Perancangan REST Web Service Untuk Pengembangan Sistem Pengajuan Simpan Pinjam PT. XYZ

Narwastu Jati Grasias[1)*)], Suprihadi [2)]

[1)2)]Teknik Informatika, Teknologi Informasi, Universitas Kristen Satya Wacana
Correspondence author : 672017006@student.uksw.edu, Salatiga, Indonesia
DOI: https://doi.org/10.37012/jtik.v8i2.1182

Abstrak

Sistem pengajuan simpan pinjam PT.XYZ merupakan sistem yang digunakan calon debitur PT.XYZ untuk melakukan pengajuan simpan pinjam. Dalam perkembangannya terdapat kendala untuk melakukan integrasi data dengan sistem lain dikarenakan perbedaan bahasa pemrograman. Oleh sebab itu dilakukan pengembangan pada sistem pengajuan simpan pinjam agar data yang tersimpan pada sistem pengajuan simpan pinjam dapat diakses dan diintegrasi dengan sistem lain PT.XYZ. Penelitian ini bertujuan merancang *backend* sistem pengajuan simpan pinjam dengan teknologi *REST web service*. Perancangan dibangun menggunakan *micro-framework* Lumen dan JWT yang berfungsi sebagai keamanan *web service*. Pengujian terhadap sistem dilakukan menggunakan metode *Blackbox* dan dihasilkan *web service* yang dapat membuat dan menyediakan informasi untuk sistem lain dalam bentuk data JSON.

Kata kunci : REST, *Web Service*, Lumen, JWT, JSON

Abstract

PT XYZ's savings and loan application system is a system used by prospective debtors of PT XYZ to submit savings and loans. In its development, there are obstacles to data integration with other systems due to differences in programming languages. Therefore, the development of the savings and loan application system is carried out so that the data stored in the savings and loan application system can be accessed and integrated with other PT. XYZ systems. This study aims to design a backend savings and loan application system with REST web service technology. The design is built using the Lumen micro-framework and JWT which functions as web service security. Testing of the system is carried out using the Blackbox method and a web service is generated that can create and provide information for other systems in the form of JSON data.

Keywords: *REST, Web Service, Lumen, JWT, JSON*

PENDAHULUAN

Lembaga yang bergerak dibidang sektor jasa keuangan tidak hanya dibutuhkan oleh masyarakat lapisan atas, tetapi juga oleh masyarakat lapisan menengah dan lapisan bawah. PT. XYZ merupakan lembaga perbankan yang melayani pengajuan pinjaman bagi pelaku UMKM (Usaha Menegah Kecil dan Mikro), serta pelayanan simpanan tabungan dan deposito untuk semua kalangan masyarakat. Dalam proses bisnisnya PT. XYZ telah menerapkan teknologi informasi, termasuk pada layanan pengajuan simpan pinjam-nya. Sistem pengajuan simpan pinjam PT.XYZ merupakan sistem berbasis web yang melayani calon debitur untuk melakukan pengajuan pinjaman dan simpanan tanpa harus mendatangi kantor dan dapat meminimalisir terjadinya kesalahan pencatatan. Namun PT. XYZ memiliki beberapa sistem yang digunakan dengan fungsi yang berbeda, hal ini membuat sistem lain

sulit mengakses data pengajuan simpan pinjam dan data menjadi tidak terpusat. Belum ada teknologi pada sistem pengajuan simpan pinjam yang dapat mengintegrasikan sistem dengan sistem lain sehingga sulit untuk melakukan pertukaran, integrasi dan pengelolaan data. Maka diperlukan sebuah pengembangan menggunakan *web service* agar data yang tersimpan pada sistem pengajuan simpan pinjam dapat diakses dan diintegrasi dengan sistem lain PT. XYZ. Dengan adanya *web service* dapat menyelesaikan masalah dalam pertukaran data, integrasi data dan pengelolaan data (Firdaus et al., 2019).

Web service merupakan aplikasi penghubung antara *front end* dengan *back end* dan dapat berkomunikasi dengan software lain, yang dijalankan di *web serve*r untuk memenuhi kebutuhan bisnis. REST *web service* atau RESTful API adalah *web service* yang mengimplementasikan arsitektur REST. REST merupakan teknologi *web service* yang sering diterapkan karena memiliki performa yang baik dalam pertukaran dan komunikasi data berupa JSON atau XML yang diakses melalui protocol HTTP (Hypertext Transfer Protocol). Gaya arsitektur REST adalah *client-server*, klien dapat mengirim *request* ke server kemudian server akan memproses dan memberikan tanggapan yang diidentifikasi melalui URI (Uniform Resource Identifiers)(Mumbaikar & Padiya, 2013). Untuk mengambil, membuat, mengubah dan menghapus data, REST *web services* menggunakan http methods GET, PUT, POST dan DELETE(Chen et al., 2017).

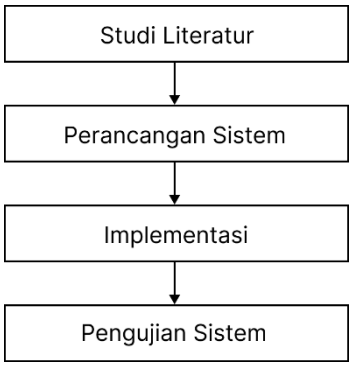
Dalam penelitian sebelumnya yang berjudul "*API REST Web service and backend system Of Lecturer's Assessment Information System on Politeknik Negeri Bali*", dibangun sistem informasi penilaian dosen menggunakan *rest web service* berdasarkan survei kebiasaan pengguna. *Web service* penilaian dosen berbasis RestFull API memanfaatkan MySQL server sebagai aplikasi pengelolaan data dengan konsep DBMS dan menggunakan bahasa pemrograman PHP versi 7.0 . Dihasilkan aplikasi data *web service* API yang dapat dikembangkan dengan menggunakan beberapa platform seperti desktop, web, atau mobile(Manuaba & Rudiastini, 2018). Pada penelitian "Analisis Dan Perancangan *Representational State Transfer (Rest)* Web Service Sistem Informasi Akademik STT Terpadu Nurul Fikri Menggunakan Yii *Framework*", dihasilkan rancangan model REST *web service* berbasis teknologi REST menggunakan Yii Framework 2.0 yang menyediakan layanan bagi sistem lain yang membutuhkan juga menawarkan kemudahan dalam menjembatani pertukuran data tanpa mempermasalahkan perbedaan platform, ataupun bahasa pemrograman(Agus Arianto et al., 2016). Penelitian lainnya yang berjudul "Implementasi Teknologi *Restful Web Service* Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi

Pertanian Universitas Brawijaya)", membahas tentang sebuah sistem informasi untuk melakukan perekaman prestasi mahasiswa di Fakultas Teknologi Pertanian. Dalam penelitian ini sistem dikembangkan dengan teknologi *web service* agar sistem dapat dijalankan di berbagai platform dan dapat diakses oleh perusahaan luar sebagai *cross check* ketika ada pelamar dari Fakultas Teknologi Pertanian. *Web service* menggunakan arsitektur REST dan diimplementasi menggunakan *micro-framework Lumen* sedangkan bagian *front end* diimplementasi menggunakanan Laravel. Dilakukan pengujian menggunakan blackbox testing dengan presentase hasil pengujian sebesar 100% yang menandakan sistem telah sesuai dengan spesifikasi (Wardhana et al., 2020).

Berdasarkan latar belakang masalah, didapatkan rumusan masalah yang melatar belakangi penelitian yaitu, bagaimana menerapkan *web service* pada sistem pengajuan simpan PT. XYZ agar dapat diintegrasikan dengan sistem lain yang membutuhkan data dari sistem pengajuan simpan pinjam PT. XYZ. Oleh karena itu dalam penelitian ini dibuat suatu rancangan REST *web service* pada sistem pengajuan simpan pinjam PT. XYZ, dengan fokus dari penelitian adalah aplikasi *back-end*. Dengan menerapkan *web service*, data pada sistem pengajuan simpan pinjam nantinya dapat diakses atau dapat terintegrasi oleh sistem atau aplikasi lain PT. XYZ yang membutuhkan data sistem pengajuan simpan pinjam.

METODE

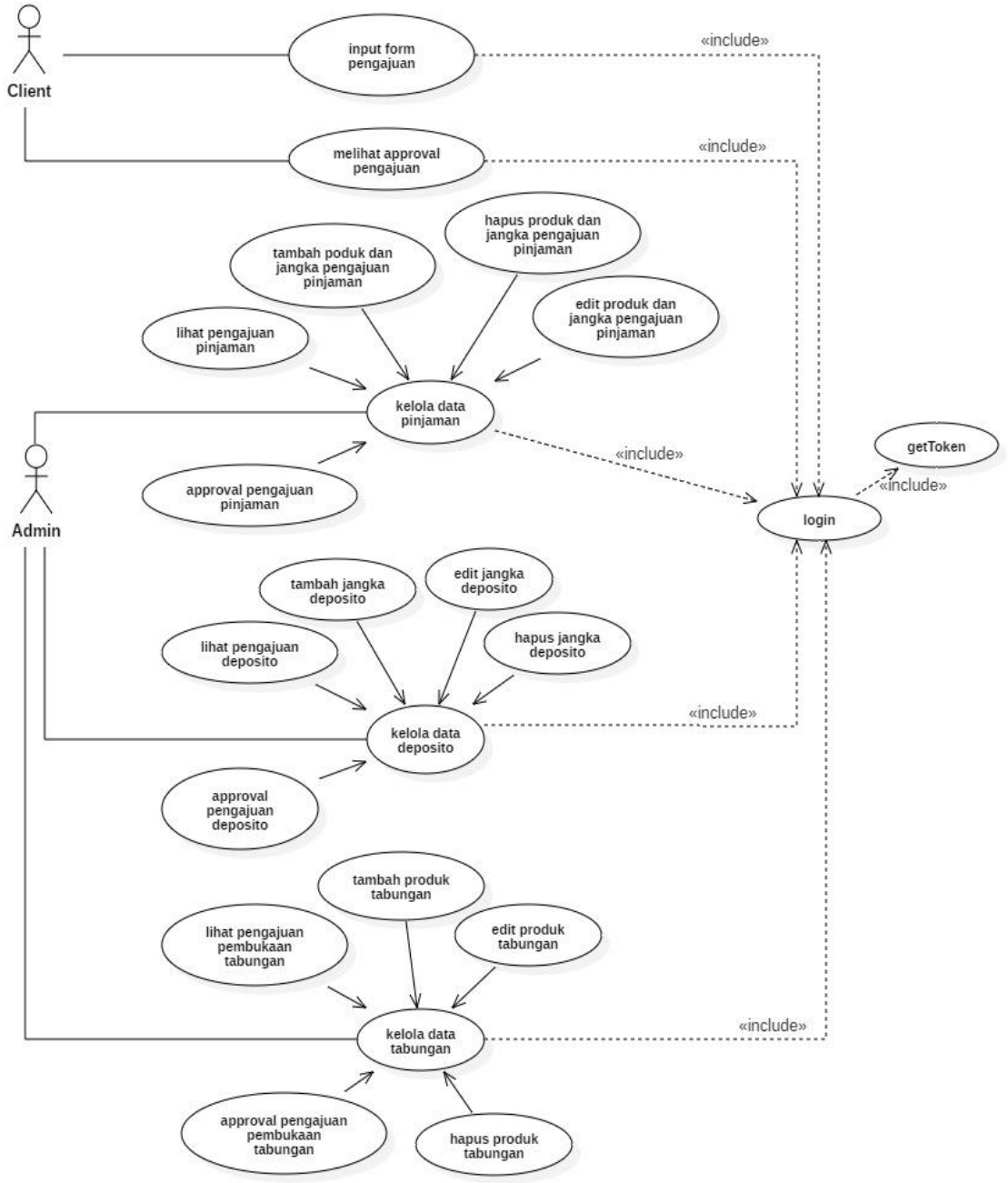
Di dalam metode penelitian akan dilakukan dengan metode *waterfall*, beberapa tahapan ialah studi literatur, perancangan sistem, implementasi serta pengujian sistem.



Gambar 1. Metode penelitian *Waterfall*

Pada tahap awal dilakukan studi literatur sebagai acuan di dalam penelitian. Studi literatur didapatkan melalui jurnal penelitian serta artikel terdahulu yang menunjang dan berkaitan dengan kasus penelitian. Dilanjutkan tahap perancangan sistem, rancangan sistem dibuat berlandaskan kebutuhan yang dibutuhkan PT. XYZ dalam wujud model UML

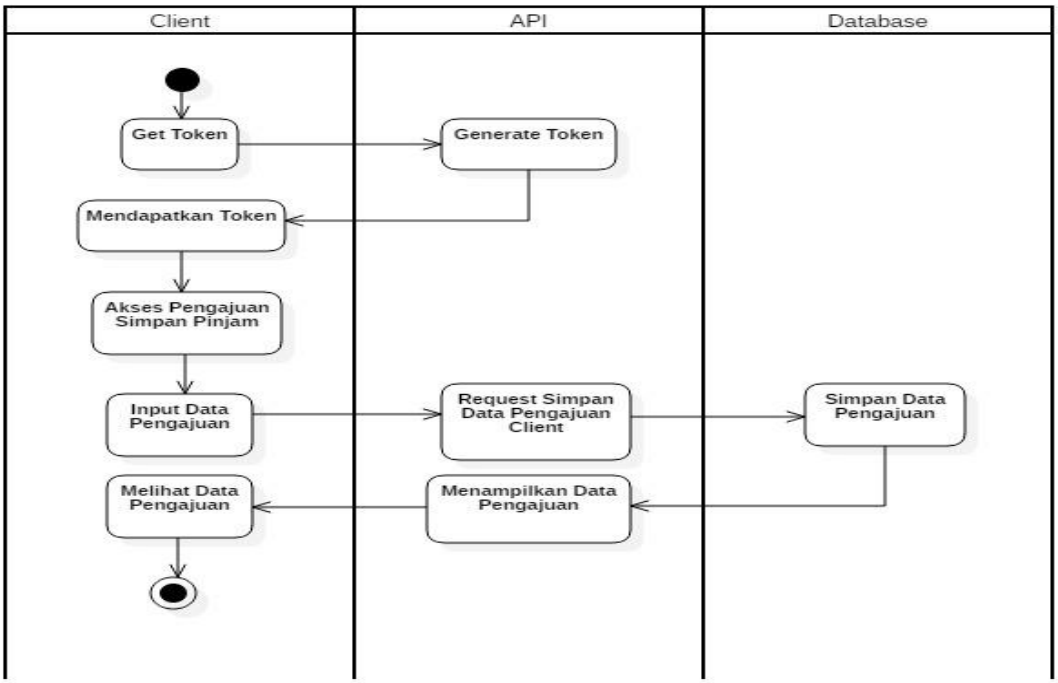
(*Unified Modeling Language*). Wujud pemodelan dari rancangan sistem berbentuk *use case diagram, activity diagram* dan *class diagram*.



Gambar 2. *Use Case Diagram*

Use case diagram adalah gambaran interaksi sistem dengan aktor pada suatu sistem atau aplikasi. Gambar 2 merupakan *Use Case Diagram* sistem pengajuan simpan pinjam yang menjelaskan bagaimana sistem berjalan. Sistem memiliki 2 aktor yang berperan yaitu *client* dan admin. *Client* merupakan user yang akan melakukan pengajuan simpan pinjam dan dapat melihat status *approval* yang diajukan *client* itu sendiri. Sedangkan admin merupakan user yang memiliki hak lebih tinggi dari *user client*. Pada *Use Case Diagram*

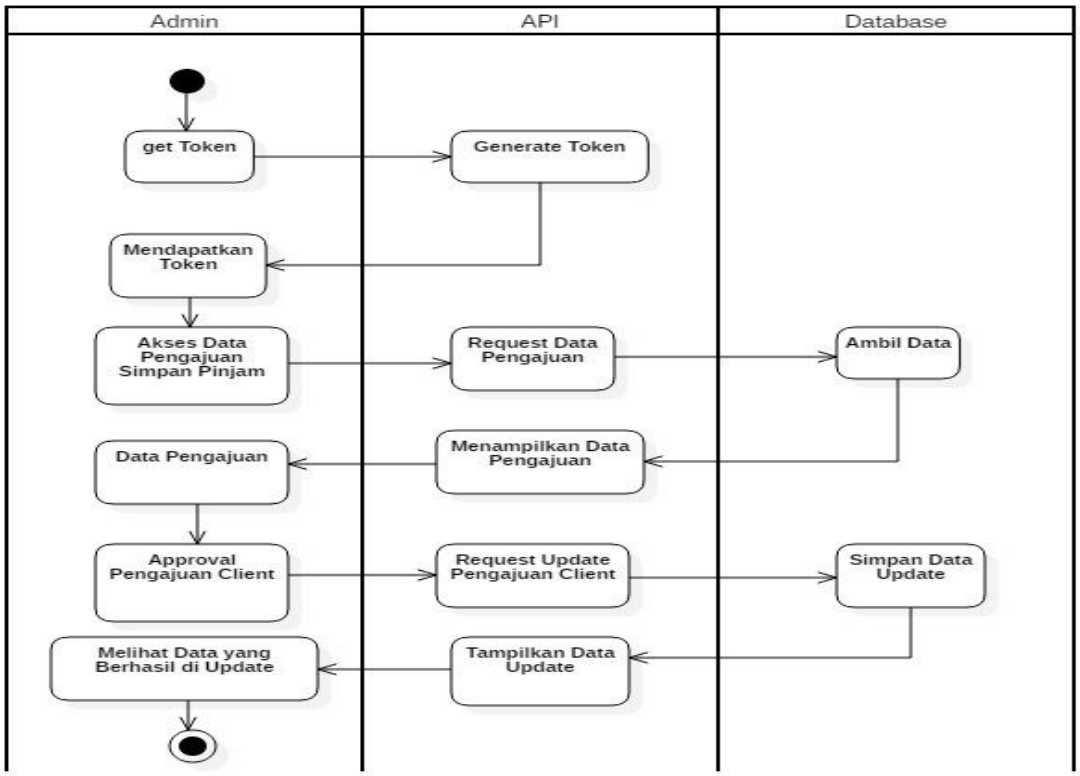
admin dapat mengakses, mengatur dan mengelola data sistem pengajuan simpan pinjam PT. XYZ.



Gambar 3. *Activity Diagram Client*

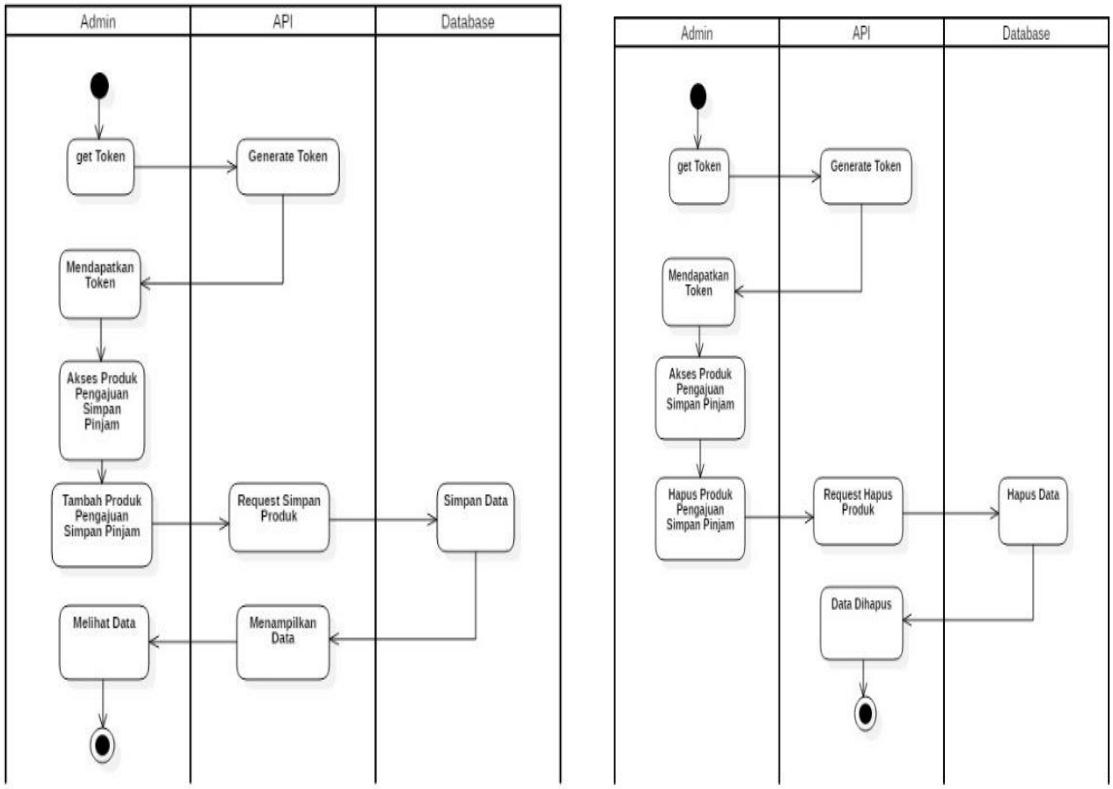
Activity diagram merupakan model aktivitas atau proses yang ada didalam sistem. Pada gambar 3 merupakan *Activity Diagram* yang menjelaskan proses saat *client* melakukan pengajuan simpan pinjam. *Client* akan mendapatkan token untuk mengakses pengajuan simpan pinjam setelah login. Dalam melakukan pengajuan, client melakukan input data pengajuan, API akan mengirimkan *request create* (*post*) yang kemudian disimpan didalam database dan ditampilkan kembali kepada *client*.

Gambar 4 adalah *Activity Diagram* yang menjelaskan bagaimana proses admin melakukan *update* data untuk memberikan *approval* kepada *client* yang melakukan pengajuan simpan pinjam. Admin login dan mendapatkan token untuk autentikasi admin. Admin dapat mengakses data pengajuan *client* dan API mengirim *request update* (*put*) data untuk *approval* pengajuan. Berdasarkan id pengajuan yang dibuat client, database akan menyimpan dan menampilkan data yang telah diupdate.

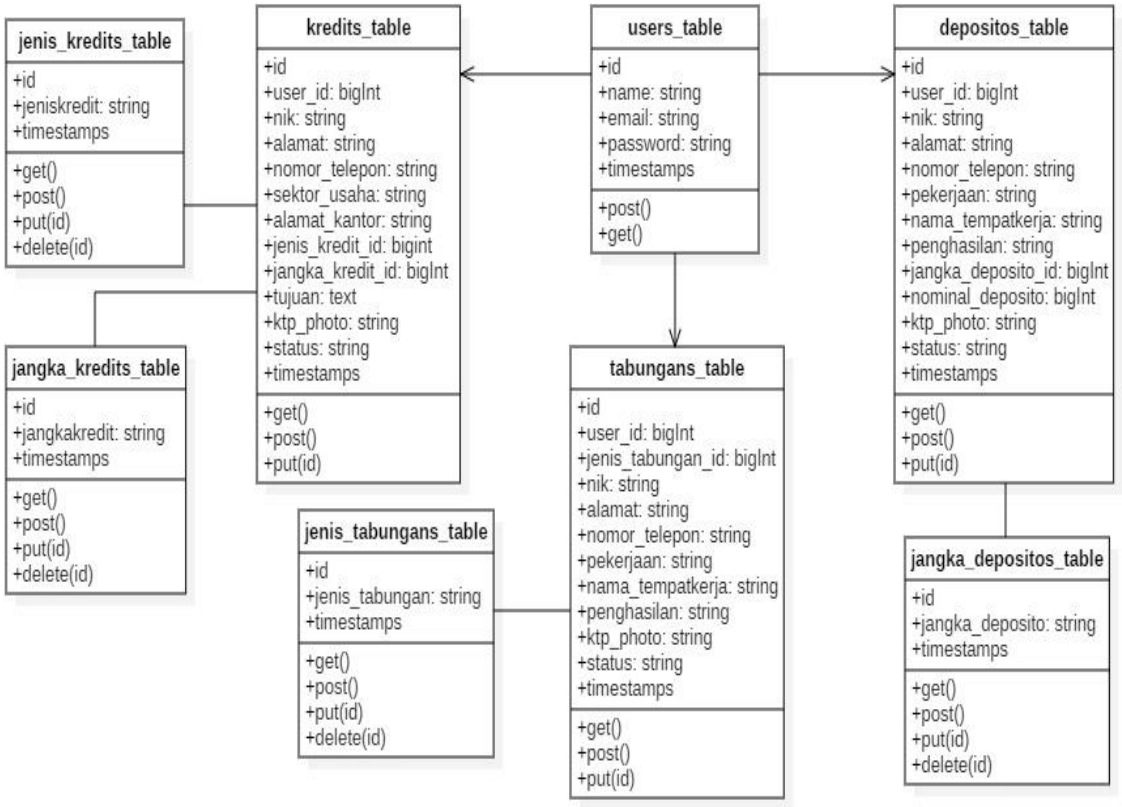


Gambar 4. *Activity Diagram* Admin – *Update*

Admin memiliki akses untuk menambah dan menghapus jenis produk dan jangka pengajuan pada sistem pengajuan simpan pinjam yang ditunjukan pada gambar 5.



Gambar 5. *Activity Diagram* Admin - *Create & Delete*



Gambar 6. *Class Diagram*

Gambar 6 merupakan *Class Diagram* dari sistem pengajuan simpan pinjam PT. XYZ yang menjelaskan hubungan method, atribut dan kelas yang saling berkaitan.

Tahap implementasi sistem dibuat menurut model perancangan sistem. Implementasi dirancang dengan arsitektur REST yang dibangun dengan *framework* Lumen serta autentikasi keamanan menggunakan JSON *Web Token* (JWT). Lumen merupakan *micro-framework* yang dikembangkan oleh Taylor Otwell yang berfokus untuk membuat, dan membangun API dengan fitur yang lebih ringkas serta mempunyai performa yang lebih baik daripada API *framework* PHP sejenis dalam mengatasi permintaan (*request*), sehingga sesuai untuk kebutuhan penelitian ini(Satrya & Kusumah, 2018). Penggunaan JSON Web Token pada web service bertujuan sebagai autentikasi untuk mengakses tiap *end point web service.* JSON Web Token adalah sebuah token berbentuk string JSON yang ringkas yang digunakan untuk sistem autentikasi dan pertukaran informasi. Token JWT dapat ditransmisikan dengan cepat melalui URL, parameter HTTP POST atau di dalam *Header* HTTP(Rahmatulloh et al., 2018).

Selanjutnya pada tahap pengujian sistem akan dilakukan pengujian fungsi dari tiap *end-point* yang dibuat guna mencari tahu apakah sistem berjalan dengan baik dan dapat memberikan *response* saat *request* dijalankan. Pengujian dilakukan dengan metode *blackbox* serta memakai dukungan aplikasi Postman. Metode *blackbox* adalah pengujian yang digunakan untuk mengetahui hasil input dan output dari perangkat lunak apakah berfungsi sesuai dengan yang diharapkan tanpa mengetahui struktur dari perangkat lunak tersebut(Wahyu Setiawan, 2011).

HASIL DAN PEMBAHASAN

Setelah tahap perancangan sistem, dilakukan implementasi dan pengujian sistem sesuai dengan perancangan sistem yang dibuat. Dalam penelitan ini, implementasi dilakukan menggunakan *micro-framework* Lumen dengan bahasa pemrograman PHP serta menambahkan library *JSON WebToken* (JWT) sebagai autentikasi keamanan *web service*.

```php
<?php
return [
    'defaults' => [
        'guard' => 'api',
        'passwords' => 'users',
    ],

    'guards' => [
        'api' => [
            'driver' => 'jwt',
            'provider' => 'users',
        ],
    ],

    'providers' => [
        'users' => [
            'driver' => 'eloquent',
            'model' => \App\Models\User::class
        ]
```

Kode Program 1. config/auth.php

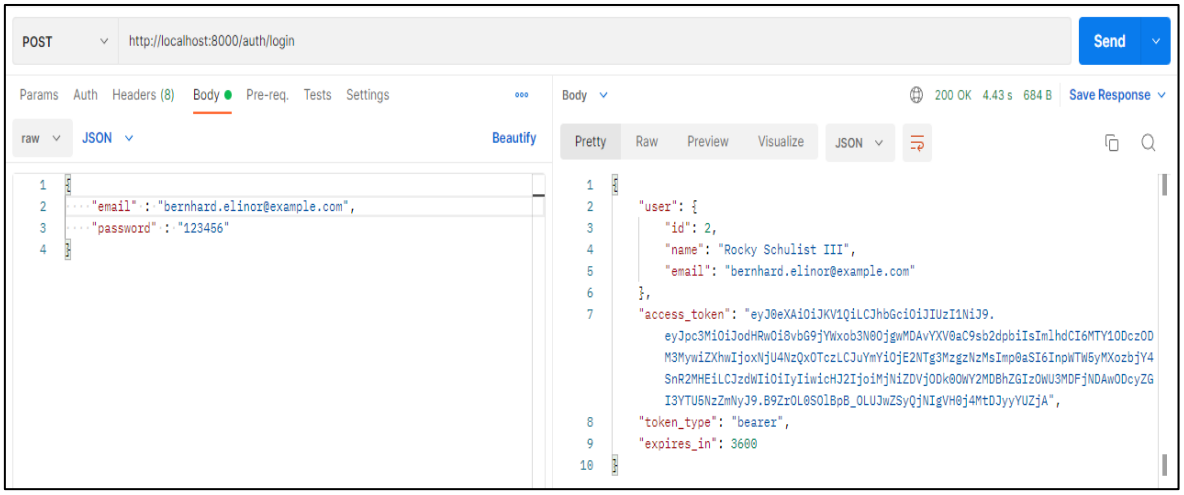
Pada kode program 1 config/auth.php menjelaskan konfigurasi kepada sistem yang dibuat untuk menggunakan JWT sebagai default penjaga saat autentikasi aktor, yang dijelaskan pada gambar 8.

```php
public function login(Request $request)
{
    $email = $request->email;
    $password = $request->password;

    if (empty($email) or empty($password)) {
        return ResponseFormatter::error
        ['status' => 'error', 'message' => 'You must fill all the fields']);

    $credentials = request(['email', 'password']);

    if (!$token = auth()->attempt($credentials)) {
        return ResponseFormatter::error(['error' => 'Unauthorized'], 401);
    }

    Return $this->respondWithToken($token);
}

protected function respondWithToken($token)
    {
        return response()->json([
            'user' => Auth::user(),
            'access_token' => $token,
            'token_type' => 'bearer',
            'expires_in' => Auth::factory()->getTTL() * 60
        ]);
```

Kode Program 2. *Controller/Authcontroller*

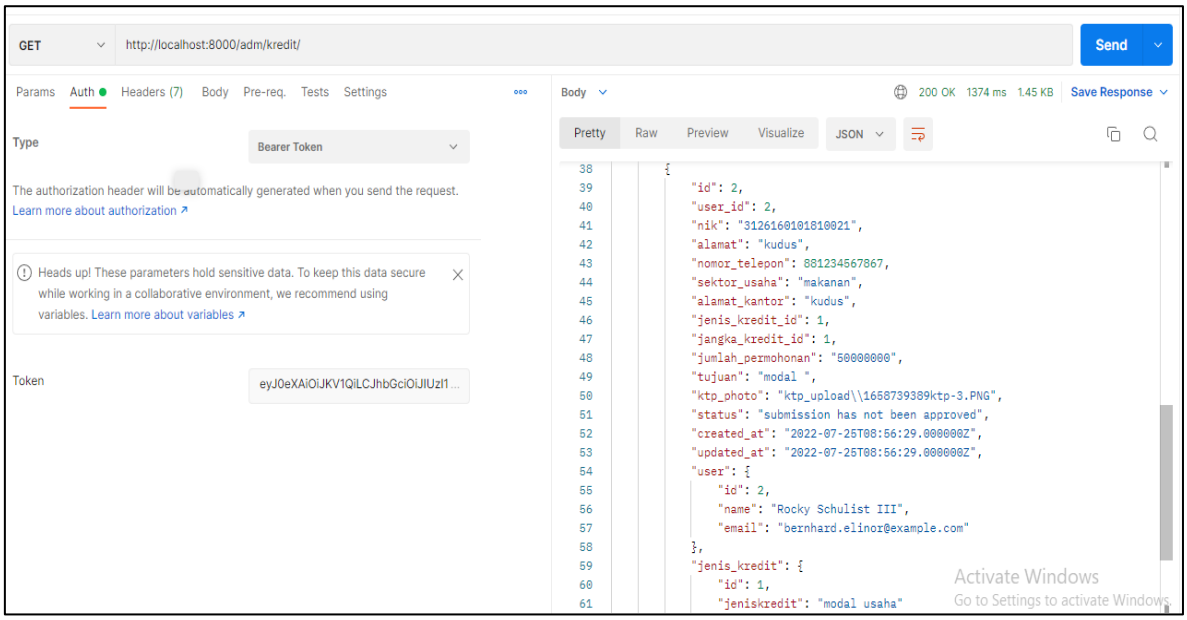
Kode program 2 merupakan *function login* dari *controller AuthController* pada sistem pengajuan simpan pinjam. JWT bekerja saat aktor berhasil melakukan login kemudian server akan mengirim respon berupa *bearer token*. Token akan tersimpan didalam database dan digunakan untuk mengakses *web service* tertentu. Pada kode program, token

yang dimiliki oleh aktor memiliki batas waktu 60 menit, ketika melewati batas waktu token tidak dapat lagi digunakan dan saat login kembali aktor akan mendapatkan token yang baru.



Gambar 7. Pengujian *Postman - Login*

Gambar 7 merupakan hasil pengujian login menggunakan Postman, didapatkan response JSON ketika aktor berhasil melakukan login. User mendapat akses token berupa bearer dengan batas waktu 3600 (60 menit).



Gambar 8. Pengujian Postman - Admin GET Data Pengajuan

Gambar 8 merupakan hasil pengujian *web service* menggunakan *postman* dengan method GET dan URI /adm/kredit/ pada aktor admin. Untuk mengakses dan mendapatkan seluruh data pengajuan kredit, admin harus mengirimkan Token berupa bearer token yang didapat pada saat login sebagai bukti autentikasi user ini sudah melakukan login.

```
class Kredit extends Model
{
    protected $fillable = [
        'nik',
        'alamat',
        'nomor_telepon',
        'sektor_usaha',
        'alamat_kantor',
        'jenis_kredit_id',
        'jangka_kredit_id',
        'jumlah_permohonan',
        'tujuan',
        'ktp_photo'
    ];

    public function user()
    {
        return
         $this->belongsTo(User::class);
    }
```

Kode Program 3. Model/Kredit

Kode program 3 adalah model dari Kredit, penggunaan *array fillable* berfungsi untuk menampung field dari tabel Kredit. Pada 3 baris terakhir Kredit memiliki relasi dengan User yang ditunjukan juga pada model User (kode program 4), yang menjelaskan bahwa satu client memiliki satu pengajuan kredit *One-to-One*.

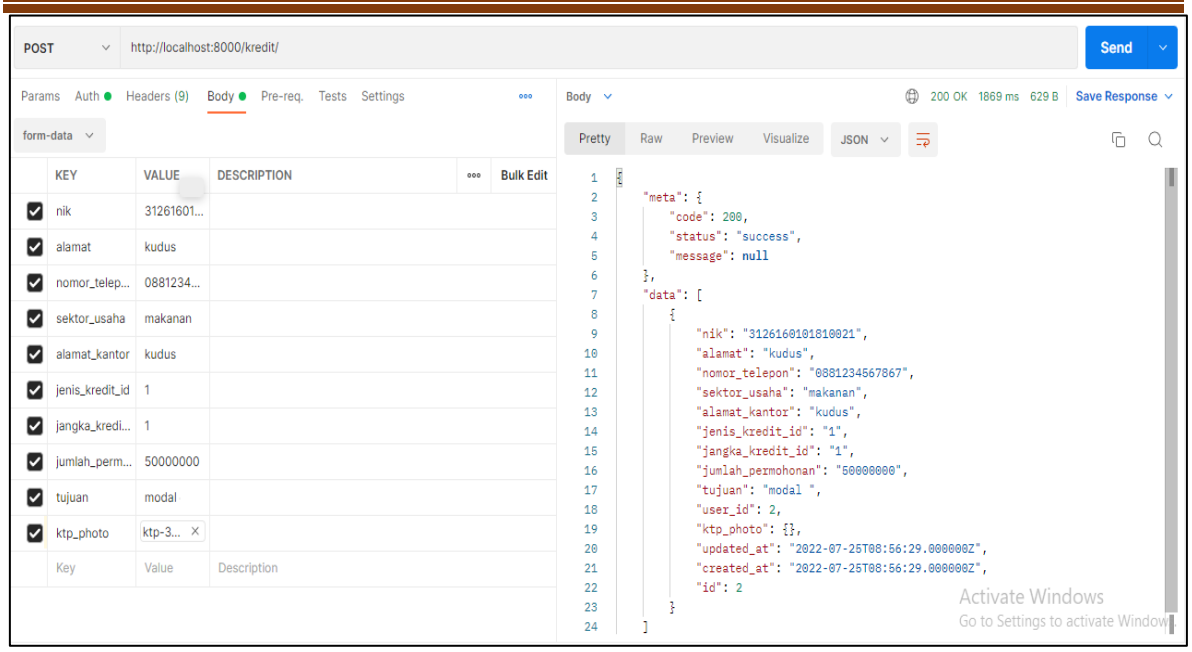
```
public function kredit()
{
    return $this->hasOne(Kredit::class);
}
```

Kode Program 4. Model/User

Kode program 5 merupakan *router* yang menangani *request* aktor berdasarkan *method*, URI, dan *function* yang akan dieksekusi. Pada baris pertama *middleware* digunakan sebagai pembatas autentikasi sesuai roles atau hak akses (*client* atau *admin*). Baris ketiga menjelaskan request client menggunakan method POST untuk membuat dan menyimpan data baru kredit (uri: http://localhost:8000/kredit/) dengan memanggil *function store* pada *controller* KreditController.php, ditunjukan pengujian menggunakan *Postman* pada gambar 9.

```
$router->group(['prefix' => 'kredit', 'middleware' => 'auth'],
            function () use ($router) {
    $router->post('/', 'KreditController@store');
});
```

Kode Program 5. *routes/web.php*



Gambar 9. Pegujian *Postman* - *Client* POST Pengajuan Kredit

Gambar 9 merupakan hasil pengujian pengajuan kredit menggunakan *Postman*. Dengan menggunakan method POST dan memanggil URI "http://localhost:8000/kredit/", data yang diinput akan di eksekusi sesuai *function* yang dipanggil pada *Route* dan tersimpan di dalam database. Didapatkan response code:200 berupa JSON sesuai dengan *key* dan *value* yang di input pada *Body*.

Tahap selanjutnya dilakukan pengujian pada setiap *endpoint* untuk mengetahui kesesuaian *response* yang dihasilkan, pengujian dilakukan dengan menggunakan metode *blackbox*, ditunjukan pada tabel 1.

Tabel 1. Pengujian *Blackbox*

Route	URI	Method	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
login	http://localhost:8000/auth/register	POST	menambahkan akun untuk user baru	user teregistrasi dan mendapatkan token.	Valid
	http://localhost:8000/auth/login	POST	mendapatkan token dengan melakukan login	user berhasil login dan mendapatkan akses token, jika gagal menampilkan respon unauthorized.	Valid
kredit	http://localhost:8000/kredit/	POST	menambahkan dan mengirim data pengajuan user	data berhasil ditambahkan dan disimpan di dalam database	Valid
	http://localhost:8000/kredit/pengajuan	GET	menampilkan data yang diajukan user	data berhasil ditampilkan	Valid

	http://localhost:8000/adm/kredit/	GET	menampilkan seluruh data pengajuan kredit kepada admin	seluruh data berhasil diambil dan ditampilkan	Valid
	http://localhost:8000/adm/kredit/approve/{id}	PUT	mengubah status pengajuan berdasarkan id pengajuan	data berhasil diubah	Valid
kredit/produk	http://localhost:8000/adm/kredit/produk	POST	dapat menambahkan jenis kredit	jenis kredit baru berhasil ditambhkan	Valid
	http://localhost:8000/adm/kredit/produk/{id}	PUT	mengubah data jenis kredit	jenis kredit berhasil diubah dan menampilkan jenis kredit yang terlah diubah	Valid
	http://localhost:8000/adm/kredit/produk/{id}	DELETE	menghapus jenis kredit	data jenis kredit berhasil dihapus	Valid
	http://localhost:8000/adm/kredit/produk	GET	dapat mengambil dan menampilkan data jenis kredit	data berhasil ditampilkan	Valid
kredit/term	http://localhost:8000/adm/kredit/term	POST	dapat menambahkan jangka kredit	jangka kredit baru berhasil ditambahkan dan ditampilkan	Valid
	http://localhost:8000/adm/kredit/term/{id}	PUT	mengubah data jangka kredit	jangka kredit berhasil diubah dan menampilkan jenis kredit yang terlah diubah	Valid
	http://localhost:8000/adm/kredit/term/{id}	DELETE	menghapus jangka kredit yang diinginkan	data jangka kredit berhasil dihapus	Valid
	http://localhost:8000/adm/kredit/term	GET	dapat mengambil dan menampilkan data jangka kredit	data berhasil ditampilkan	Valid
tabungan	http://localhost:8000/tabungan/	POST	menambahkan dan mengirim data tabungan user	data berhasil ditambahkan dan disimpan di dalam database	Valid
	http://localhost:8000/tabungan/pengajuan	GET	menampilkan data yang diajukan user	data berhasil ditampilkan	Valid
	http://localhost:8000/adm/tabungan	GET	menampilkan seluruh data pengajuan tabungan kepada admin	seluruh data berhasil diambil dan ditampilkan	Valid
	http://localhost:8000/adm/tabungan/approve/{id}	PUT	mengubah status pengajuan berdasarkan id pengajuan	data berhasil diubah	Valid
tabungan/produk	http://localhost:8000/adm/tabungan/produk	POST	dapat menambahkan jenis tabungan	jenis kredit baru berhasil ditambhkan	Valid
	http://localhost:8000/adm/tabungan/produk/{id}	PUT	mengubah data produk tabungan	produk tabungan berhasil diubah dan menampilkan produk tabungan yang telah diubah	Valid
	http://localhost:8000/adm/tabungan/produk/{id}	DELETE	menghapus produk tabungan	data produk tabungan berhasil dihapus	Valid

	http://localhost:8000/adm/tabungan/produk	GET	dapat mengambil dan menampilkan data jenis kredit	data berhasil ditampilkan	Valid
deposito	http://localhost:8000/deposito/	POST	menambahkan dan mengirim data deposito user	data berhasil ditambahkan dan disimpan di dalam database	Valid
	http://localhost:8000/deposito/pengajuan	GET	menampilkan data yang diajukan user	data berhasil ditampilkan	Valid
	http://localhost:8000/adm/deposito	GET	menampilkan seluruh data pengajuan deposito kepada admin	seluruh data berhasil diambil dan ditampilkan	Valid
	http://localhost:8000/adm/deposito/approve/{id}	PUT	mengubah status pengajuan berdasarkan id pengajuan	data berhasil diubah	Valid
deposito/term	http://localhost:8000/adm/deposito/term	POST	dapat menambahkan jangka deposito	jangka deposito baru berhasil ditambhkan	Valid
	http://localhost:8000/adm/deposito/term/{id}	PUT	mengubah data jangka deposito	jangka deposito berhasil diubah dan menampilkan jangka deposito yang telah diubah	Valid
	http://localhost:8000/adm/deposito/term/{id}	DELETE	menghapus jangka deposito	data jangka deposito berhasil dihapus	Valid
	http://localhost:8000/adm/deposito/term	GET	dapat mengambil dan menampilkan data jangka deposito	data berhasil ditampilkan	Valid

KESIMPULAN DAN REKOMENDASI

Berdasarkan implementasi dan pengujian blackbox yang dilakukan pada seluruh endpoint menggunakan Postman, dihasilkan web service Pengajuan Simpan Pinjam PT. XYZ dengan teknologi arsitektur REST menggunakan micro-framework Lumen yang menghasilkan output dengan bentuk data JSON. REST web service dilengkapi dengan keamanan autentikasi token JSON Web Token untuk mengakses setiap endpoint sehingga web service menjadi lebih aman. Dengan rancangan web service pada sistem pengajuan simpan pinjam PT. XYZ, dapat dilakukan integrasi data dari sistem yang berbeda sehingga data menjadi terpusat.

REFERENSI

Agus Arianto, M., Munir, S., & 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).

Chen, X., Ji, Z., Fan, Y., & Zhan, Y. (2017). Restful API Architecture Based on Laravel Framework. *Journal of Physics: Conference Series*, *910*(1). https://doi.org/10.1088/1742-6596/910/1/012016

Firdaus, A., Widodo, S., Sutrisman, A., Gading, S., Nasution, F., Mardiana, R., Komputer, J. T., Negeri, P., & Palembang, S. (2019). RANCANG BANGUN SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN WEB SERVICE PADA JURUSAN TEKNIK KOMPUTER POLSRI. *Jurnal Informanika*, *5*(2). www.kursuswebsite.org

Manuaba, I. B. P., & Rudiastini, E. (2018). API REST Web service and backend system of Lecturer's Assessment Information System on Politeknik Negeri Bali. *Journal of Physics: Conference Series*, *953*(1). https://doi.org/10.1088/1742-6596/953/1/012069

Mumbaikar, S., & Padiya, P. (2013). Web Services Based On SOAP and REST Principles. *International Journal of Scientific and Research Publications*, *3*(5)

Rahmatulloh, A., Sulastri, H., & Nugroho, R. (2018). Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512. In *JNTETI* (Vol. 7, Issue 2).

Satrya, F., & Kusumah, F. (2018). *RANCANG BANGUN WEB SERVICE UNTUK TRANSAKSI DATA PADA APLIKASI SAHABAT JASA DENGAN METODE REST*. 256–264.

Wahyu Setiawan, G. (2011). *SOFTWARE TESTING USING BLACK BOX METHOD ON SANATA DHARMA UNIVERSITY'S EXELSA*.

Wardhana, W. G., Arwani, I., & Rahayudi, B. (2020). Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer; Vol 4 No 2 (2020)*, *4*(2), 680–689