

# DOKUMENTASI BACKEND

## Menu Mata Pelajaran

- Membuat Mata Pelajaran

Membuat data mata pelajaran ini didapat dari code file API yaitu menggunakan code :

```
Route::post('mapel', [Mapel::class, '_create']);
```

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:7500/api/mapel`. The request body is in the 'Form' tab, with a field 'mapel' containing the value 'Bahasa Lampung'. The response is a 201 Created status with a JSON body: `{ "catatan": "Mata Pelajaran Berhasil Di Input", "aksi": true }`.

- Mengambil Semua Data Mata Pelajaran

Mengambil semua data Mata Pelajaran ini didapat dari code file API yaitu menggunakan code :

```
Route::get('mapel', [Mapel::class, '_read']);
```

The screenshot shows a REST client interface with a GET request to `http://127.0.0.1:7500/api/mapel`. The response is a 200 OK status with a JSON array of subject data. The data includes fields for id, nama\_mapel, created\_at, and updated\_at.

id	nama_mapel	created_at	updated_at
1	STAFF TU	2023-05-20T07:15:36.000000Z	2023-05-20T07:17:09.000000Z
2	IPA	2023-05-20T07:15:44.000000Z	2023-05-20T07:15:44.000000Z
3	IPS	2023-05-20T07:15:52.000000Z	2023-05-20T07:15:52.000000Z
4	BIOLOGI	2023-05-20T07:16:07.000000Z	2023-05-20T07:16:07.000000Z
5	UTAMA		

- Menghapus Mata Pelajaran

Menghapus Mata Pelajaran ini didapat dari code file API yaitu menggunakan code :

```
Route::delete('mapel/{parameter}', [Mapel::class, '_delete']);
```

DELETE  http://127.0.0.1:7500/api/mapel/6

Status: 200 OK Size: 0 Bytes Time: 1.05 s

Query Headers<sup>2</sup> Auth Body Tests Pre Run

Response Headers<sup>10</sup> Cookies<sup>1</sup> Results Docs

Query Parameters

<input type="checkbox"/>	parameter	value
--------------------------	-----------	-------

1 {  
2 "catatan": "Mata Pelajaran Berhasil Dihapus",  
3 "aksi": true  
4 }

- Mengupdate Mata Pelajaran

Mengupdate Mata Pelajaran ini didapat dari code file API yaitu menggunakan code :

```
Route::post('mapel_update', [Mapel::class, '_update']);
```

POST  http://127.0.0.1:7500/api/mapel\_update

Status: 200 OK Size: 0 Bytes Time: 1.13 s

Query Headers<sup>2</sup> Auth Body<sup>1</sup> Tests Pre Run

Response Headers<sup>10</sup> Cookies<sup>1</sup> Results Docs

JSON XML Text Form Form-encode GraphQL Binary

Form Fields

<input checked="" type="checkbox"/>	mapel	Bahasa Lampung 2
<input checked="" type="checkbox"/>	id	7
<input type="checkbox"/>	field name	value

☐ Files

1 {  
2 "catatan": "Mata Pelajaran Berhasil Diubah",  
3 "aksi": true  
4 }

## MENU KELAS

- Membuat Kelas

Perintah Membuat Kelas ini didapat dari code file API yaitu menggunakan code :

```
Route::post('kelas', [Kelas::class, '_create']);
```

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:7500/api/kelas
- Status:** 201 Created
- Size:** 0 Bytes
- Time:** 896 ms
- Response:**

```
{  "catatan": "Kelas Berhasil Di Input",  "aksi": true}
```
- Form Fields:** A table with two rows: one for 'kelas' (checked) with value 'XI - A', and one for 'field name' (unchecked) with value 'value'.

- Mengambil Semua Data Kelas

Perintah Mengambil Semua Data Kelas didapat dari code file API yaitu menggunakan code :

```
Route::get('kelas', [Kelas::class, '_read']);
```

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:7500/api/kelas
- Status:** 200 OK
- Size:** 0 Bytes
- Time:** 1.05 s
- Response:**

```
{  "kelas": [    {      "id": 1,      "nama_kelas": "KANTOR",      "created_at": "2023-05-20T07:16:51.000000Z",      "updated_at": "2023-05-20T07:16:51.000000Z"    },    {      "id": 2,      "nama_kelas": "X - A",      "created_at": "2023-05-20T07:17:33.000000Z",      "updated_at": "2023-05-20T07:17:33.000000Z"    },    {      "id": 3,      "nama_kelas": "X - B",      "created_at": "2023-05-20T07:17:43.000000Z",      "updated_at": "2023-05-20T07:17:43.000000Z"    },    {      "id": 4,      "nama_kelas": "X - C",      "created_at": "2023-05-20T07:17:53.000000Z",      "updated_at": "2023-05-20T07:17:53.000000Z"    }  ]}
```
- Form Fields:** A table with one row: 'field name' (unchecked) with value 'value'.

- Menghapus Data Kelas

Perintah Menghapus Data Kelas didapat dari code file API yaitu menggunakan code :

```
Route::delete('kelas/{parameter}', [Kelas::class, '_delete']);
```

DELETE

Query Headers<sup>2</sup> Auth **Body** Tests Pre Run

JSON XML Text **Form** Form-encode GraphQL Binary

Form Fields ☐ Files

☐ field name value

Status: 200 OK Size: 0 Bytes Time: 907 ms

Response Headers<sup>10</sup> Cookies<sup>1</sup> Results Docs

```
1 {
2   "catatan": "Kelas Berhasil Dihapus",
3   "aksi": true
4 }
```

- Mengupdate Data Kelas

Perintah Mengupdate Data Kelas didapat dari code file API yaitu menggunakan code :

```
Route::post('kelas_update', [Kelas::class, '_update']);
```

POST

Query Headers<sup>2</sup> Auth **Body<sup>1</sup>** Tests Pre Run

JSON XML Text **Form** Form-encode GraphQL Binary

Form Fields ☐ Files

☒ nama\_kelas Ruang Guru

☒ id 1

☐ field name value

Status: 200 OK Size: 0 Bytes Time: 924 ms

Response Headers<sup>10</sup> Cookies<sup>1</sup> Results Docs

```
1 {
2   "catatan": "Kelas Berhasil Diubah",
3   "aksi": true
4 }
```

## MENU GURU

- Membuat Login Guru

Perintah Membuat Login Guru didapat dari code file API yaitu menggunakan code :

```
Route::post('login_guru', [Guru::class, '_login']);
```

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:7500/api/login\_guru
- Status:** 200 OK
- Size:** 0 Bytes
- Time:** 842 ms
- Body Type:** Form
- Form Fields:**
  - ☒ user: na
  - ☒ pass: na
  - ☐ field name: value
- Response:**

```
1 {
2   "catatan": "Anda Berhasil Login",
3   "aksi": true,
4   "id": 4,
5   "nama": "Erna Novita"
6 }
```

- Membuat Guru

Perintah Membuat Guru didapat dari code file API yaitu menggunakan code:

```
Route::post('guru', [Guru::class, '_create']);
```

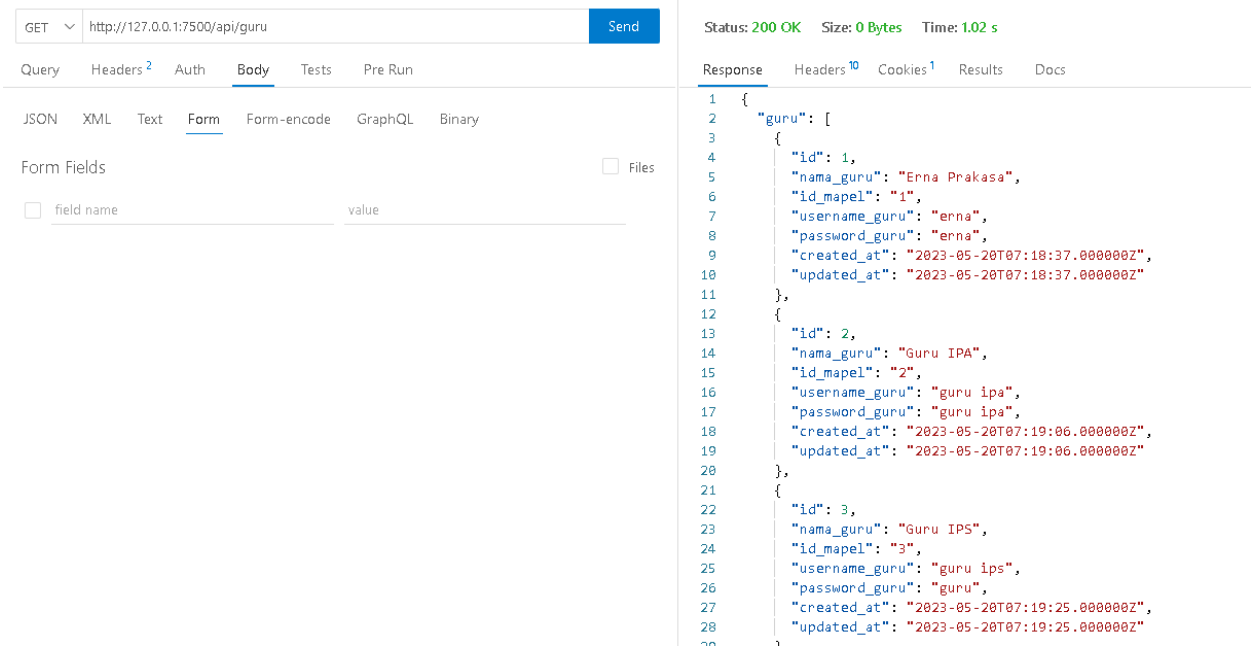
The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:7500/api/guru
- Status:** 201 Created
- Size:** 0 Bytes
- Time:** 1.03 s
- Body Type:** Form
- Form Fields:**
  - ☒ nama: Erna Novita
  - ☒ id: 1
  - ☒ user: na
  - ☒ pass: na
  - ☐ field name: value
- Response:**

```
1 {
2   "catatan": "User Guru Berhasil Di Input",
3   "aksi": true
4 }
```

- Mengambil Semua Data Guru  
Perintah Mengambil Semua Data Guru didapat dari code file API yaitu menggunakan code :

```
Route::get('guru', [Guru::class, '_read']);
```



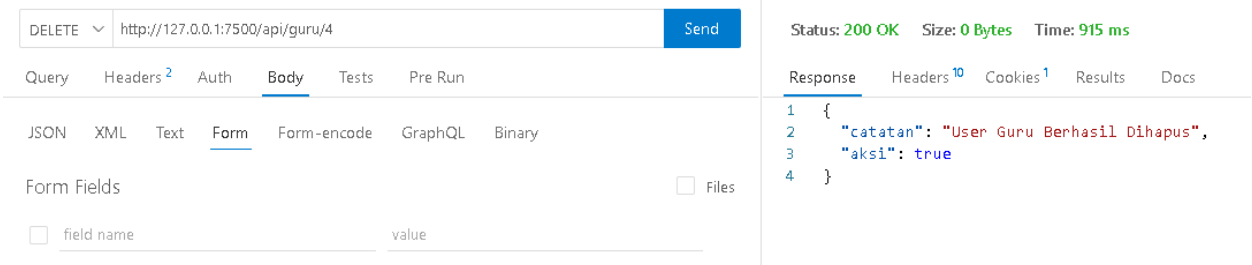
The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:7500/api/guru
- Status:** 200 OK
- Size:** 0 Bytes
- Time:** 1.02 s
- Response:** A JSON array of three guru records.

```
{
  "guru": [
    {
      "id": 1,
      "nama_guru": "Erna Prakasa",
      "id_mapel": "1",
      "username_guru": "erna",
      "password_guru": "erna",
      "created_at": "2023-05-20T07:18:37.000000Z",
      "updated_at": "2023-05-20T07:18:37.000000Z"
    },
    {
      "id": 2,
      "nama_guru": "Guru IPA",
      "id_mapel": "2",
      "username_guru": "guru ipa",
      "password_guru": "guru ipa",
      "created_at": "2023-05-20T07:19:06.000000Z",
      "updated_at": "2023-05-20T07:19:06.000000Z"
    },
    {
      "id": 3,
      "nama_guru": "Guru IPS",
      "id_mapel": "3",
      "username_guru": "guru ips",
      "password_guru": "guru",
      "created_at": "2023-05-20T07:19:25.000000Z",
      "updated_at": "2023-05-20T07:19:25.000000Z"
    }
  ]
}
```

- Menghapus Data Guru  
Perintah Menghapus Data Guru didapat dari code file API yaitu menggunakan code :

```
Route::delete('guru/{parameter}', [Guru::class, '_delete']);
```



The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** http://127.0.0.1:7500/api/guru/4
- Status:** 200 OK
- Size:** 0 Bytes
- Time:** 915 ms
- Response:** A JSON object indicating successful deletion.

```
{
  "catatan": "User Guru Berhasil Dihapus",
  "aksi": true
}
```

- Mengupdate Data Guru  
Perintah Mengupdate Data Guru didapat dari code file API yaitu menggunakan code :

```
Route::post('guru_update', [Guru::class, '_update']);
```

POST	http://127.0.0.1:7500/api/guru_update	Send	Status: 200 OK	Size: 0 Bytes	Time: 929 ms
Query	Headers <sup>2</sup>	Auth	Body <sup>1</sup>	Tests	Pre Run
JSON	XML	Text	Form	Form-encode	GraphQL
Form Fields					
<input checked="" type="checkbox"/>	id	1			
<input checked="" type="checkbox"/>	id_mapel	1			
<input checked="" type="checkbox"/>	user	erna			
<input checked="" type="checkbox"/>	pass	erna			
<input checked="" type="checkbox"/>	nama	Erna Novianto			
<input type="checkbox"/>	field name	value			
			Files		
			Response	Headers <sup>10</sup>	Cookies <sup>1</sup>
			<pre> 1 { 2   "catatan": "User Guru Berhasil Diubah", 3   "aksi": true 4 }</pre>		

## MENU ADMIN

- Membuat Login Admin  
Perintah Membuat Login Admin didapat dari code file API yaitu menggunakan code :

```
Route::post('login_admin', [Admin::class, '_login']);
```

POST	http://127.0.0.1:7500/api/login_admin	Send	Status: 200 OK	Size: 0 Bytes	Time: 968 ms
Query	Headers <sup>2</sup>	Auth	Body <sup>1</sup>	Tests	Pre Run
JSON	XML	Text	Form	Form-encode	GraphQL
Form Fields					
<input checked="" type="checkbox"/>	user	ernaerna			
<input checked="" type="checkbox"/>	pass	ernaerna			
<input type="checkbox"/>	field name	value			
			Files		
			Response	Headers <sup>10</sup>	Cookies <sup>1</sup>
			<pre> 1 { 2   "catatan": "Anda Berhasil Login", 3   "aksi": true, 4   "data": { 5     "id": 4, 6     "nama_admin": "Erna Novita", 7     "username_admin": "ernaerna", 8     "password_admin": "ernaerna", 9     "created_at": "2023-05-30T14:11:57.000000Z", 10    "updated_at": "2023-05-30T14:11:57.000000Z" 11  } 12 }</pre>		

- Membuat Admin

Perintah Membuat Admin didapat dari code file API yaitu menggunakan code :

```
Route::post('admin', [Admin::class, '_create']);
```

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:7500/api/admin
- Status:** 201 Created
- Size:** 0 Bytes
- Time:** 730 ms
- Body:** Form Fields
- Form Fields:**
  - ☒ admin: Erna Novita
  - ☒ user: ernaerna
  - ☒ pass: ernaerna
  - ☐ field name: value
- Response:**

```
1 {
2   "catatan": "User Admin Berhasil Di Input",
3   "aksi": true
4 }
```

- Mengambil Semua Data Admin

Perintah Mengambil Semua Data Admin didapat dari code file API yaitu menggunakan code :

```
Route::get('admin', [Admin::class, '_read']);
```

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:7500/api/admin
- Status:** 200 OK
- Size:** 0 Bytes
- Time:** 1.01 s
- Body:** Form Fields
- Form Fields:**
  - ☐ field name: value
- Response:**

```
1 {
2   "admin": [
3     {
4       "id": 2,
5       "nama_admin": "Erna",
6       "username_admin": "erna",
7       "password_admin": "erna",
8       "created_at": "2023-05-20T07:15:01.000000Z",
9       "updated_at": "2023-05-20T07:15:01.000000Z"
10    },
11    {
12      "id": 3,
13      "nama_admin": "Admin",
14      "username_admin": "admin",
15      "password_admin": "admin",
16      "created_at": "2023-05-20T07:19:44.000000Z",
17      "updated_at": "2023-05-20T07:19:44.000000Z"
18    }
19  ]
20 }
```



- Menghapus Data Admin

Perintah Menghapus Data Admin didapat dari code file API yaitu menggunakan code :

```
Route::delete('admin/{parameter}', [Admin::class, '_delete']);
```

DELETE

Status: 200 OK Size: 0 Bytes Time: 1.18 s

Query Headers<sup>2</sup> Auth **Body** Tests Pre Run

JSON XML Text **Form** Form-encode GraphQL Binary

Form Fields ☐ Files

☐ field name value

Response Headers<sup>10</sup> Cookies<sup>1</sup> Results Docs

```
1 {
2   "catatan": "User Admin Berhasil Dihapus",
3   "aksi": true
4 }
```

- Mengupdate Data Admin

Perintah Mengupdate Data Admin didapat dari code file API yaitu menggunakan code :

```
Route::post('admin_update', [Admin::class, '_update']);
```

POST

Status: 200 OK Size: 0 Bytes Time: 935 ms

Query Headers<sup>2</sup> Auth **Body<sup>1</sup>** Tests Pre Run

JSON XML Text **Form** Form-encode GraphQL Binary

Form Fields ☐ Files

☒ id 2

☒ user erna

☒ pass erna

☒ admin Erna Novianto

☐ field name value

Response Headers<sup>10</sup> Cookies<sup>1</sup> Results Docs

```
1 {
2   "catatan": "User Admin Berhasil Diubah",
3   "aksi": true
4 }
```

## MENU ABSEN

- Guru Absen

Perintah Guru Absen didapat dari code file API yaitu menggunakan code :

```
Route::post('absen', [Absen::class, '_create']);
```

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:7500/api/absen
- Status:** 201 Created
- Size:** 0 Bytes
- Time:** 929 ms
- Body:** The 'Form' tab is selected. It contains two checked fields: 'id\_k' with value '1' and 'id\_g' with value '1'. There is also an unchecked field 'field name' with value 'value'.
- Response:** The 'Response' tab shows a JSON object: 

```
{  "catatan": "Absen Hari Ini Berhasil Di Input",  "aksi": true}
```

- Mengambil Data Absen

Perintah Mengambil Data Absen didapat dari code file API yaitu menggunakan code :

```
Route::get('absen', [Absen::class, '_read']);
```

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:7500/api/absen
- Status:** 200 OK
- Size:** 0 Bytes
- Time:** 1.24 s
- Form Fields:** The 'Form' tab is selected. It contains one unchecked field: 'field name' with value 'value'.
- Response:** The 'Response' tab shows a JSON array of three objects: 

```
{  "absen": [    {      "id": 3,      "tanggal": "20",      "bulan": "05",      "tahun": "2023",      "nama_guru": "Erna Novianto",      "nama_kelas": "X - B"    },    {      "id": 2,      "tanggal": "20",      "bulan": "05",      "tahun": "2023",      "nama_guru": "Erna Novianto",      "nama_kelas": "X - A"    },    {      "id": 1,      "tanggal": "20",      "bulan": "05",      "tahun": "2023",      "nama_guru": "Erna Novianto",      "nama_kelas": "Ruang Guru"    }  ]}
```

- Menghapus Data Absen  
Perintah Menghapus Data Absen didapat dari code file API yaitu menggunakan code :

```
Route::delete('absen/{parameter}', [Absen::class, '_delete']);
```

The screenshot displays a REST client interface with the following details:

- Request:**
  - Method: DELETE
  - URL: `http://127.0.0.1:7500/api/absen/5`
  - Buttons: Query, Headers<sup>2</sup>, Auth, Body (selected), Tests, Pre Run, Send
  - Body Tabs: JSON, XML, Text, Form (selected), Form-encode, GraphQL, Binary
  - Form Fields: A table with columns 'field name' and 'value'.
  - Files: ☐ Files
- Response:**
  - Status: 200 OK
  - Size: 0 Bytes
  - Time: 842 ms
  - Buttons: Response (selected), Headers<sup>10</sup>, Cookies<sup>1</sup>, Results, Docs
  - Response Body (JSON):

```
1 {  
2   "catatan": "Absen Berhasil Dihapus",  
3   "aksi": true  
4 }
```