

## 19. BAB IV HASIL DAN PEMBAHASAN

### 1. Coding Phase

Pada tahapan ini merupakan tahapan implementasi *web service* BPM UII. Berikut tahapan yang harus dilalui dalam membangun *web service* dengan menggunakan *framework* Lumen.

#### 1. Membangun Database

Membangun *database* merupakan hal yang wajib dilakukan untuk membangun *web service*. Struktur tabel *database* yang akan dibangun akan disesuaikan dengan rancangan ERD yang sudah dilakukan pada tahap sebelumnya. ERD yang sudah dirancang akan dijadikan sebagai struktur tabel di *database* dengan menggunakan menu yang sudah disediakan oleh MySQLWorkbench.

#### 2. Membuat Endpoint

*Endpoint* akan diakses oleh aplikasi *client* yang untuk mendapatkan layanan. Kode program untuk membuat *endpoint* ditulis dalam *folder* */routes*. Setiap *endpoint* dibatasi akses menggunakan *middleware*. *Middleware* berfungsi sebagai verifikasi antara *request* yang masuk.

```
<?php
$router->group(['prefix' =>'api/v1','middleware' => ['jwt.auth','lvl']],
function() use ($router){
    $router->group(['prefix' =>'rekomendasi'], function() use($router){
        $router->post('setRekomendasi/{uuidButir}', ['uses'=>
'Auditee\RekomendasiTindakLanjutController@setRekomendasi',
'middleware' => ['cutoff','auditee']]);
        $router->get('getData/{uuidButir}',
'Auditee\RekomendasiTindakLanjutController@getData');
        $router->get('getData',
'Auditee\RekomendasiTindakLanjutController@getAllData');
    });
    $router->group(['prefix' =>'sasaranmutu'], function() use($router){
        $router->post('komentar/{uuidButir}',
['uses'=>'Auditee\SasaranMutuController@postKomentar',
```

```

'middleware'=>[ 'cutoff', 'auditor' ]]);

    $router->get ('komentar/{uuidButir}',
[ 'uses'=>'Auditee\SasaranMutuController@komentar',
'middleware'=>'cutoff' ]]);

    $router->post ('laporanAMI',
[ 'uses'=>'Auditee\SasaranMutuController@postLaporanTemuanAMI',
'middleware'=>'auditor' ]]);

    $router->get ('laporanAMI',
[ 'uses'=>'Auditee\SasaranMutuController@laporanAMI' ]]);

    $router->put ('selesai/{uuidButir}',
[ 'uses'=>'Auditee\SasaranMutuController@sasaranMutuSelesai',
'middleware'=>[ 'cutoff', 'auditor' ]]);

});

    $router->get ('validitas',
[ 'uses'=>'Auditee\SasaranMutuController@validitas',
'middleware'=> [ 'cutoff', 'auditee' ]]);

    $router->put ('validitas',
[ 'uses'=>'Auditee\SasaranMutuController@validitasUpdate',
'middleware' => [ 'cutoff', 'auditee' ]]);

    $router->get ('role/organisasiRole',
'OrganisasiRoleController@listOrganisasiRole');

});

```

Gambar 19.1 Routes Sasaran Mutu

Gambar 19.1 merupakan kode program dari *endpoint* yang berkaitan dengan penjaminan mutu. Setiap *endpoint* memiliki fungsi, metode HTTP dan *middleware* yang berbeda. Perbedaan dari setiap *endpoint* dapat dilihat pada Tabel 19.1.

Tabel 19.1 Endpoint Sasaran Mutu

Fungsi	Endpoint	Middleware	Routes
Menambahkan data rekomendasi tindak lanjut berdasarkan uuid butir	POST /setRekomendasi/ {uuidButir}	jwt.auth, lvl, cutoff, dan auditee	\$router->post ( 'setRekomendasi/{uuidButir}' , [ 'uses'=>'Auditee\RekomendasiTindakLanjutController@setRekomendasi', 'middleware' => [ 'cutoff', 'auditee' ]]);
Menampilkan data	GET /getData/ {uuidButir}	jwt.auth dan lvl	\$router->get ( 'getData/{uuidButir}', 'Auditee\RekomendasiTindak

<b>Fungsi</b>	<b>Endpoint</b>	<b>Middleware</b>	<b>Routes</b>
rekomendasi tindak lanjut berdasarkan uuid butir			LanjutanController@getData') ;
Menampilkan semua data rekomendasi tindak lanjut	GET /getData	jwt.auth dan lvl	\$router->get('getData', 'Auditee\RekomendasiTindakLanjutanController@getAllData');
Menambahkan data komentar dari setiap butir berdasarkan uuid butir	POST /komentar/{uuidButir}	jwt.auth, lvl, cutoff, dan auditor	\$router->post('komentar/{uuidButir}', ['uses'=>'Auditee\SasaranMutuController@postKomentar', 'middleware'=>['cutoff', 'auditor']]);
Menambahkan data laporan temuan AMI	POST /laporanAMI	jwt.auth, lvl, dan auditor	\$router->post('laporanAMI', ['uses'=>'Auditee\SasaranMutuController@postLaporanTemuanAMI', 'middleware'=>'auditor']);
Menampilkan data laporan temuan AMI	GET /laporanAMI	jwt.auth dan lvl	\$router->get('laporanAMI', ['uses'=>'Auditee\SasaranMutuController@laporanAMI']);
Mengubah status selesai sasaran mutu dari setiap butir berdasarkan uuid butir	PUT /selesai/{uuidButir}	jwt.auth, lvl, cutoff, dan auditor	\$router->put('selesai/{uuidButir}', ['uses'=>'Auditee\SasaranMutuController@sasaranMutuSelesai', 'middleware'=>['cutoff', 'auditor']]);
Menampilkan data yang akan di validasi oleh Auditee	GET /validitas	jwt.auth, lvl, cutoff, dan auditee	\$router->get('validitas', ['uses'=>'Auditee\SasaranMutuController@validitas', 'middleware'=>['cutoff', 'auditee']]);

Fungsi	Endpoint	Middleware	Routes
Memvaliditas data sasaran mutu	PUT /validitas	jwt.auth, lvl, cutoff, dan auditee	<code>\$router-&gt;put('validitas', ['uses'=&gt;'Auditee\SasaranMutuController@validitasUpdate','middleware'=&gt;['cutoff','auditee']]);</code>
Menampilkan data organisasi dengan role yang dimiliki	GET /role/organisasiRole	jwt.auth dan lvl	<code>\$router-&gt;get('role/organisasiRole','OrganisasiRoleController@listOrganisasiRole');</code>

Setiap *endpoint* memilikidua parameter yaitu alamat *endpoint* itu sendiri dan untuk parameter kedua, mempunyai dua kemungkinan yaitu pertama nama kelas *controller* serta fungsinya saja atau nama kelas *controller* dengan fungsi beserta dengan *middleware*.

### 3. Membuat Model

Model berfungsi sebagai penghubung interaksi dengan database. Setiap tabel memiliki model tersendiri. Model pada web service BPM UII akan diletakkan di folder /app/models.

```
<?php
namespace App\Models;

use Illuminate\Auth\Authenticatable;
use Laravel\Lumen\Auth\Authorizable;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Contracts\Auth\Authenticatable as AuthenticatableContract;
use Illuminate\Contracts\Auth\Access\Authorizable as AuthorizableContract;

use App\Models\Periodes;
use App\Models\Butirs;
use App\Models\RekomendasiTindakLanjuts;
use App\Models\Organisasis;

class SasaranMutus extends Model implements AuthenticatableContract,
AuthorizableContract{
...

    public function periode(){
        return $this->belongsTo(new Periodes,'periodes_id');
    }

    public function butir(){
        return $this->belongsTo(new Butirs,'butirs_id');
```

```

    }

    public function rekomendasiTindakLanjut() {
        return $this->hasMany(new RekomendasiTindakLanjuts);
    }

    public function organisasi() {
        return $this->hasMany(new Organisasi);
    }
}

```

Gambar 19.2 Contoh Models Sasaran Mutu

Gambar 19.2 merupakan contoh model dari tabel sasaran\_mutus dan turunan dari kelas model. Di model SasaranMutus memiliki 4 *method function* yang berfungsi sebagai relasi antar model tabel lain. Empat *method* tersebut yaitu:

1. *Method* *periodes* yaitu model SasaranMutus akan berelasi dengan model *periodes*. *Method* *periodes* digunakan sebagai merujuk kepada model tabel *periodes* dengan nilai kembalian *method belongsTo*.

*Method* *butir* yaitu model SasaranMutus akan berelasi dengan model *butirs*. *Method* *butir* digunakan sebagai merujuk kepada model tabel *butirs* dengan nilai kembalian *method belongsTo*.

*Method* *rekomendasiTindakLanjut* yaitu model SasaranMutus akan berelasi dengan model *RekomendasiTindakLanjuts*. *Method* *rekomendasiTindakLanjut* berguna sebagai rujukan kepada model tabel *rekomendasi\_tindak\_lanjuts* dengan nilai kembalian *method hasMany*.

*Method* *organisasi* yaitu model SasaranMutus akan berelasi dengan model *Organisasi*. *Method* *organisasi* berguna sebagai rujukan kepada model tabel *organisasi* dengan nilai kembalian *method hasMany*.

## 1. Membuat Controller

*Controller* mengatur *request* yang dikirim oleh aplikasi klien dengan sebuah *endpoint*. *Controller* berisikan beberapa *method function* yang di dalamnya memiliki pengelolaan fungsionalitas berdasarkan *request* dari aplikasi klien.

```
<?php

namespace App\Http\Controllers\Auditee;
use Laravel\Lumen\Routing\Controller as BaseController;
use App\Models\SasaranMutus;
use App\Models\Butirs;
use App\Models\RekomendasiTindakLanjuts;
use App\Models\Periodes;
use App\Models\Komentars;
use App\Models\Organisasis;
use App\Models\LaporanTemuanAmis;
use App\Models\UraianTemuans;
use Helpers;

use \Illuminate\Http\Request;

class SasaranMutuController extends BaseController{

    public function validitas(Request $request){
    ...
    }

    public function validitasUpdate(Request $request){
    ...
    }

    public function postKomentar($uuidButir,Request $request){
    ...
    }

    public function postLaporanTemuanAMI (Request $request){
    ...
    }

    public function laporanAMI (Request $request){
    ...
    }

    public function sasaranMutuSelesai($uuidButir,Request $request){
    ...
    }
}
```

Gambar 19.3 Contoh *Controller* Sasaran Mutu

Pada Gambar 19.3 merupakan *controller* sasaran mutu. *SasaranMutuController* memiliki 6 *method* yang berisikan fungsionalitas yang berbeda – beda. Enam *method* tersebut yaitu:

1. *Method* validitas berfungsi sebagai mengembalikan data yang akan divalidasi oleh auditee.
2. *Method* validitasUpdate berfungsi sebagai mengubah status di tabel sasaran\_mutus yang berguna sebagai sasaran mutu sudah divalidasi oleh auditee.
3. *Method* postKomentar berfungsi sebagai menambahkan komentar di sasaran mutu setiap butir dari sasaran mutu dengan memiliki parameter uuid butir.
4. *Method* postLaporanTemuanAMI berfungsi sebagai menambahkan data laporan temuan AMI.
5. *Method* laporanAMI berfungsi sebagai mengembalikan data laporan AMI.
6. *Method* sasaranMutuSelesai berfungsi sebagai mengubah status di tabel sasaran yang berguna sebagai audit di butir sasaran mutu sudah selesai.

### 1. Test Phase

Pada tahapan ini merupakan pengujian dari setiap *endpoint* yang telah dibangun. Tujuan dilakukan pengujian agar memenuhi kebutuhan dari BPM UII dan kualitas peformas dari *web service*. Pada tahapan ini akan dilakukan 2 pengujian pada *web service* yang telah dibangun yaitu *endpoint testing* dan *endpoint performance testing*. Berikut merupakan *testing web service* BPM UII yang dapat dilihat pada Tabel 19.2.

Tabel 19.2 Testing web service BPM UII

No	Endpoint	Peran	Method HTTP	Status	Gambar	Sesuai/Tidak Sesuai
1	Login	Auditee, Auditor, dan Admin	POST	Berhasil	Gambar 19.4	Sesuai
			POST	Gagal	Gambar 19.5	Sesuai
2	List peran pengguna	Auditee, Auditor, dan Admin	GET	Berhasil	Gambar 19.6	Sesuai
			GET	Gagal	Gambar 19.7	Sesuai

No	Endpoint	Peran	Method HTTP	Status	Gambar	Sesuai/Tidak Sesuai
3	Sasaran Mutu	Auditee, Auditor dan Admin	GET	Berhasil	Gambar 19.8	Sesuai
			GET	Gagal	Gambar 19.9	Sesuai
4	Bukti Capaian	Auditee, Auditor, Admin	GET	Berhasil	Gambar 19.10	Sesuai
			GET	Gagal	Gambar 19.9	Sesuai
		Auditee	POST	Berhasil	Gambar 19.12	Sesuai
		Auditee	POST	Gagal	Gambar 19.13	Sesuai
		Auditee	DELETE	Berhasil	Gambar 19.14	Sesuai
		Auditee	DELETE	Gagal	Gambar 19.15	Sesuai
5	Rekomendasi tindak lanjut	Auditee, Auditor, Admin	GET	Berhasil	Gambar 19.16	Sesuai
		Auditee	GET	Gagal	Gambar 19.17	Sesuai
		Auditee	POST	Berhasil	Gambar 19.18	Sesuai
		Auditee	POST	Gagal	Gambar 19.19	Sesuai
		Auditee	DELETE	Berhasil	Gambar 19.20	Sesuai
		Auditee	DELETE	Gagal	Gambar 19.21	Sesuai
6	Keranjang validasi	Auditee	GET	Berhasil	Gambar 19.22	Sesuai



No	Endpoint	Peran	Method HTTP	Status	Gambar	Sesuai/Tidak Sesuai
		Auditee	GET	Gagal	Gambar 19.23	Sesuai
7	Memvalidasi Keranjang validasi	Auditee	PUT	Berhasil	Gambar 19.24	Sesuai
		Auditee	PUT	Gagal	Gambar 19.25	Sesuai
8	Laporan temuan AMI	Auditor	GET	Berhasil	Gambar 19.26	Sesuai
		Auditor	GET	Gagal	Gambar 19.27	Sesuai
		Auditor	POST	Berhasil	Gambar 19.28	Sesuai
		Auditor	POST	Gagal	Gambar 19.29	Sesuai
9	Grafik	Auditee, Auditor, Admin	GET	Berhasil	Gambar 19.30	Sesuai
		Auditee, Auditor, Admin	GET	Gagal	Gambar 19.31	Sesuai
10	Otorisasi	Auditee, Auditor, Admin	POST	Berhasil	Gambar 19.32	Sesuai
11	<i>Cut Off</i>	Auditee, Auditor	POST	Berhasil	Gambar 19.33	Sesuai
12	Peforma Endpoint	Auditor		Berhasil	Gambar 19.34	Sesuai
		Auditee		Berhasil	Gambar 19.36	Sesuai
		Admin		Berhasil	Gambar 19.38	Sesuai

### 1. Endpoint Testing

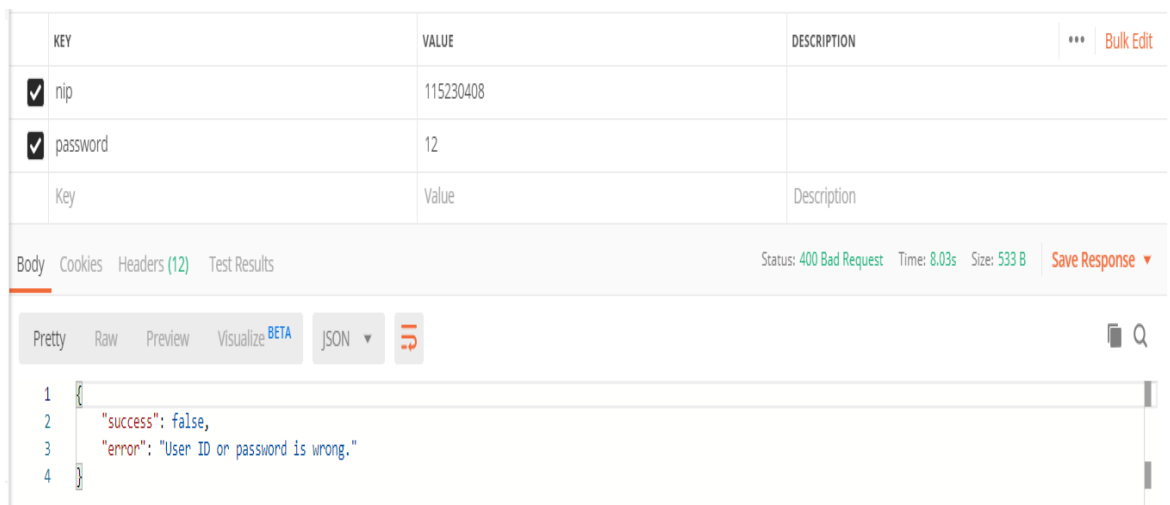
*Endpoint testing* merupakan pengujian dari *web service* terhadap *response* yang diberikan sesuai dengan yang diharapkan oleh pengguna berdasarkan *request* yang dikirimkan ke



### 5. Log In yang gagal

Jika pengguna mengisikan nip dan password yang tidak benar maka pengguna mendapatkan status *response code* “400”. Pada Gambar 19.5 merupakan struktur konten *response* dari *endpoint* “POST /auth/login” dengan berisikan:

1. *Success* yang berisi nilai *false*.
2. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.



Gambar 19.5 *Endpoint Log In* yang Gagal

### 3. Skenario tampil list peran pengguna

1. Tampil list peran pengguna yang berhasil

Diketahui pengguna ingin melakukan pemilihan peran untuk memasuki sistem BPM UII, maka akan mengirimkan *request* dengan *header* bernama *authorization* ke *endpoint* “GET /api/v1/level”. *Authorization* yaitu token yang didapatkan ketika pengguna melakukan *log in*. Setelah pengguna mengirimkan *request* jika berhasil maka pengguna mendapatkan status *response code* “200”. Pada Gambar 19.6 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/level” dengan berisikan:

1. *Success* yang berisi nilai *true*.
2. *Result* berisikan list peran pengguna sesuai lingkup aksesnya.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6Ikpzcm9udW11...	
Key	Value	Description

Temporary Headers (10)

Body Cookies Headers (12) Test Results Status: 200 OK Time: 8.87s Size: 2.06 KB Save Response

Pretty Raw Preview Visualize BETA JSON

```

1  {
2    "success": true,
3    "result": [
4      {
5        "nama_level": "Auditee",
6        "nama_organisasi": "Informatika",
7        "nama_akses": "Auditee - Informatika",
8        "uuid": "1af5756e-f56d-11e9-b95a-0cc47afc359e",
9        "uuid_cakupan": "3f77ddb9-f240-11e9-b95a-0cc47afc359e"
10     },
11     {
12       "nama_level": "Auditee",
13       "nama_organisasi": "TEKNOLOGI INDUSTRI",
14       "nama_akses": "Auditee - TEKNOLOGI INDUSTRI",
15       "uuid": "69959e5f-f570-11e9-b95a-0cc47afc359e",
16       "uuid_cakupan": "2b624b02-f240-11e9-b95a-0cc47afc359e"
17     },
18     {
19       "nama_level": "Auditee",
20       "nama_organisasi": "Arsitektur Program Internasional",
21       "nama_akses": "Auditee - Arsitektur Program Internasional",
22       "uuid": "1af5756e-f56d-11e9-b95a-0cc47afc359e"
23     }
24   ]
25 }

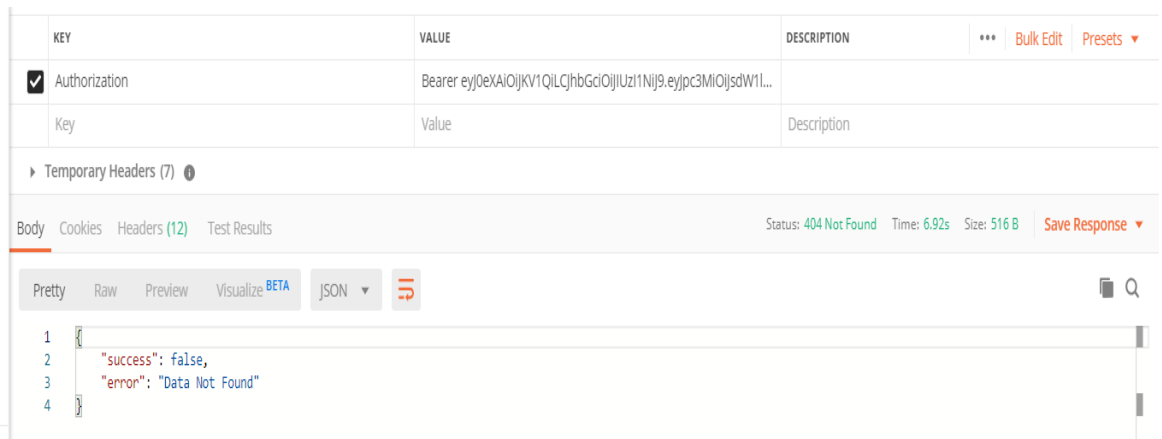
```

Gambar 19.6 Endpoint List Peran Pengguna yang Berhasil

### 1. List peran pengguna yang gagal

Jika pengguna tidak memiliki peran maka pengguna mendapatkan status *response code* “404”. Pada Gambar 19.7 merupakan struktur konten *response* dari endpoint “GET /api/v1/level” dengan berisikan:

1. *Success* yang berisi nilai *false*.
2. *Error* berisikan pesan dari pesan kesalahan yang dikirimkan dari server.



Gambar 19.7 Endpoint List Peran Pengguna yang Gagal

### 3. Skenario tampil data sasaran mutu

#### 1. Tampil data sasaran mutu yang berhasil

Diketahui pengguna ingin menampilkan data dari salah satu butir sasaran mutu ke dalam sistem BPM UII, maka akan mengirimkan data *request* ke endpoint “GET /api/v1/sasaranmutu/data/uuidButir” dengan tiga *header* yaitu

1. *Header authorization* yang didapatkan ketika *log in*.
2. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
3. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* jika berhasil maka pengguna mendapatkan status *response code* “200”. Pada Gambar 19.8 merupakan struktur konten *response* dari endpoint “GET /api/v1/sasaranmutu/data/uuidButir” dengan berisikan:

1. *Success* yang berisi nilai *true*.
2. *Result* berisikan koleksi data sasaran mutu sesuai lingkup aksesnya.

The screenshot shows the Headers section of a browser's developer tools. The 'Headers (3)' section is expanded, showing three headers: 'Authorization' (Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJsZW11...), 'access-uuid' (1af5756e-f56d-11e9-b95a-0cc47afc359e), and 'cakupan-uuid' (3f77ddb9-f240-11e9-b95a-0cc47afc359e). Below this, the 'Temporary Headers (7)' section is also visible. The main content area shows the response body in JSON format, which is a success response with a detailed result object.

```

1  {
2    "success": true,
3    "result": {
4      "id": 36,
5      "capaian": "3.45",
6      "nilai_akhir": "4",
7      "kategori_capaian": "sesuai",
8      "target": "3.25",
9      "status": null,
10     "standar": "",
11     "pegawais_id": 385,
12     "butirs_id": 6,
13     "organisasias_id": 90,
14     "perodes_id": 1,
15     "uuid": "8ad61d2f-1408-11ea-ba45-0cc47afc359e",
16     "created_at": "2019-12-01 07:02:32",
17     "updated_at": "2019-12-11 22:47:10",
18     "butin": {

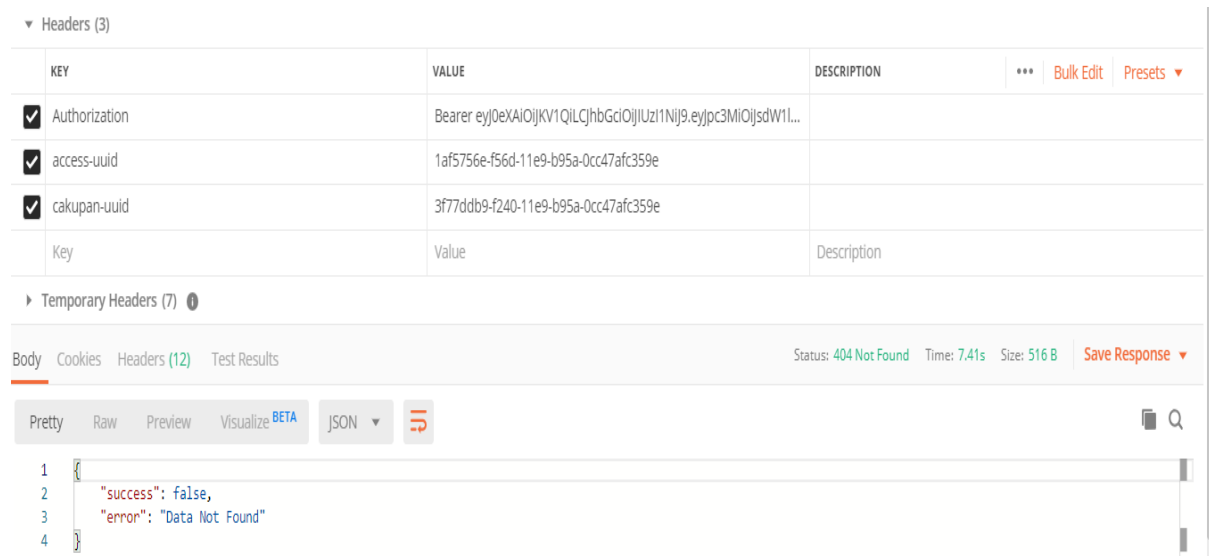
```

Gambar 19.8 *Endpoint* Menampilkan Sasaran Mutu yang Berhasil

### 1. Tampil data sasaran mutu yang gagal

Jika pengguna tidak memiliki data sasaran mutu maka pengguna mendapatkan status *response code* “404”. Pada Gambar 19.9 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/sasaranmutu/data/uuidButir” dengan berisikan:

1. *Success* yang berisi nilai *false*.
2. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.



Gambar 19.9 Endpoint Tampil Sasaran Mutu yang Gagal

### 3. Skenario tampildata bukti capaian

#### 1. Tampil data bukti capaian yang berhasil

Diketahui pengguna ingin melihat bukti capaian untuk salah satu butir sasaran mutu (Bidang Penelitian Dosen dengan Jurnal Internasional) ke dalam sistem BPM UII, maka akan mengirimkan data *request* ke *endpoint* “GET /api/v1/formulir/BidangPenelitianDosen/jurnalInternasional” dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* jika berhasil maka pengguna mendapatkan *status response code* “200”. Pada Gambar 19.10 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/formulir/BidangPenelitianDosen/jurnalInternasional” dengan berisikan:

1. *Success* yang berisi nilai *true*.
2. *Result* berisikan koleksi data dari bukti capaian sesuai lingkup aksesnya.

The screenshot shows the 'Headers' and 'Body' sections of a web browser's developer tools. The 'Headers' section is expanded to show three headers: 'Authorization' (Bearer eyJ0eXAI0jKV1QILCjhbGciOiJIUzI1NiJ9.eyJpc3MiOiJsZW11...), 'Access-uuid' (1af5756e-f56d-11e9-b95a-0cc47afc359e), and 'Cakupan-uuid' (3f77ddb9-f240-11e9-b95a-0cc47afc359e). The 'Body' section shows a JSON response with the following structure:

```

1  {
2    "success": true,
3    "result": {
4      "formulir": "Luaran Penelitian Dosen",
5      "data": [
6        {
7          "nidn": "115231",
8          "nama_dosen": "Rahadian",
9          "judul_artikel": "Rock n Roll",
10         "bulan": 7,
11         "tahun": "2019",
12         "kategori": "Paten",
13         "nama_media": "Medium",
14         "indeks": "Scopus",
15         "taraf": "Internasional",
16         "nama_file_excel": "",
17         "bukti_pelaksanaan": null,
18         "nidn": "4b540082-f481-11e9-b05a-0cc47afc359e"

```

Gambar 19.10 *Endpoint* Tampil Data Bukti Capaian yang Berhasil

### 1. Tampil data sasaran mutu yang gagal

Jika pengguna tidak memiliki data sasaran mutu maka pengguna mendapatkan status *response code* "404". Pada Gambar 19.11 merupakan struktur konten *response* dari *endpoint* "GET /api/v1/formulir/BidangPenelitianDosen/jurnalInternasional" dengan berisikan:

2. *Success* yang berisi nilai *false*.
3. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.





KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> nidn	15523091			
<input checked="" type="checkbox"/> nama_dosen	Hakaman			
<input checked="" type="checkbox"/> judul_artikel	ALPRO 1			
<input checked="" type="checkbox"/> bulan	11			
<input checked="" type="checkbox"/> tahun	2019			
<input checked="" type="checkbox"/> kategori	Paten			
<input checked="" type="checkbox"/> nama_media	Medium			
<input checked="" type="checkbox"/> indeks	Scopus			
<input checked="" type="checkbox"/> taraf	Internasional			
<input checked="" type="checkbox"/> penulis_utama	1			
Key	Value	Description		

Body Cookies Headers (12) Test Results Status: 200 OK Time: 23.89s Size: 516 B Save Response ▾

Pretty Raw Preview Visualize BETA JSON ▾

```

1
2  "success": true,
3  "message": "Success to save data"
4

```

Gambar 19.12 *Endpoint* Tambah Bukti Capaian yang Berhasil

#### 7. Tambah bukti capaian yang gagal

Jika pengguna tidak lengkap memasukkan data maka pengguna mendapatkan status *response code* “422”. Pada Gambar 19.13 merupakan struktur konten *response* dari *endpoint* “POST /api/v1/formulir/BidangPenelitianDosen/jurnalInternasional” dengan berisikan:

1. *Success* yang berisi nilai *false*.
2. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> nidn	15523091			
<input checked="" type="checkbox"/> nama_dosen	Hakaman			
<input checked="" type="checkbox"/> judul_artikel	ALPRO 1			
<input checked="" type="checkbox"/> bulan	11			
<input checked="" type="checkbox"/> tahun	2019			
<input checked="" type="checkbox"/> kategori	Paten			
<input checked="" type="checkbox"/> nama_media	Medium			
<input checked="" type="checkbox"/> indeks	Scopus			
<input checked="" type="checkbox"/> taraf	Internasional			
<input type="checkbox"/> penulis_utama	1			
Key	Value	Description		

Body Cookies Headers (12) Test Results Status: 422 Unprocessable Entity Time: 7.11s Size: 549 B Save Response

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "success": false,
3   "error": "The penulis utama field is required."
4 }

```

Gambar 19.13 *Endpoint* Tambah Bukti Capaian yang Gagal

### 3. Skenario hapus data bukti capaian yang berhasil

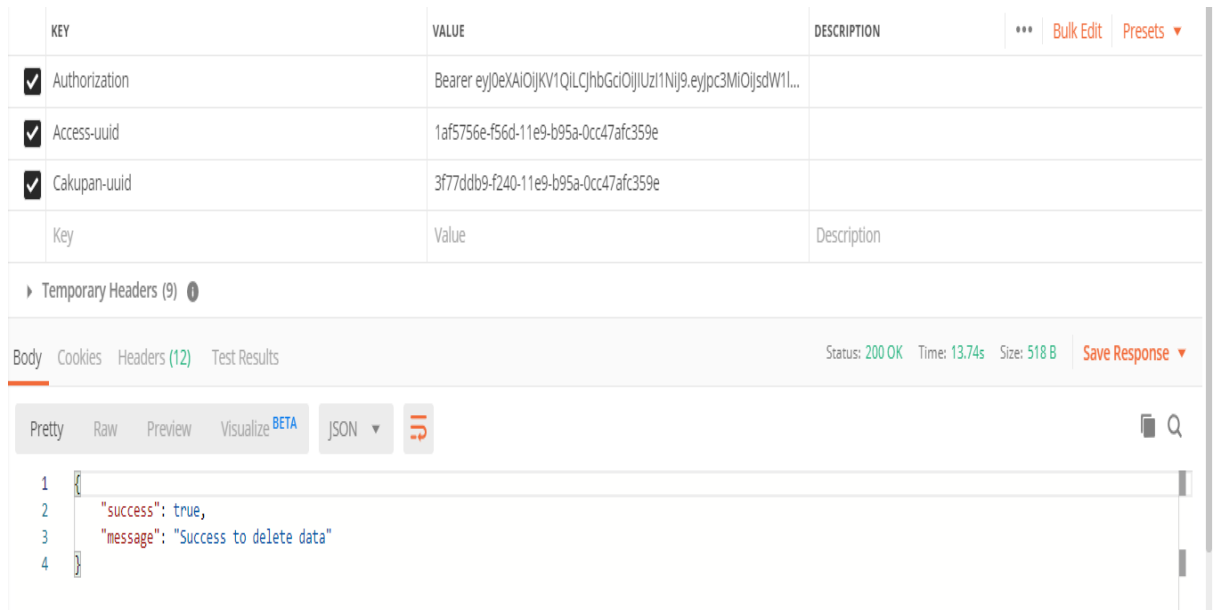
#### 1. Hapus data bukti capaian yang berhasil

Diketahui auditee ingin menghapus data bukti capaian dari salah satu data bukti capaian, maka akan mengirimkan data *request* ke *endpoint* “DELETE /api/v1/formulir/BidangPenelitianDosen/jurnalInternasional/uuidData” dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* ke *endpoint* “DELETE formulir/BidangPenelitianDosen/jurnalInternasional/uuidData” jika berhasil maka auditee mendapatkan status *response code* “200”. Pada Gambar 19.14 merupakan struktur konten *response* dari *endpoint* “DELETE formulir/BidangPenelitianDosen/jurnalInternasional/uuidData” dengan berisikan

1. *Success* yang berisi nilai *true*.
2. *Message* berisikan pesan dari informasi yang dikirimkan server.

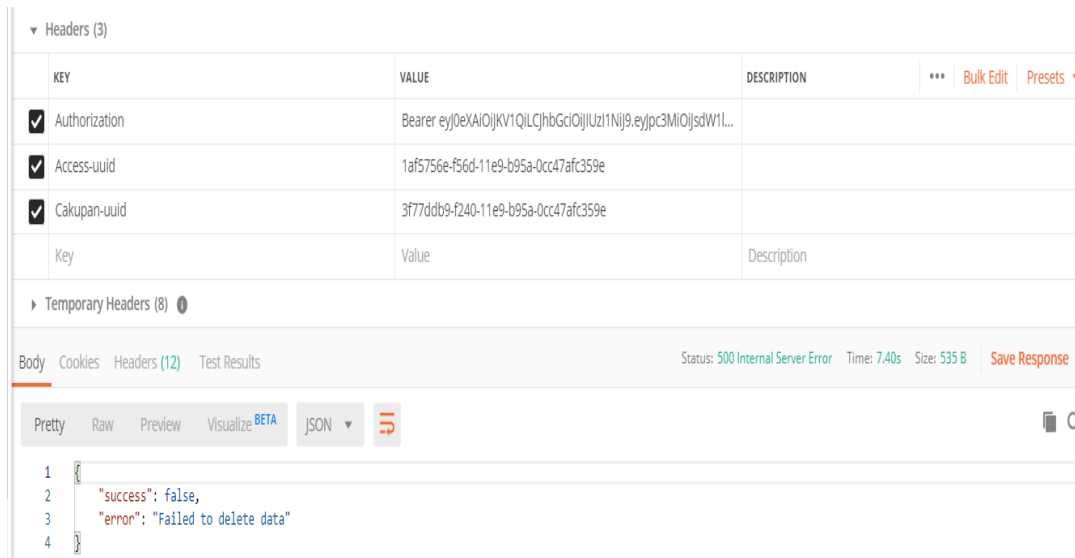


Gambar 19.14 *Endpoint* Hapus Data Bukti Capaian yang Berhasil

### 3. Hapus data bukti capaian yang gagal

Jika pengguna melakukan kesalahan dalam mengirimkan *request* maka pengguna mendapatkan status *response code* “500”. Pada Gambar 19.13 merupakan struktur konten *response* dari *endpoint* “DELETE /api/v1/formulir/BidangPenelitianDosen/jurnalInternasional/uuidData” dengan berisikan

4. *Success* yang berisi nilai *false*.
5. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.



Gambar 19.15 *Endpoint* Hapus Data Bukti Capaian yang Gagal

## 6. Skenario tampil data rekomendasi tindak lanjut

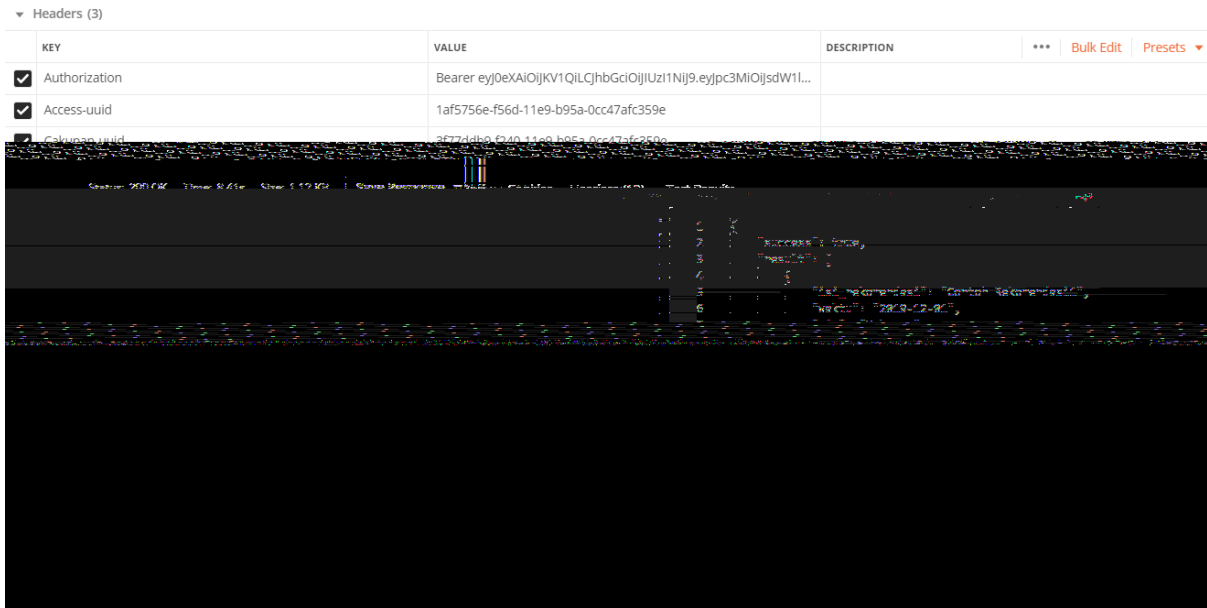
### 1. Tampil data rekomendasi tindak lanjut yang berhasil

Diketahui pengguna ingin menampilkan data rekomendasi tindak lanjut untuk salah satu butir sasaran mutu ke dalam sistem BPM UII, maka akan mengirimkan *request* dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* ke *endpoint* “GET /api/v1/rekomendasi/getData/uuidButir” jika berhasil maka pengguna mendapatkan *status response code* “200”. Pada Gambar 19.16 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/rekomendasi/getData/uuidButir” dengan berisikan

1. *Success* yang berisi nilai *true*.
2. *Result* berisikan koleksi data rekomendasi tindak lanjut sesuai dengan lingkup aksesnya.

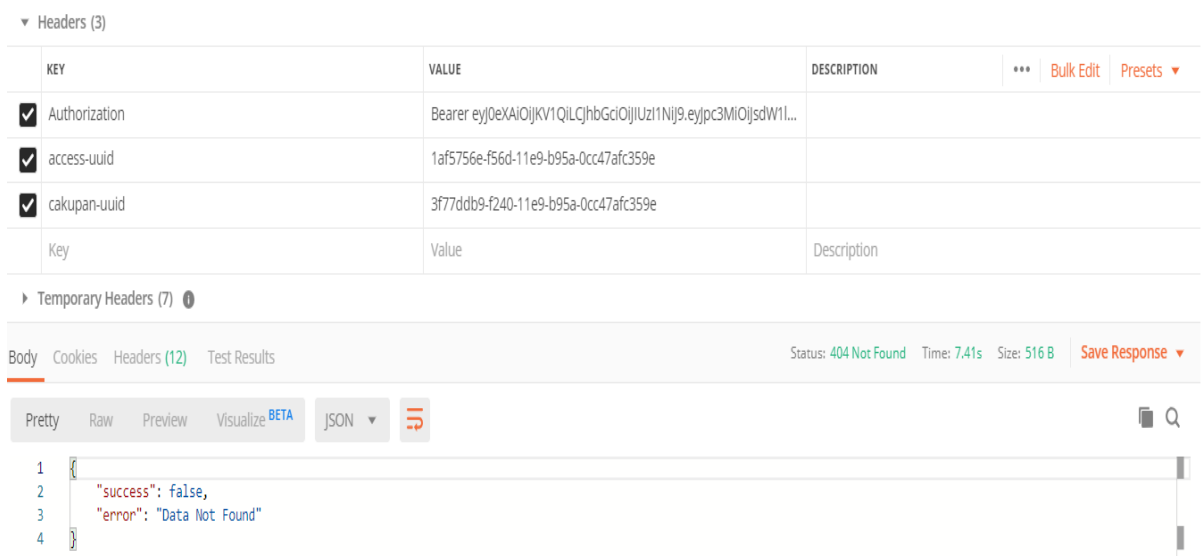


Gambar 19.16 Tampil Data Rekomendasi Tindak Lanjut yang berhasil

### 3. Tampil data rekomendasi tindak lanjut yang gagal

Jika pengguna dalam mengirimkan *request* tetapi tidak ada data rekomendasi tindak lanjut maka pengguna mendapatkan status *response code* “422”. Pada Gambar 19.17 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/rekomendasi/getData/uuidButir” dengan berisikan:

4. *Success* yang berisi nilai *false*.
5. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.



Gambar 19.17 Tampil Data Rekomendasi Tindak Lanjut yang gagal

## 6. Skenario tambah data rekomendasi tindak lanjut

### 1. Tambah data rekomendasi tindak lanjut yang berhasil

Diketahui auditee ingin menambahkan data rekomendasi tindak lanjut untuk salah satu butir sasaran mutu ke dalam sistem BPM UII, maka akan mengirimkan data *request* dengan isi\_rekomendasi, waktu, dan pic dan dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* ke *endpoint* “POST /api/v1/rekomendasi/setRekomendasi/uuidButir” jika berhasil maka auditee mendapatkan *statusresponse code* “200”. Pada Gambar 19.18 merupakan struktur konten *response* dari *endpoint* “POST /api/v1/rekomendasi/setRekomendasi/uuidButir” dengan berisikan

5. *Success* yang berisi nilai *true*.
6. *Message* berisikan pesan dari informasi yang dikirimkan server

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> isi_rekomendasi	Contoh Rekomendasi1			
<input checked="" type="checkbox"/> waktu	2019-12-01			



```

{
  "success": true,
  "message": "Rekomendasi telah ditambahkan."
}

```

Gambar 19.18 *Endpoint* Tambah Rekomendasi Tindak Lanjut yang berhasil

#### 7. Tambah rekomendasi tindak lanjut yang gagal

Jika pengguna tidak lengkap memasukkan data maka pengguna mendapatkan status *response code* “422”. Pada Gambar 19.19 merupakan struktur konten *response* dari *endpoint* “POST /api/v1/rekomendasi/ setRekomendasi/uuidButir” dengan berisikan:

8. *Success* yang berisi nilai *false*.
9. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.

KEY	VALUE	DESCRIPTION	...	Bulk Edit



```

{
  "success": false,
  "error": "Rekomendasi yang ditambahkan tidak valid."
}

```

Gambar 19.19 *Endpoint* Tambah Rekomendasi Tindak Lanjut Gagal



## 10. Skenario hapus rekomendasi tindak lanjut

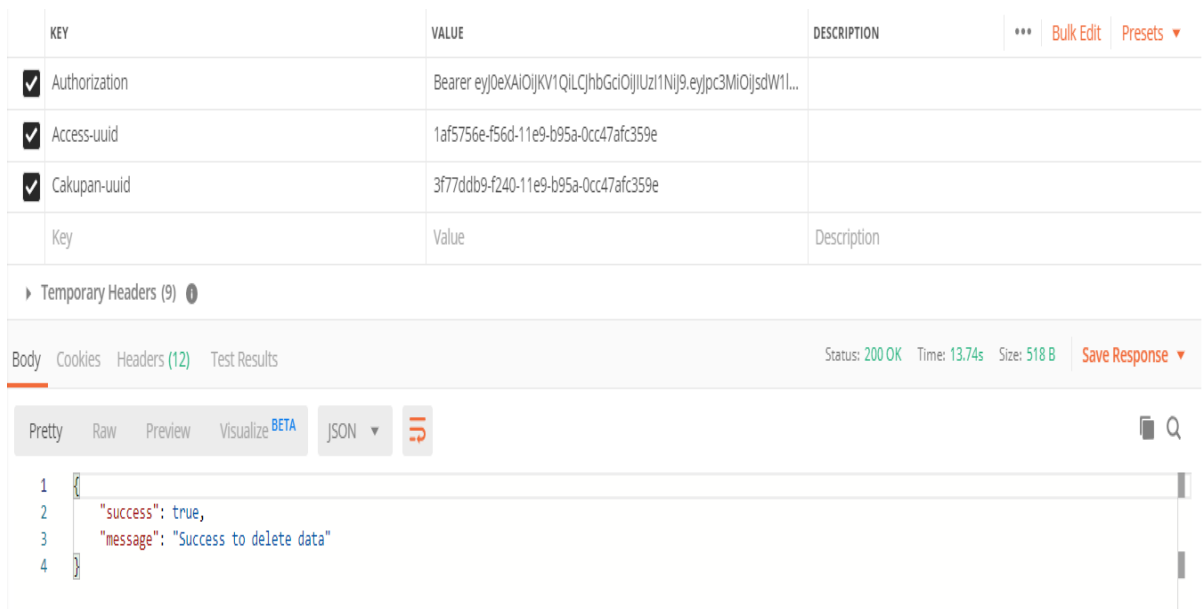
### 1. Hapus data rekomendasi tindak lanjut yang berhasil

Diketahui auditee ingin menghapus data rekomendasi tindak lanjut dari salah satu data rekomendasi tindak lanjut, maka akan mengirimkan data *request* ke *endpoint* “DELETE /api/v1/rekomendasi/deleteData/uuidData” dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* ke *endpoint* “DELETE formulir/BidangPenelitianDosen/{uuidData}” jika berhasil maka auditee mendapatkan *statusresponse code* “200”. Pada Gambar 19.20 merupakan struktur konten *response* dari *endpoint* “DELETE /api/v1/rekomendasi/deleteData/uuidData” dengan berisikan

5. *Success* yang berisi nilai *true*.
6. *Message* berisikan pesan dari informasi yang dikirimkan server.

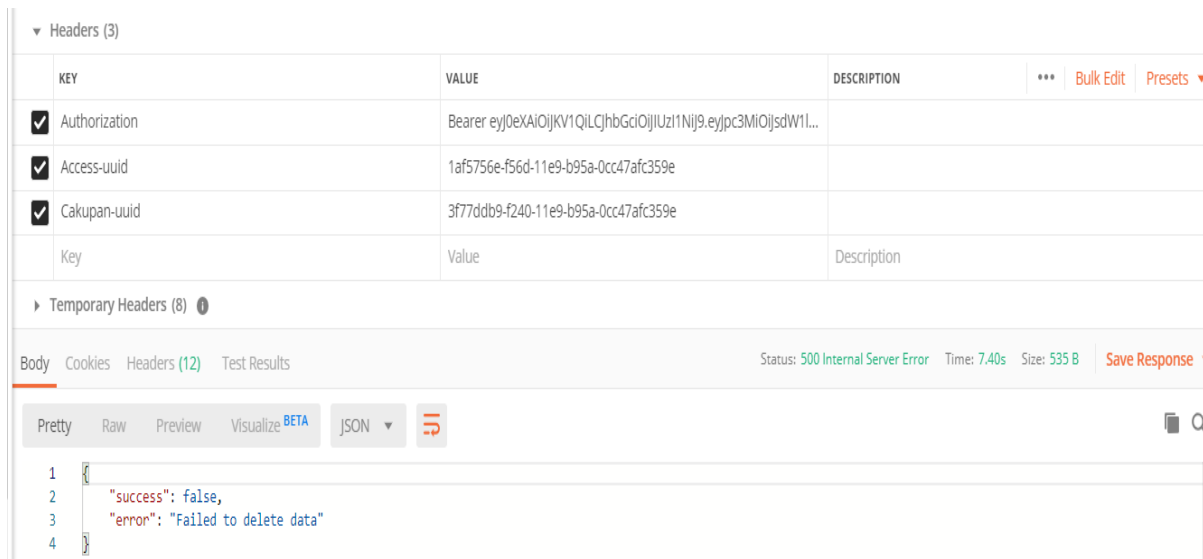


Gambar 19.20 *Endpoint* Hapus Rekomendasi Tindak Lanjut yang Berhasil

### 7. Hapus data rekomendasi tindak lanjut yang gagal

Jika pengguna melakukan kesalahan dalam mengirimkan *request* maka pengguna mendapatkan status *response code* “500”. Pada Gambar 19.21 merupakan struktur konten *response* dari *endpoint* “DELETE /api/v1/rekomendasi/deleteData/uuidData” dengan berisikan

8. *Success* yang berisi nilai *false*.
9. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.



Gambar 19.21 *Endpoint* Hapus Rekomendasi Tindak Lanjut yang Gagal

## 10. Skenario keranjang validasi

### 1. Tampil data keranjang validasi yang berhasil

Diketahui auditee ingin menampilkan data keranjang validasi, maka akan mengirimkan data *request* ke *endpoint* “GET /api/v1/validitas” dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* ke *endpoint* “GET/api/v1/validitas” maka jika berhasil maka auditee mendapatkan status *response code* “200”. Pada Gambar 19.22 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/validitas” dengan berisikan

5. *Success* yang berisi nilai *true*.

6. *Result* berisikan koleksi data keranjang validasi sesuai dengan lingkup aksesnya.

The screenshot shows the 'Headers' section of a browser's developer tools. Below the headers, the 'Body' section is expanded to show the response in JSON format. The response is a 200 OK status with a response time of 8.60s and a size of 5.37 KB. The JSON body is as follows:

```

1  {
2    "success": true,
3    "result": [
4      {
5        "id": 5,
6        "capaian": "0",
7        "nilai_akhir": "0",
8        "kategori_capaian": "menyimpang",
9        "target": "0.5%",
10       "status": "check-validasi-auditee",
11       "standar": "",
12       "pegawais_id": 385,
13       "butirs_id": 5,
14       "organisasias_id": 90,
15       "perisodes_id": 1,
16       "uuid": "5f890a85-ec1a-11e9-8758-0cc47afc359e",
17       "created_at": "2019-10-11 11:29:22",
18       "updated_at": "2019-11-27 03:32:36"
19     }
20   ]
21 }

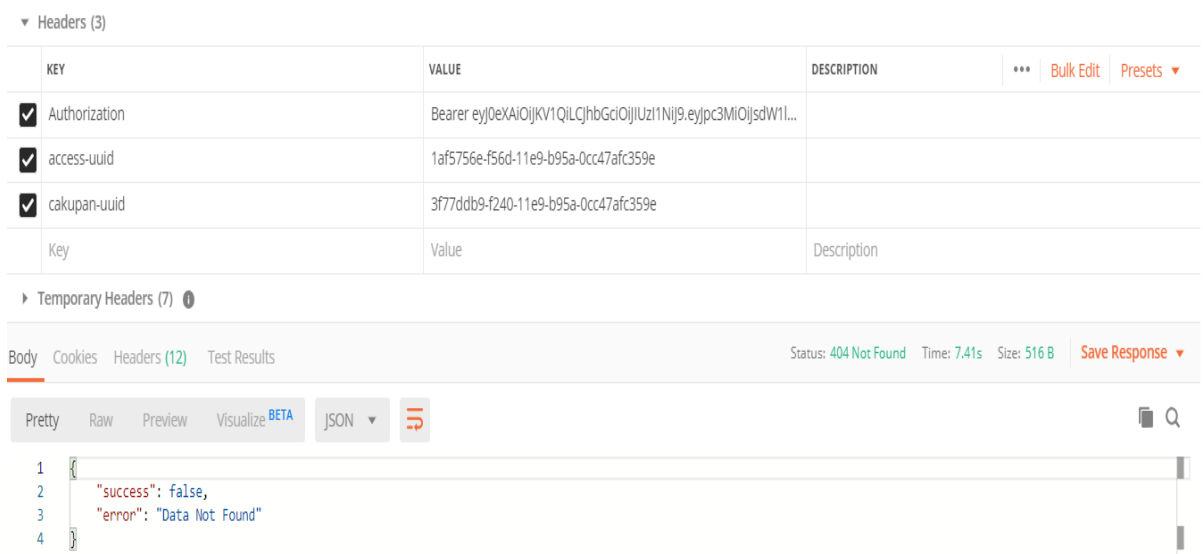
```

Gambar 19.22 *Endpoint* Tampil Keranjang Validasi yang Berhasil

### 1. Tampil data keranjang validasi yang gagal

Jika pengguna mengirimkan *request* tetapi tidak ada data di keranjang validasi maka pengguna mendapatkan status *response code* “422”. Pada Gambar 19.23 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/validitas” dengan berisikan:

2. *Success* yang berisi nilai *false*.
3. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.



Gambar 19.23 *Endpoint* Tampil Keranjang Validasi yang Gagal

#### 4. Skenario memvalidasi keranjang validasi

##### 1. Memvalidasi data keranjang validasi yang berhasil

Diketahui auditee ingin memvalidasi data dari keranjang validasi, maka akan mengirimkan data *request* ke *endpoint* “PUT /api/v1/validitas” dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* ke *endpoint* “PUT /api/v1/validitas” jika berhasil maka auditee mendapatkan *statusresponse code* “200”. Pada Gambar 19.24 merupakan struktur konten *response* dari *endpoint* “PUT /api/v1/validitas” dengan berisikan

5. *Success* yang berisi nilai *true*.
6. *Message* berisikan pesan yang dikirimkan dari server.

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets ▾
<input checked="" type="checkbox"/> Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJsZW11...				
<input checked="" type="checkbox"/> Access-uuid	1af5756e-f56d-11e9-b95a-0cc47afc359e				
<input checked="" type="checkbox"/> Cakupan-uuid	3f77ddb9-f240-11e9-b95a-0cc47afc359e				

Gambar 19.24 *Endpoint* Memvalidasi Keranjang Validasi yang Berhasil

#### 1. Memvalidasi data keranjang validasi yang gagal

Ketika pengguna tidak memiliki data keranjang validasi maka pengguna mendapatkan status *response code* “404”. Pada Gambar 19.25 merupakan struktur konten *response* dari *endpoint* “PUT /api/v1/validitas” dengan berisikan:

2. *Success* yang berisi nilai *false*.
3. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets ▾
[The content of this table is obscured by a black redaction box.]					

Gambar 19.25 *Endpoint* Memvalidasi Keranjang Validasi yang Gagal

#### 4. Skenario tampil laporan temuan AMI

##### 1. Tampil data laporan temuan AMI yang berhasil

Diketahui auditee ingin menampilkan data laporan temuan AMI, maka akan mengirimkan data *request* ke *endpoint* “GET /api/v1/sasaranmutu/laporanAMI” dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* ke *endpoint* “GET /api/v1/sasaranmutu/laporanAMI” jika berhasil maka auditee mendapatkan status *response code* “200”. Pada Gambar 19.26 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/sasaranmutu/laporanAMI” dengan berisikan

5. *Success* yang berisi nilai *true*.
6. *Result* berisikan koleksi data laporan temuan AMI sesuai dengan lingkup aksesnya.

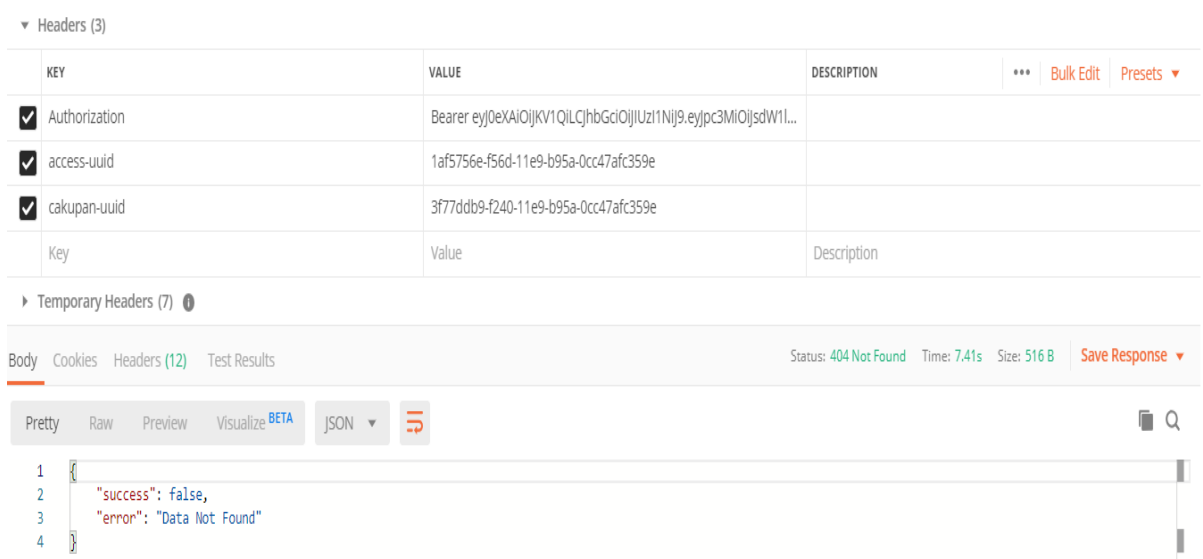
KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJsZW11L...				
<input checked="" type="checkbox"/> Access-uuid	1af5756e-f56d-11e9-b95a-0cc47afc359e				
<input checked="" type="checkbox"/> Cakupan-uuid	3f77dd9-f240-11e9-b95a-0cc47afc359e				

Gambar 19.26 *Endpoint* Tampil Laporan AMI yang Berhasil

#### 1. Tampil data laporan temuan AMI yang gagal

Jika pengguna mengirimkan *request* tetapi tidak ada data laporan temuan AMI maka pengguna mendapatkan status *response code* “404”. Pada Gambar 19.27 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/validitas” dengan berisikan:

2. *Success* yang berisi nilai *false*.
3. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.



Gambar 19.27 *Endpoint* Tampil Laporan AMI yang Gagal

#### 4. Skenario tambah data laporan temuan AMI

##### 1. Tambah data laporan temuan AMI yang berhasil

Diketahui auditee ingin menambahkan data laporan temuan AMI ke dalam sistem BPM UII, maka akan mengirimkan data *request* dengan array yang bernama data dengan indeks bernama analisis\_penyebab, perbaikan, tindakan\_korektif dan dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* ke *endpoint* “POST /api/v1/sasaranmutu/isianTemuanAMI” jika berhasil maka auditee mendapatkan status *response code* “200”. Pada Gambar 19.28 merupakan struktur konten *response* dari *endpoint* “POST /api/v1/sasaranmutu/isianTemuanAMI” dengan berisikan

5. *Success* yang berisi nilai *true*.
6. *Message* berisikan pesan dari informasi yang dikirimkan server

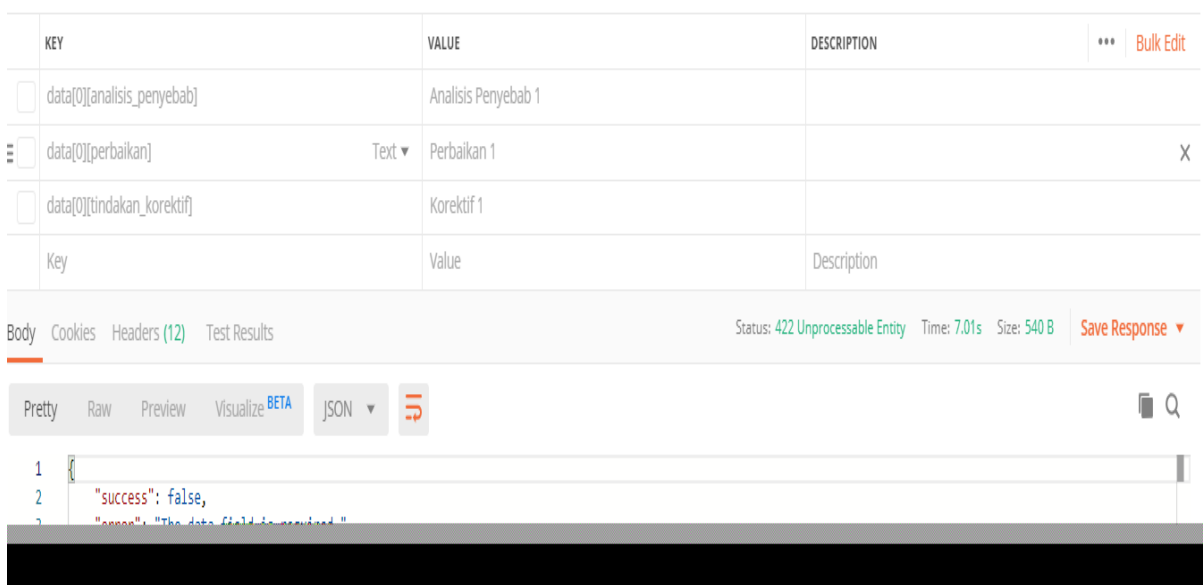
KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> data[0][analisis_penyebab]	Analisis Penyebab 1	
<input checked="" type="checkbox"/> data[0][perbaikan]	Perbaikan 1	



Gambar 19.28 *Endpoint* Tambah Isian Laporan Temuan AMI yang berhasil

1. Tambah data isian laporan temuan AMI yang gagal  
 Jika pengguna tidak lengkap memasukkan data maka pengguna mendapatkan status *response code* “422”. Pada Gambar 19.29 merupakan struktur konten *response* dari *endpoint* “POST /api/v1/rekomendasi/ setRekomendasi/uuidButir” dengan berisikan:
  2. *Success* yang berisi nilai *false*.
  3. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.





Gambar 19.29 Endpoint Tambah Isian Laporan Temuan AMI yang Gagal

#### 4. Skenario tampil data grafik

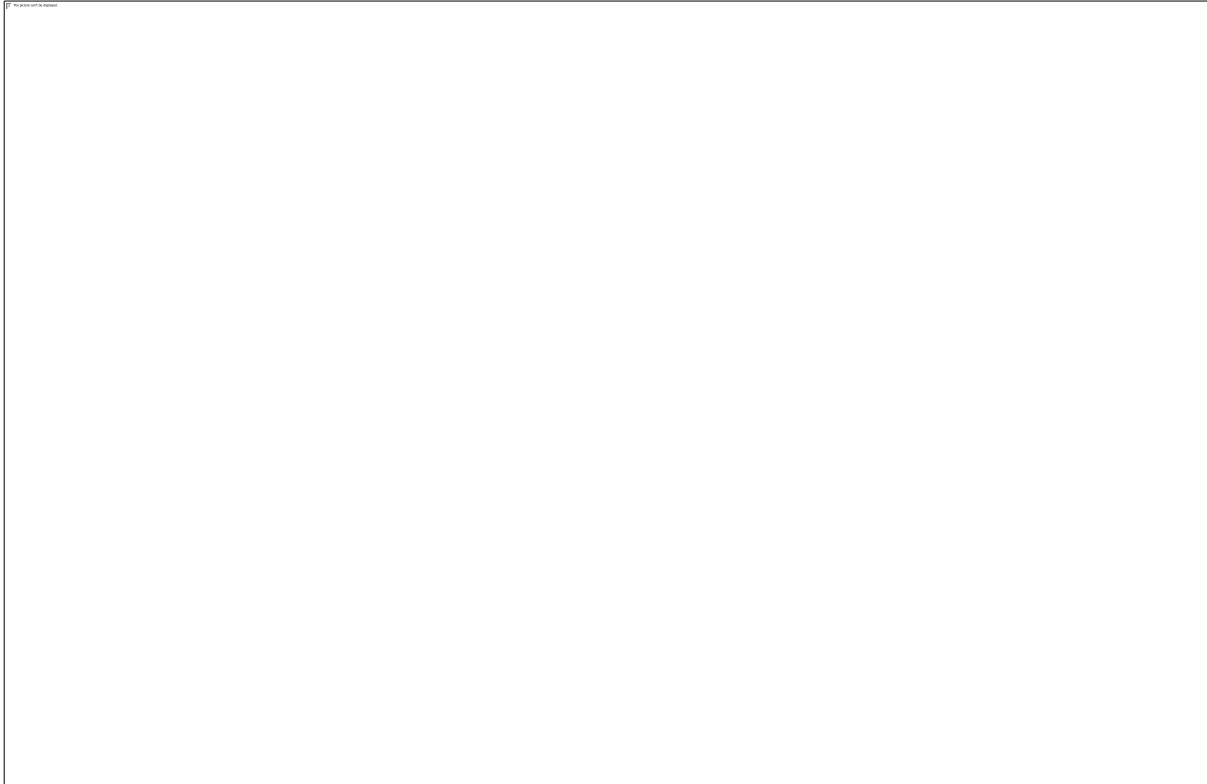
##### 1. Tampil data grafik yang berhasil

Diketahui auditee ingin menampilkan data grafik, maka akan mengirimkan data *request* ke *endpoint* “GET /api/v1/sasaranmutu/laporanAMI” dengan tiga *header* yaitu

2. *Header authorization* yang didapatkan ketika *log in*.
3. *Header access-uuid* yaitu *uuid* yang didapatkan ketika pemilihan peran pengguna.
4. *Header cakupan-uuid* yaitu *uuid\_cakupan* yang didapatkan ketika pemilihan peran pengguna.

Setelah pengguna mengirimkan *request* ke *endpoint* “GET /api/v1/sasaranmutu/grafikStandar” jika berhasil maka auditee mendapatkan *statusresponse code* “200”. Pada Gambar 19.30 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/sasaranmutu/grafikStandar” dengan berisikan

5. *Success* yang berisi nilai *true*.
6. *Result* berisikan koleksi data grafik sesuai dengan lingkup aksesnya.



Gambar 19.30 *Endpoint* Tampil Data Grafik yang Berhasil

1. Tampil data laporan temuan AMI yang gagal

Jika pengguna mengirimkan *request* tetapi tidak ada data laporan temuan AMI maka pengguna mendapatkan status *response code* “404”. Pada Gambar 19.31 merupakan struktur konten *response* dari *endpoint* “GET /api/v1/ sasaranmutu/grafikStandar” dengan berisikan:

2. *Success* yang berisi nilai *false*.
3. *Error* berisikan pesan dari kesalahan yang dikirimkan dari server.

The screenshot shows the 'Headers' tab of a browser's developer tools. It lists three headers: 'Authorization' (Bearer eyJ0eXAI0jKV1QILCjhbGciOiJIUzI1NiJ9.eyJpc3MiOiJsdW11...), 'access-uuid' (1af5756e-f56d-11e9-b95a-0cc47afc359e), and 'cakupan-uuid' (3f77ddb9-f240-11e9-b95a-0cc47afc359e). Below the headers, the 'Temporary Headers' tab is active, showing a status of '404 Not Found', a time of '7.41s', and a size of '516 B'. The 'Body' tab is selected, displaying a JSON response: 

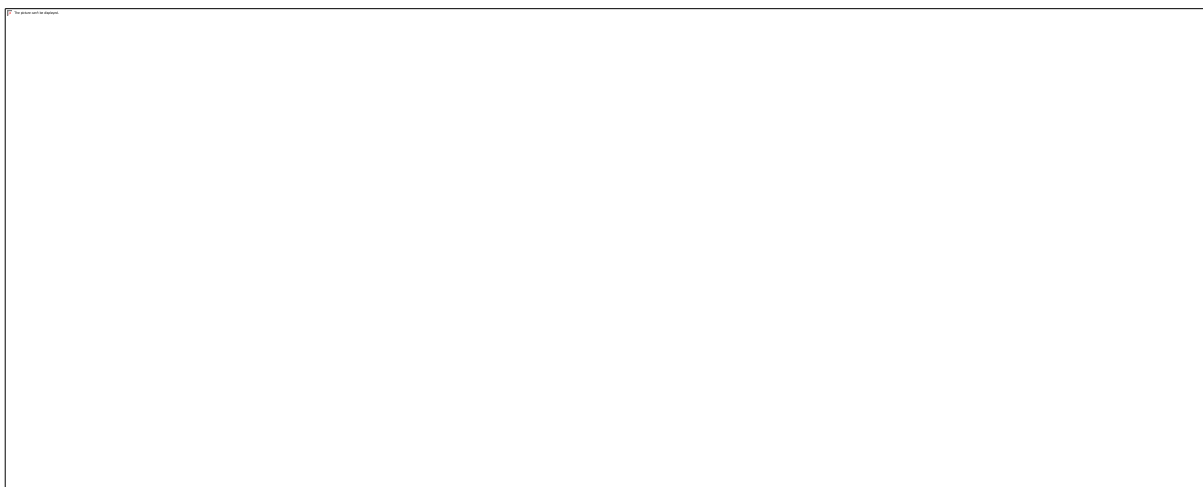
```
{
  "success": false,
  "error": "Data Not Found"
}
```

Gambar 19.31 *Endpoint* Tampil Data Grafik yang Gagal

#### 4. Skenario otorisasi

Diketahui auditee ingin menambahkan komentar untuk salah satu sasaran mutu ke dalam sistem BPM UII, maka akan mengirimkan data *request* komentar ke *endpoint* “POST /sasaranmutu/komentar/uuidButir” maka klien mendapatkan *statusresponsecode* “401”. Status *responsestatuscode* “401” dikarenakan komentar sasaran mutu dari butir hanya diperbolehkan pada *role* pengguna auditor. Dengan struktur konten yaitu

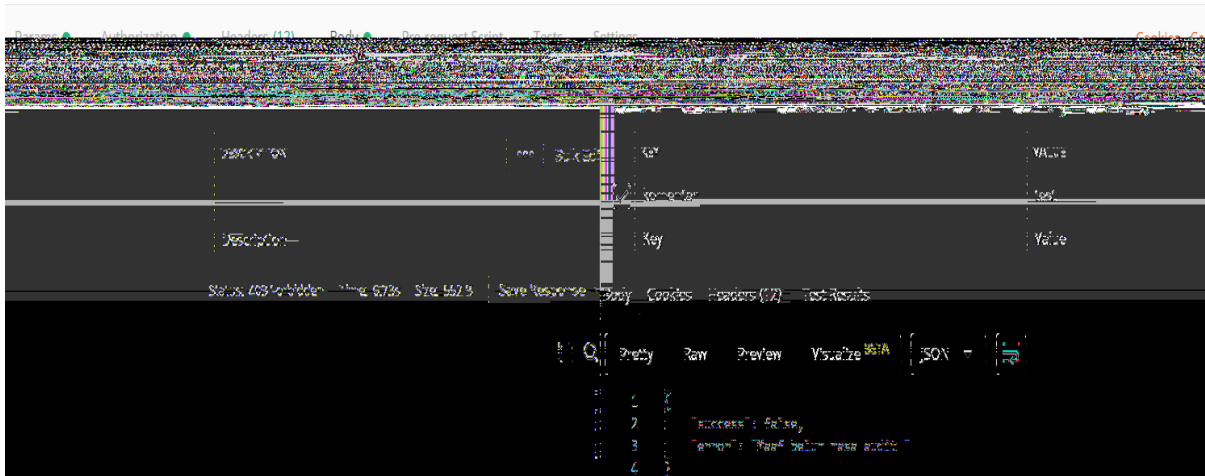
1. *Success* yang berisi nilai *true*.
2. *Error* berisikan pesan dari informasi yang dikirimkan server



Gambar 19.32 *Endpoint* yang terkena *Unauthorized*

### 3. Skenario *Cut Off*

Diketahui klien ingin menambahkan komentar untuk salah satu sasaran mutu ke dalam sistem BPM UII, maka akan mengirimkan data *request* komentar *endpoint* “POST /sasaranmutu/komentar/uuidButir”” maka klien mendapatkan *statusresponse code* “200”. *Status response code* “200” dikarenakan belum memasuki masa audit, Dengan struktur konten yaitu



Gambar 19.33 Skenario *Cut Off*

#### 1. *Performance Testing*

*Performancetesting* merupakan pengujian yang dilakukan untuk mengukur kriteria performa pada sebuah sistem aplikasi (Matam,2017). Untuk mengukur performa pada sebuah sistem dilakukan menggunakan aplikasi Apache Jmeter. Aplikasi apache Jmeter merupakan salah satu aplikasi pengujian *performance testing* yang digunakan untuk *load testing* aplikasi web, aplikasi FTP, dan *database server test*. *Performance testing* akan dibagi menjadi 3 bagian yaitu *performance testing* untuk auditee, *performance testing* untuk auditor, dan *performance testing* untuk admin.

#### **Auditor**

Pada bagian ini akan dilakukan *performance testing* dengan keseluruhan *endpoint* yang dapat diakses oleh auditor. Setelah dilakukan pengujian mendapatkan hasil bahwa *endpoint* dengan waktu terlalu lama dalam memberikan *response* yaitu PUT /api/v1/sasaranmutu/selesai/uuidButir. Sedangkan untuk rata – rata waktu kecepatan dari *responseendpoint* yang dapat diakses oleh auditor yaitu 904 dan rata – rata ukuran

*response* sebesar 5483.6 *bytes*. Selengkapnya dapat dilihat pada Gambar 19.34 dan Gambar 19.35.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Through...	Received ...	Sent KB...	Avg. Bytes
POST /auth/login	1	1317	1317	1317	0.00	0.00%	45.6/min	1.34	0.73	1809.0
GET /api/v1/role/organisasiRole	1	597	597	597	0.00	0.00%	1.7/sec	1.30	0.81	796.0
GET /rekomendasi/getData/uuidButir	1	705	705	705	0.00	0.00%	1.4/sec	4.33	0.74	3128.0
GET /api/v1/rekomendasi/getData	1	901	901	901	0.00	0.00%	1.1/sec	6.50	0.54	5997.0
DELETE /api/v1/rekomendasi/deleteData/uuidButir	1	600	600	600	0.00	0.00%	1.7/sec	0.87	1.01	532.0
POST /api/v1/sasaranomby/komanda/uuidButir	1	601	601	601	0.00	0.00%	1.7/sec	0.85	1.40	525.0

Gambar 19.34 *Performance Testing Endpoint Auditor*

Sa...	Start Time	Thread Name	Label	Sample ...	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	20:28:38.725	Pengujian Web Serv...	POST /auth/login	1317	✓	1809	987	1316	118
2	20:28:40.045	Pengujian Web Serv...	GET /api/v1/role/organisasiRole	597	✓	796	496	597	0
3	20:28:40.643	Pengujian Web Serv...	GET /rekomendasi/getData/uuidButir	705	✓	3128	533	705	0
4	20:28:41.349	Pengujian Web Serv...	GET /api/v1/rekomendasi/getData	901	✓	5997	496	630	41

Gambar 19.35 *Endpoint Status Auditor*

## Auditee

Pada bagian ini akan dilakukan *performance testing* dengan keseluruhan *endpoint* yang dapat diakses oleh auditee. Setelah dilakukan pengujian mendapatkan hasil bahwa *endpoint* dengan waktu terlalu lama dalam memberikan *response* yaitu POST /api/v1/rekomendasi/setRekomendasi/uuidButir. Sedangkan untuk rata – rata waktu kecepatan dari *response endpoint* yang dapat diakses oleh auditor yaitu 1376 dan rata – rata ukuran *response* sebesar 5516.8 *bytes*. Selengkapnya dapat dilihat pada Gambar 19.36 dan Gambar 19.37.

Label	# S...	Average	Min	Max	Std. Dev.	Error %	Through...	Received ...	Sent KB/...	Avg. Bytes
POST /auth/login	1	839	839	839	0.00	0.00%	1.2/sec	2.11	0.69	1809.0
GET /api/v1/role/organisasiROle	1	1098	1098	1098	0.00	0.00%	54.6/min	0.71	0.44	796.0
POST /api/v1/rekomendasi/setRekomendasi/uuidButir	1	2027	2027	2027	0.00	0.00%	1.8/sec	1.208	0.77	530.0
GET /api/v1/rekomendasi/getData/uuidButir	1	1438	1438	1438	0.00	0.00%	1.8/sec	0.533	0.77	4193.0
GET /api/v1/rekomendasi/getData	1	1897	1897	1897	0.00	0.00%	1.8/sec	0.496	0.77	7062.0
DELETE /api/v1/rekomendasi/deleteData	1	3659	3659	3659	0.00	0.00%	1.7/sec	0.622	0.85	532.0
GET /api/v1/sasaranmutu/komentar/uuidButir	1	1140	1140	1140	0.00	0.00%	1.8/sec	0.534	0.47	17489.0
GET /api/v1/validitas	1	614	614	614	0.00	0.00%	1.8/sec	0.486	0.47	1279.0
PUT /api/v1/validitas	1	702	702	702	0.00	0.00%	1.8/sec	0.569	0.47	509.0
GET /api/v1/butir/listButir/uuidStandar	1	602	602	602	0.00	0.00%	1.8/sec	0.529	0.47	1092.0
GET /api/v1/butir/listButir	1	1126	1126	1126	0.00	0.00%	1.8/sec	0.492	0.47	25394.0

Gambar 19.36 Performance Testing Endpoint Auditee

Sample #	Start Time	Th...	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect T...
1	20:55:54.715	Pen...	POST /auth/login	839	✓	1809	589	839	12
2	20:55:55.555	Pen...	GET /api/v1/role/organisasiROle	1098	✓	796	496	1098	0
3	20:55:56.654	Pen...	POST /api/v1/rekomendasi/setRekomendasi/uuidButir	2027	✓	530	1208	2027	0
4	20:55:58.682	Pen...	GET /api/v1/rekomendasi/getData/uuidButir	1438	✓	4193	533	1438	747
5	20:56:00.121	Pen...	GET /api/v1/rekomendasi/getData	1897	✓	7062	496	1897	0
6	20:56:02.019	Pen...	DELETE /api/v1/rekomendasi/deleteData	3659	✓	532	622	3659	3027
7	20:56:05.679	Pen...	GET /api/v1/sasaranmutu/komentar/uuidButir	1140	✓	17489	534	602	5
8	20:56:06.820	Pen...	GET /api/v1/validitas	614	✓	1279	486	613	0
9	20:56:07.434	Pen...	PUT /api/v1/validitas	702	✓	509	569	702	0
10	20:56:08.137	Pen...	GET /api/v1/butir/listButir/uuidStandar	602	✓	1092	529	602	7
11	20:56:08.740	Pen...	GET /api/v1/butir/listButir	1126	✓	25394	492	591	0

Gambar 19.37 Status Testing Endpoint Auditee

## Admin

Pada bagian ini akan dilakukan performance testing dengan keseluruhan *endpoint* yang dapat diakses oleh admin. Setelah dilakukan pengujian mendapatkan hasil bahwa *endpoint* dengan waktu terlalu lama dalam memberikan *response* yaitu POST /api/v1/rekomendasi/setRekomendasi/uuidButir. Sedangkan untuk rata – rata waktu kecepatan dari *response endpoint* yang dapat diakses oleh auditor yaitu 1376 dan rata – rata ukuran *response* sebesar 5516.8 bytes. Selengkapnya dapat dilihat pada Gambar 19.38 dan Gambar 19.39.

Samp...	Start Time	Thread Name	Label	Samp...	Status	Bytes	Sent B...	Latency	Connect ...
1	00:33:47.656	Pengujian Web ...	POST /auth/login	1296	✓	1809	975	1296	11
2	00:33:48.953	Pengujian Web ...	GET /api/v1/manajemen/pegawai	2580	✓	287307	494	666	0
3	00:33:51.534	Pengujian Web ...	GET /api/v1/manajemen/user	304	✓	12636	491	297	6
4	00:33:51.838	Pengujian Web ...	PUT /api/v1/manajemen/user/uuidRoleUser	575	✓	532	611	575	0
5	00:33:52.413	Pengujian Web ...	DELETE /api/v1/user/uuidRoleUser	573	✓	533	614	573	0
6	00:33:52.986	Pengujian Web ...	GET /api/v1/manajemen/organisasi	596	✓	36275	497	317	0
7	00:33:53.583	Pengujian Web ...	GET /api/v1/manajemen/level	635	✓	879	492	635	10
8	00:33:54.219	Pengujian Web ...	POST /api/v1/manajemen/addUserRole	557	✓	532	1246	557	0
9	00:33:54.777	Pengujian Web ...	GET /api/v1/manajemen/listPeriode	308	✓	521	498	308	0

Gambar 19.38 Performance Testing Endpoint Admin

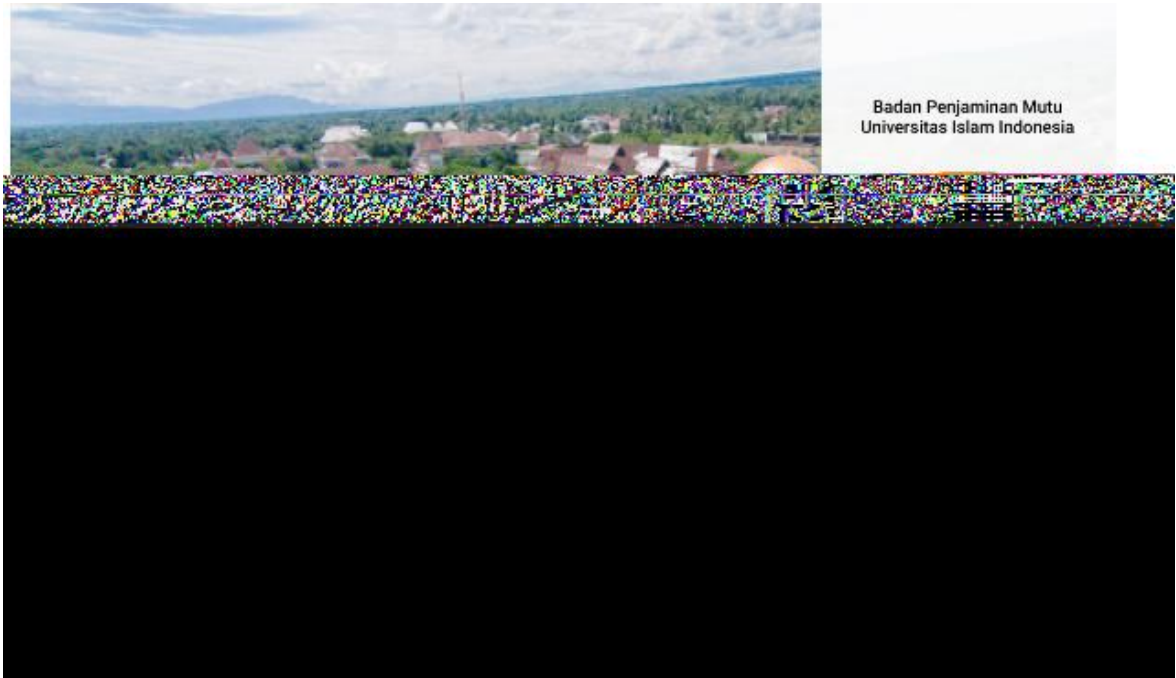
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
POST /auth/login	1	1296	1296	1296	0.00	0.00%	46.3/min	1.36	0.73	1809.0
GET /api/v1/m...	1	2580	2580	2580	0.00	0.00%	23.3/min	108.75	0.19	287307.0
GET /api/v1/m...	1	304	304	304	0.00	0.00%	3.3/sec	40.59	1.58	12636.0
PUT /api/v1/m...	1	575	575	575	0.00	0.00%	1.7/sec	0.90	1.04	532.0
DELETE /api/v...	1	573	573	573	0.00	0.00%	1.7/sec	0.91	1.05	533.0
GET /api/v1/m...	1	596	596	596	0.00	0.00%	1.7/sec	59.44	0.81	36275.0
GET /api/v1/m...	1	635	635	635	0.00	0.00%	1.6/sec	1.35	0.76	879.0
POST /api/v1/...	1	557	557	557	0.00	0.00%	1.8/sec	0.93	2.18	532.0
GET /api/v1/m...	1	308	308	308	0.00	0.00%	3.2/sec	1.65	1.58	521.0
POST /api/v1/...	1	555	555	555	0.00	0.00%	1.8/sec	0.94	0.88	532.0
POST /api/v1/...	1	596	596	596	0.00	0.00%	1.7/sec	0.87	1.10	528.0
GET /api/v1/m...	1	559	559	559	0.00	0.00%	1.8/sec	9.86	0.88	5644.0
PUT /api/v1/m...	1	574	574	574	0.00	0.00%	1.7/sec	0.92	1.35	540.0
PUT /api/v1/m...	1	320	320	320	0.00	0.00%	3.1/sec	1.64	2.27	539.0
GET /api/v1/m...	1	334	334	334	0.00	0.00%	3.0/sec	56.34	1.48	19270.0
PUT /api/v1/b...	1	555	555	555	0.00	0.00%	1.8/sec	0.94	1.08	532.0
TOTAL	16	682	304	2580	537.90	0.00%	1.5/sec	32.94	0.92	23038.1

Gambar 19.39 Status Endpoint Admin

## 2. Deployment Phase

Web Services yang telah dibangun maka akan di *publish* dan digunakan oleh aplikasi klien. Berikut tampilan dari aplikasi klien yang telah dibangun.

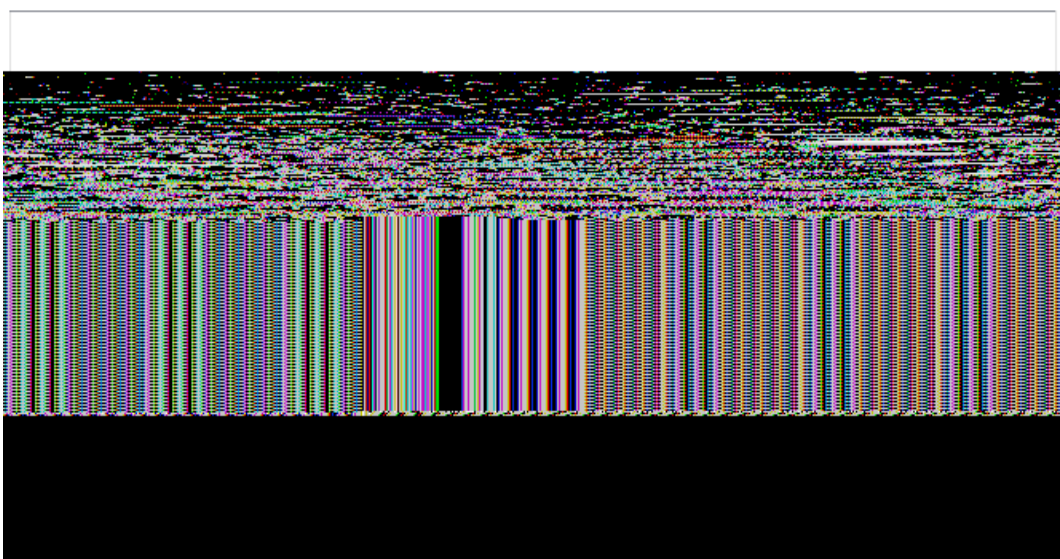
## 1. Login



Gambar 19.40 Tampilan *Login*

Pada Gambar 19.40 merupakan tampilan *login* dengan menggunakan *endpoint* POST `/auth/login`. Pengguna akan diminta untuk melakukan *login* terlebih dahulu dengan memasukkan nip dan password.

## 2. Halaman Pemilihan Hak Akses

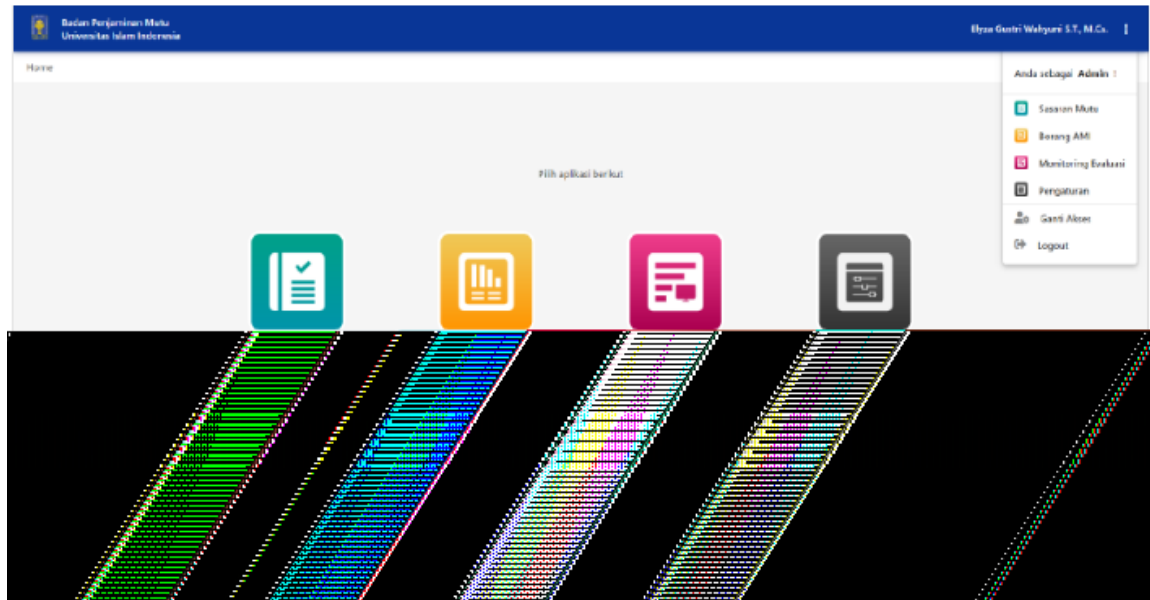


Gambar 19.41 Tampilan Halaman Pemilihan Hak Akses



Pada Gambar 19.41 merupakan tampilan ketika pengguna berhasil masuk ke dalam SIM BPM UII dan akan melakukan pemilihan hak akses yang dimilikinya, hal ini ditujukan karena beberapa aktor memiliki satu atau lebih hak akses.

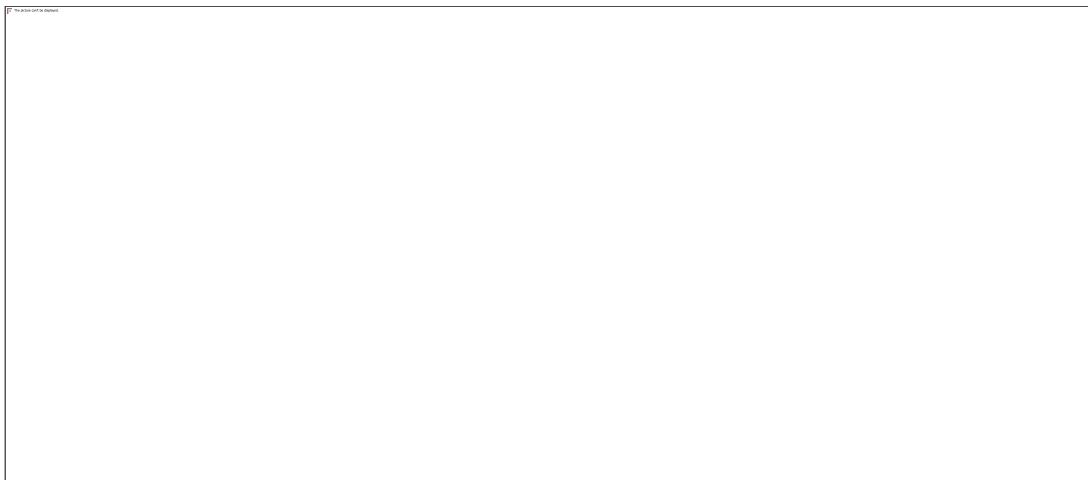
### 3. Halaman Pemilihan Modul



Gambar 19.42 Tampilan Pemilihan Modul

Pada Gambar 19.42 merupakan halaman pemilihan modul yang bertujuan untuk pengguna untuk memilih pilihan modul yang tersedia pada SIMP BPM UII.

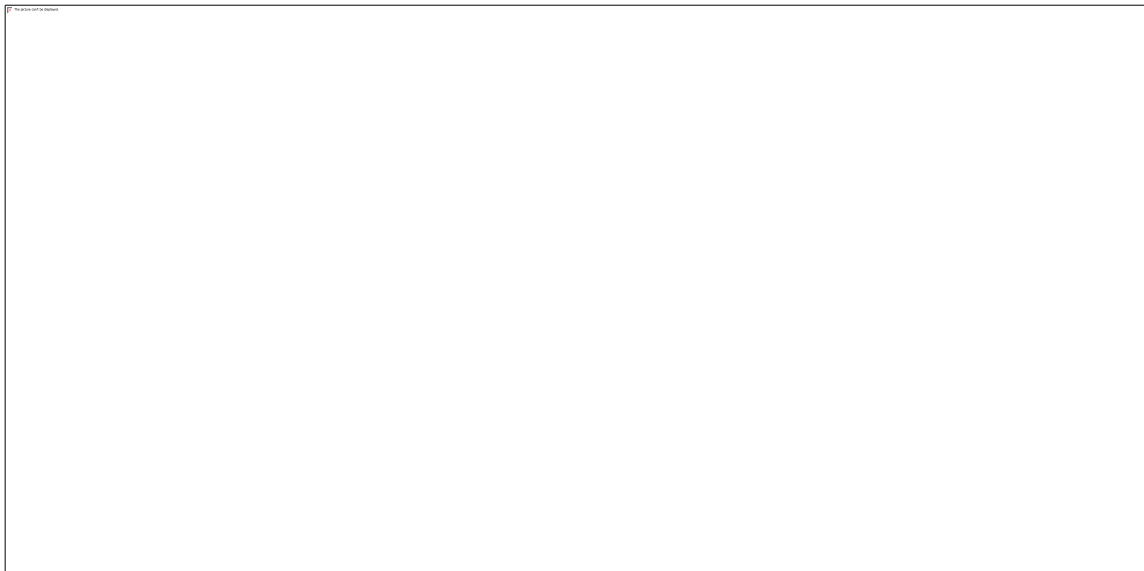
### 4. Halaman Utama Borang AMI



Gambar 19.43 Tampilan Halaman Utama Borang AMI

Pada Gambar 19.43 merupakan halaman utama borang AMI yang memiliki beberapa menu pilihan *sidebar* berupa lingkup audit, validasi, revisi (hanya di auditor), dan informasi. Pada halaman utama borang AMI juga memungkinkan jika pengguna ingin melakukan pemilihan lingkup akses, hal ini memungkinkan auditee/auditor di tingkat fakultas atau universitas dapat melihat data unit lain yang berada dibawah naungan unit tersebut dan terakhir terdapat tampilan diagram untuk melihat representasi hasil capaian unit setiap periode.

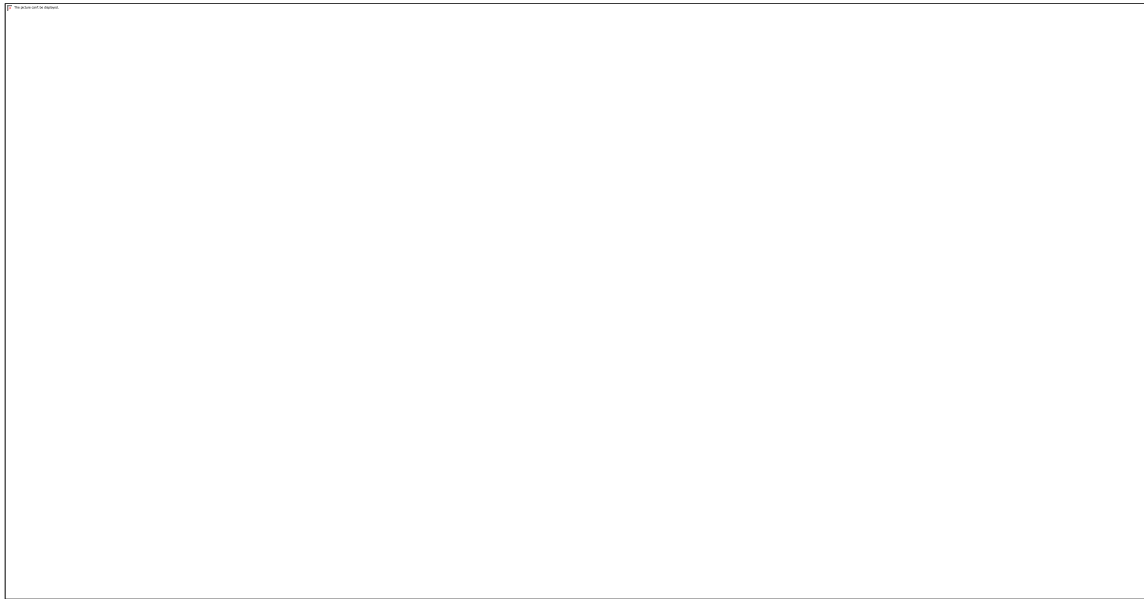
### 5. Halaman Salah Satu Butir Borang AMI



Gambar 19.44 Tampilan Halaman Salah Satu Butir Borang AMI

Menu lingkup audit menyimpan beberapa pilihan yang terdiri dari setiap menu cakupan MERCY OF GOD, jika menu dari salah satu menu MERCY OF GOD dipilih maka akan menampilkan butir yang ada pada cakupannya masing – masing. Tampilan dapat dilihat pada Gambar 19.44.

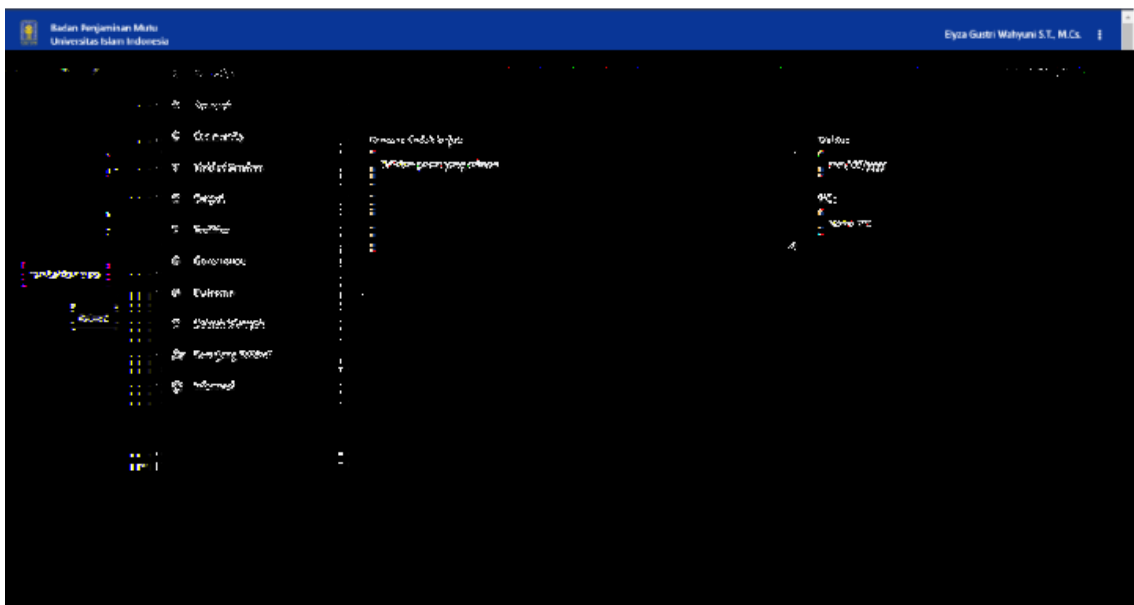
## 6. Halaman Salah Satu Bukti Capaian Borang AMI



Gambar 19.45 Tampilan Salah Satu Bukti Capaian Borang AMI

Pada Gambar 19.45 merupakan tampilan salah satu bukti capaian borang AMI yang digunakan sebagai manipulasi data pemberkasan borang AMI.

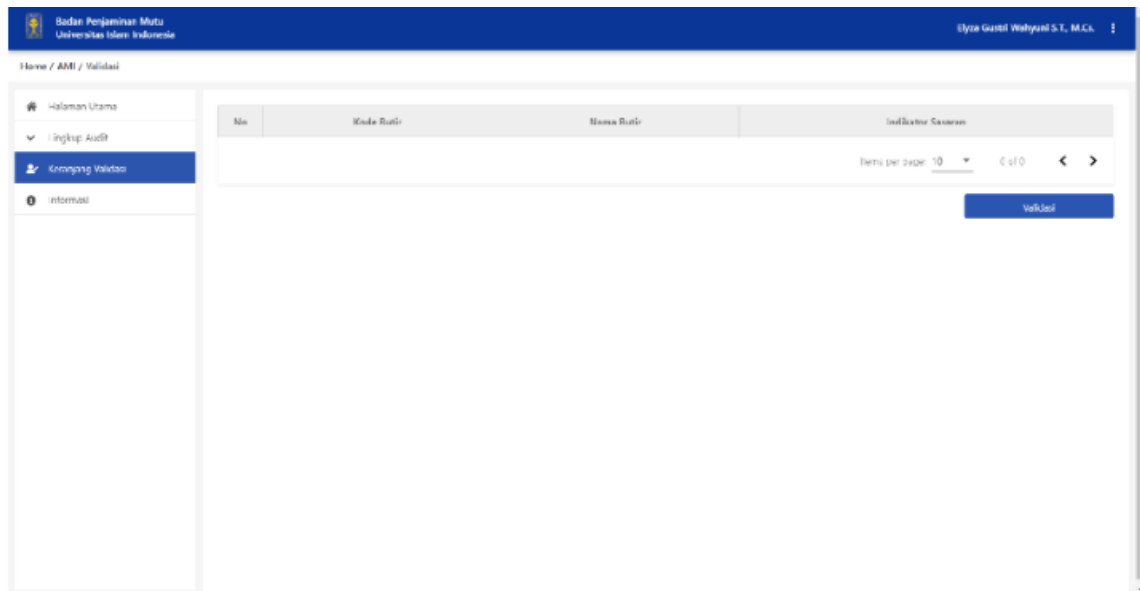
## 7. Halaman Rekomendasi Tindak Lanjut



Gambar 19.46 Tampilan Halaman Rekomendasi Tindak Lanjut

Pada Gambar 19.46 merupakan tampilan halaman rekomendasi tindak lanjut yang digunakan sebagai pendataan temuan-temuan yang dijadikan rekomendasi keberlanjutan suatu unit.

## 8. Halaman Validasi Borang AMI



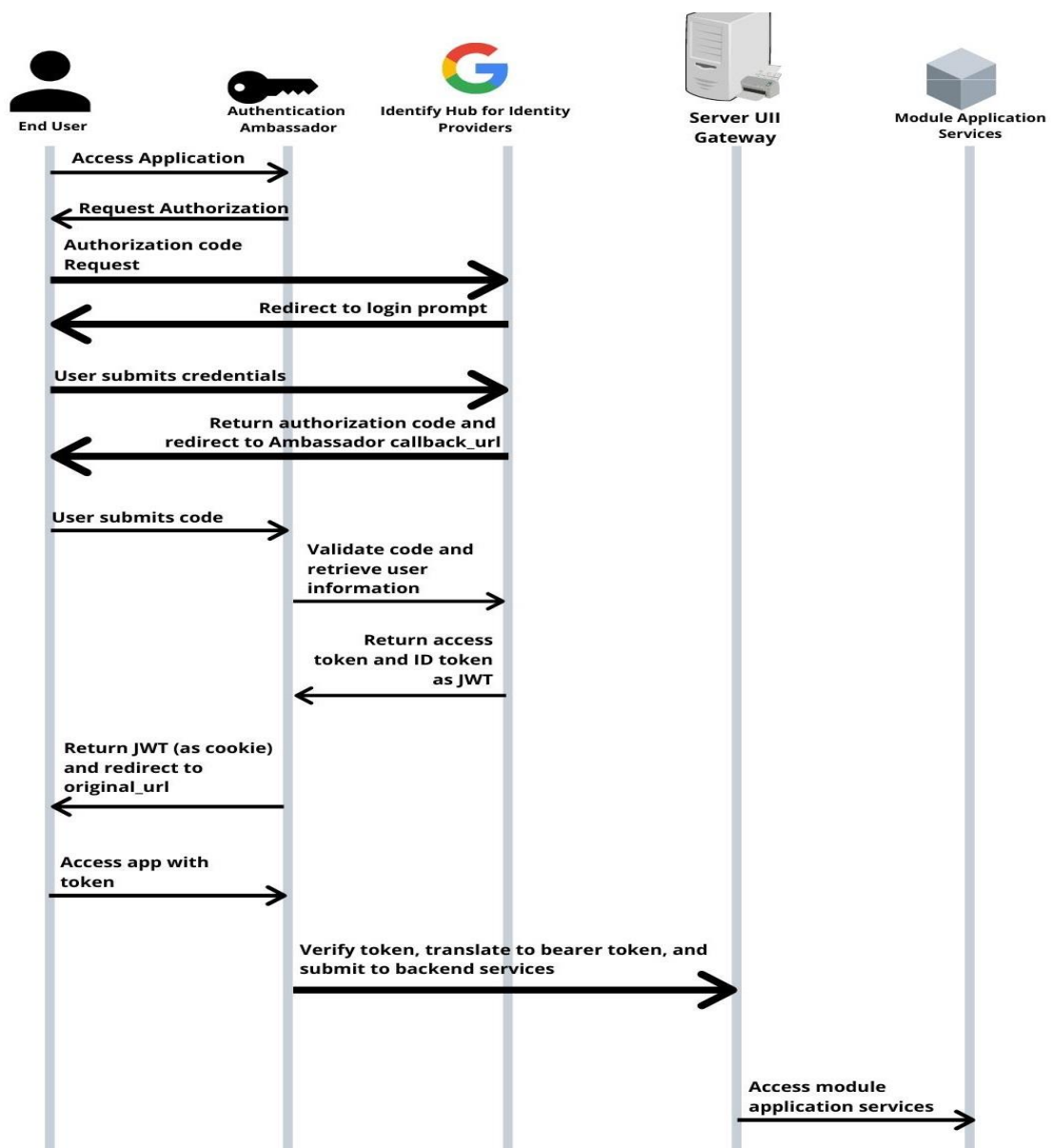
Gambar 19.47 Tampilan Validasi Borang AMI

Pada Gambar 19.47 merupakan tampilan validasi borang AMI yang digunakan ketika terdapat butir dari borang AMI yang sudah terisi maka akan tampil/masuk ke dalam keranjang validasi borang AMI.

## 3. Pembahasan

Dari hasil semua tahapan yang telah dilakukan, pada tahap *deployment* merupakan tahap pemasangan *web services* agar sudah siap untuk digunakan oleh pengguna. Proses *deployment* harus direncanakan dengan baik sehingga meminimalisir *down system* yang akan berdampak pada produktivitas pengguna. Tahap *deployment* memiliki berbagai macam tugas yang perlu dilakukan seperti *installasion*, *system build*, *change control*. Pada penelitian ini tahap *deployment web services* BPM UII, masih sebatas *localhost* untuk dapat dikonsumsi oleh aplikasi klien. Dikarenakan *web services* BPM UII masih akan berlanjut untuk dikembangkan sampai ke lingkup universitas. Setelah *web services* BPM UII sudah dikembangkan sampai lingkup universitas, nantinya akan digabung dengan sistem UIIGateway atau menjadi modul sistem aplikasi di sistem UIIGateway. Karena harapan kedepannya pengguna dengan memiliki berbagai keperluan dapat mengakses hanya satu sistem yaitu UIIGateway yang mempunyai

berbagai modul aplikasi dengan salah satunya yaitu modul sistem aplikasi BPM UII. Sebelum proses pemasangan/penggabungan ke dalam sistem UIIGateway, terlebih dahulu dilakukan pengemasan ke dalam *container* docker. Docker merupakan satu platform yang dibangun berdasarkan *container*/wadah. Untuk otentikasi akan menggunakan aplikasi pihak ketiga yaitu Ambassador Authentication, sedangkan otentikasi JWT(JSON Web Token Authentication) akan tidak digunakan lagi. Ambassador Authentication merupakan aplikasi pihak ketiga yang dapat mengotentikasi permintaan yang masuk sebelum merutekannya ke layanan dukungan. Arsitektur tersebut dapat dilihat pada Gambar 19.48.



Gambar 19.48 Arsitektur Ambassador di UIIGatewa