Otentikasi

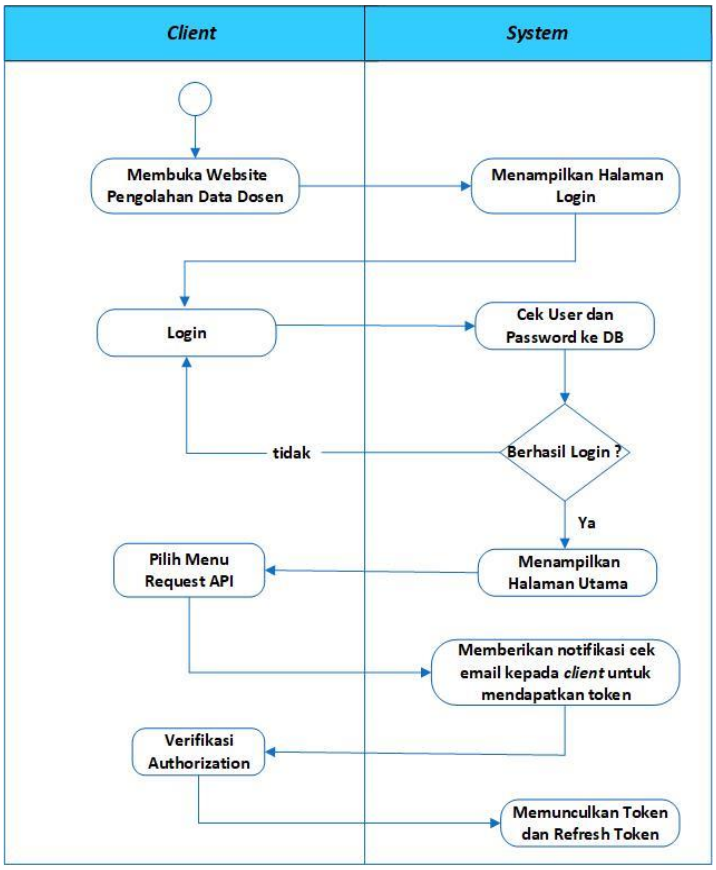
3.2 Perancangan Sistem

Aktor yang terlibat pada sistem pengolahan data dosen ini terdiri dari dua aktor utama yaitu operator perguruan tinggi (PT) dan *Client*. Aktor operator perguruan tinggi merupakan aktor yang dapat mengelola data dosen secara keseluruhan sedangkan aktor *client*, *client* dapat melakukan *login system*, *request* API, *refresh* token dan melakukan transaksi data dosen GET, POST dan PUT. *Use case diagram* ditunjukkan pada gambar 3 berikut ini.



Gambar 3. Use Case Diagram

Proses inti dari sistem ini ada pada pihak ketiga (*client*), dimana *client* dapat mendapatkan sumber daya data dosen seizin pemilik dan dapat melakukan transaksi data. Berikut merupakan *activity* diagram dari proses *request* API.



Gambar 4. Activity Diagram Request API

Gambar diatas menjelaskan proses *client request* API sampai dengan mendapatkan kode token dan kode refresh token untuk bisa akses API data dosen. Melalui token tersebut, *client* dapat melakukan transaksi baik itu GET (menampilkan), POST (menambah) dan PUT (*update*).

3.3 Implementasi Sistem

Setelah melakukan analisa serta perancangan sistem, langkah selanjutnya adalah impelementasi pada sistem yang akan dibuat. Terdapat 3 komponen utama di dalam OAuth2 yaitu :

a. Authentication Component

Komponen yang menyediakan *login page* yang ditampilkan ke *user*. Pada sistem ini dengan membuat *function authenticate* yaitu fungsi yang memberikan *return* dengan HTTP *status code*.

HomeController.php

```
50    public function authenticate()
51    {
52        $status = Auth::check();
53        if (!$status) {
54            return response()->json(['unauthorized'], 401);
55        }
56        return response()->json('authorized', 200);
57    }
58
```

Gambar 5. Function Authentication

b. Consent Component

Komponen ini akan muncul setelah proses *authentication component* selesai. Pada sistem ini terdapat 3 transaksi yang bisa dilakukan oleh *user* yaitu GET (menampilkan), POST (menambah) dan PUT (*update*). Terlebih dahulu harus mempersiapkan *routes* API data dosen Universitas Islam Nusantara.

api.php

```php
<?php

use Illuminate\Http\Request;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

// Route::group(['prefix' => 'data-dosen'], function(){
//  Route::get('/', 'api\\DataDosenController@index');
// });

Route::get('data-dosen', 'api\\DataDosenController@index');
Route::post('data-dosen', 'api\\DataDosenController@add');
Route::put('data-dosen/{id}', 'api\\DataDosenController@edit');

```

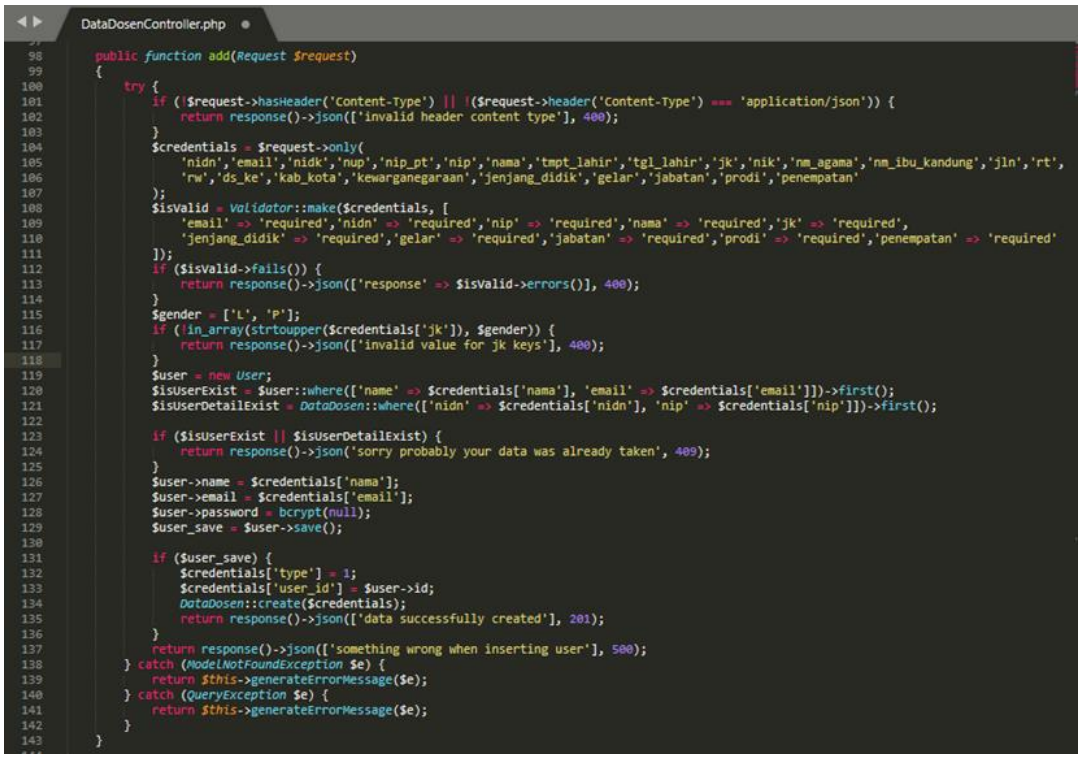
Gambar 6. Route API Data Dosen Uninus

DataDosenController.php

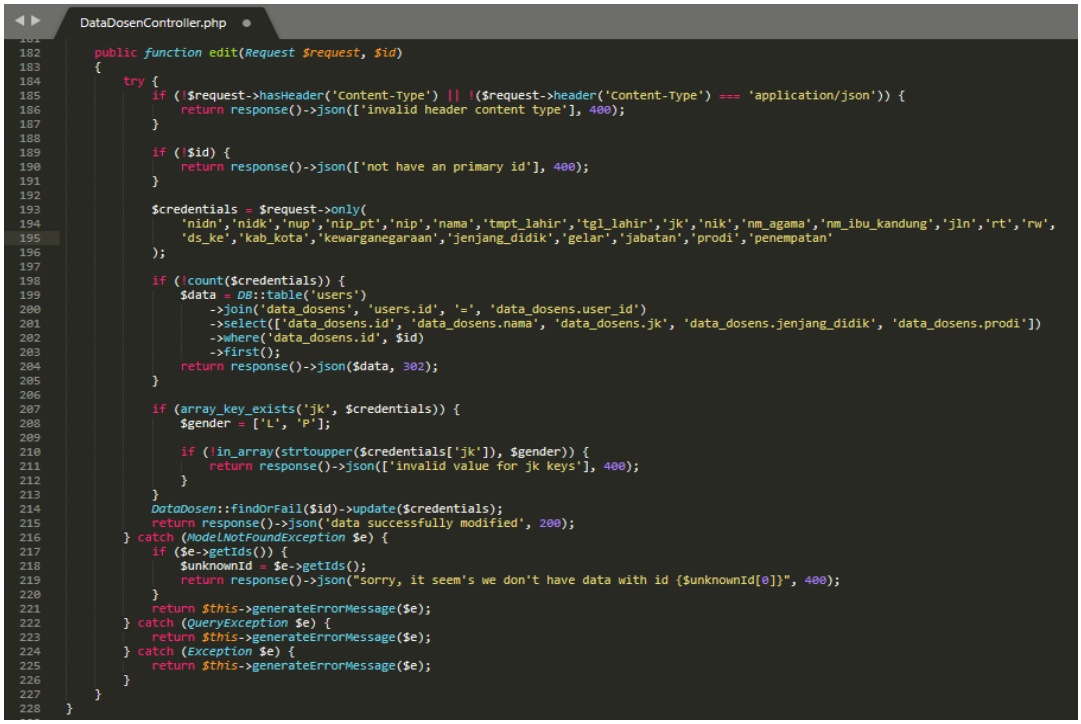
```php
        $allData = DB::table('users')
            ->join('data_dosens', 'users.id', '=', 'data_dosens.user_id')
            ->select(['data_dosens.id', 'data_dosens.nama', 'data_dosens.jk', 'data_dosens.jenjang_didik', '
                data_dosens.prodi'])
            ->where($query)
            ->get();
```

Gambar 7. GET API Data Dosen Uninus

AINET Jurnal Informatika
e-ISSN : 2686 – 1917
p-ISSN : 2657 – 0653



Gambar 8. POST API Data Dosen Uninus



Gambar 9. UPDATE API Data Dosen Uninus

c. *Token management*

Komponen ini bertugas untuk menjaga token – token yang sudah diberikan oleh Oauth2 yang tersimpan pada database seperti *request* token, *refresh* token dan *views* token.

```
home.blade.php
592
593         $('#reqToken').click(function() {
594             @if(Auth::check())
595                 var oauth = getOauthClient()
596                 oauth.success(function(response, jqXHR, textStatus) {
597                     if (!response.length) {
598                         createOauthClient()
599                     }
600                     sendMail().success(function(response, jqXHR, textStatus) {
601                         if (textStatus.status === 200) {
602                             $('#request-modal').find('.modal-title').html('Request Token processing')
603                             $('#request-modal').find('.modal-body h4').html('Please check your email to grant your
                                access')
604                             $('#request-modal').modal('show')
605                         }
606                     });
607                 });
608             @else
609                 $('#request-modal').find('input').css('color', 'black')
610                 $('#request-modal').modal('show')
611             @endif
612         });
```

Gambar 10. Request Token

```
home.blade.php
587             $('#refSubmit').click(function() {
588                 if ($('#refresh-modal').find('input').val()) {
589                     refreshToken($('#refresh-modal').find('input').val())
590                 }
591             });
592
```

Gambar 11. RefreshToken

```
token.blade.php
17          <br/>
18          <br/>
19          <!-- request -->
20  @endsection
21
22  @section('js')
23      <script type="text/javascript">
24          $(document).ready(function() {
25              $.ajax({
26                  method: "post",
27                  url: '{{ url("oauth/token") }}',
28                  headers: {
29                      'X-CSRF-TOKEN': '{{ csrf_token() }}'
30                  },
31                  data: {
32                      grant_type: 'authorization_code',
33                      client_id: '{{ $data["client_id"] }}',
34                      client_secret: '{{ $data["client_secret"] }}',
35                      redirect_uri: '{{ $data["redirect_uri"] }}',
36                      code: '{{ $data["code"] }}',
37                  },
38                  success: function(response, jqXHR, textStatus) {
39                      $('#success-header').find('h3').html('token successfully acquired')
40                      var template = '<div>Token Type: '+ response.token_type +'</div>'+
41                      '<div>Access Token: ' + response.access_token + '</div>' +
42                      '<div>expired : ' + response.expires_in + '</div>' +
43                      '<div>Refresh Token: '+ response.refresh_token + '</div>' +
44                      '<div><h3>Please save this credentials because it will be not showed again</h3></div>'
45                      $('#success-body').html(template)
46                  },
47                  error: function(jqXHR) {
48                      console.log(jqXHR)
49                  },
50              });
51          });
52      </script>
53  @endsection
```

Gambar 12. ViewToken

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan. Pertama, sistem web API data dosen menggunakan metode OAuth2 dan SMTP berhasil dibuat dengan menggunakan *Framework* Laravel dan basis data MySQL, menggunakan metode OAuth2 yang diterapkan pada fitur *login* serta menggunakan tambahan protokol SMTP yang membantu proses *authentication*. Kedua, sistem web API data dosen ini diharapkan dapat berguna bagi para pengembang sistem informasi untuk mendapatkan sumber daya maupun transaksi data dosen khususnya para pengembang sistem informasi Universitas Islam Nusantara. Ketiga, keamanan data sangat terjaga karena dengan mengimplementasi metode OAuth2 semua data tidak bisa di akses jika pengguna tidak terautentikasi.

Referensi

[1] R. Munadi, Z. Musliyana, T. Y. Arif, A. Afdhal, and S. Syahrial, “Kombinasi Waktu Sinkronisasi dan Nilai Salt untuk Peningkatan Keamanan pada Single Sign-On,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 5, no. 3, pp. 201–206, 2016.

[2] A. Parecki, “OAuth Community Site,” *OAuth*, 2017. [Online]. Available: https://oauth.net/. [Accessed: 18-May-2020].

[3] M. Jones and D. Hardt, “The oauth 2.0 authorization framework: Bearer token usage,” RFC 6750, October, 2012.

[4] Y. Y. Y. Syahrir, X. B. N. Najoan, and A. A. E. Sinsuw, “Rancang Bangun Aplikasi Cross Protocol Email dan SMS,” *J. Tek. Inform.*, vol. 13, no. 1, 2018.

[5] P. Roger S, *Rekayasa Perangkat Lunak*, Edisi 7. Yogyakarta: Andi, 2012.

[6] D. Rachmawati and A. Muchtar, “Perancangan Dan Implementasi Resource Server Dan Authorization Server Menggunakan Teknologi Otentikasi Oauth 2.”