

Dokumentasi Backend

USER ADMIN

- Login Admin

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8000/api/login`. The request body is in Form format with two fields: `email` and `password`, both with the value `admin2`. The response status is `200 OK` with a size of `144 Bytes` and a time of `947 ms`. The response body is a JSON object:

```
{
  "pesan": "Anda Berhasil Login",
  "pesan_b": true,
  "data": {
    "attributes": {},
    "request": {},
    "query": {},
    "server": {},
    "files": {},
    "cookies": {},
    "headers": {}
  }
}
```

Validasi Data login ini didapat dari code file API yaitu menggunakan code `route::post('user_admin', [UserAdminController::class, 'buat_user'])`;

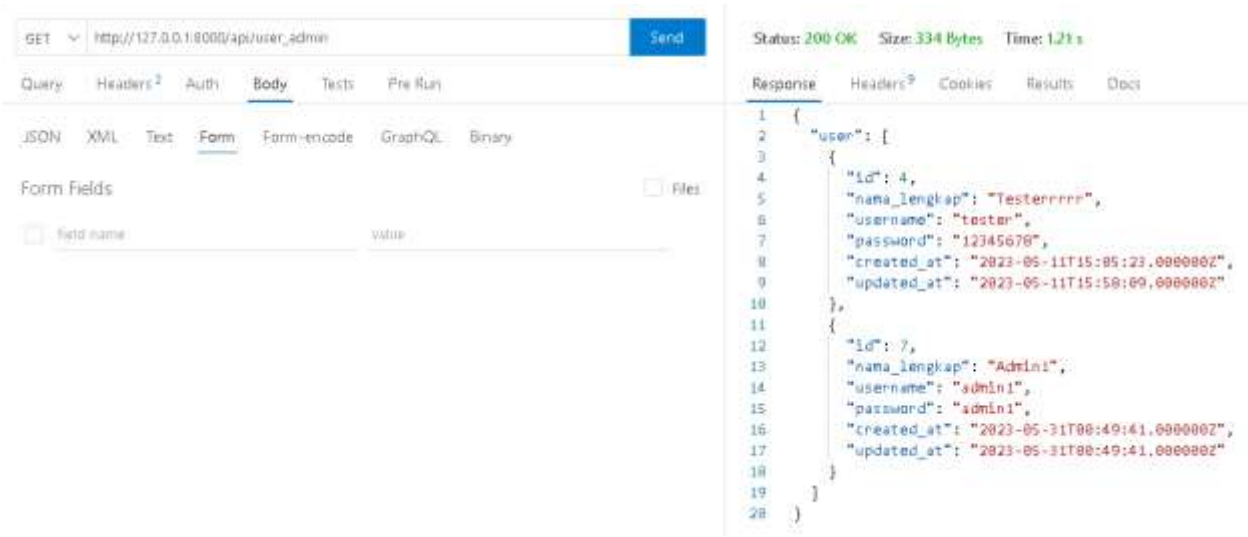
- Membuat User Admin

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8000/api/user_admin`. The request body is in Form format with three fields: `nama`, `email`, and `password`, all with the value `admin1`. The response status is `201 Created` with a size of `55 Bytes` and a time of `1.13 s`. The response body is a JSON object:

```
{
  "pesan": "User Admin Berhasil Di Input",
  "pesan_b": true
}
```

Membuat user Admin didapat dari code file API yaitu menggunakan Code `Route::post('user_peminjam', [UserPinjamController::class, 'buat_user'])`;

- Mengambil Semua Data User Admin



GET http://127.0.0.1:8000/api/user_admin Send

Status: 200 OK Size: 334 Bytes Time: 1.21 s

Query Headers² Auth Body Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

Form Fields ☐ Files

field name value

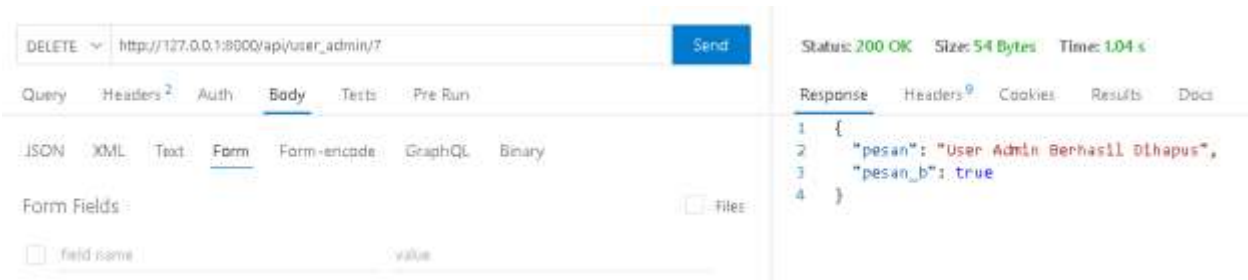
```

1 {
2   "user": [
3     {
4       "id": 4,
5       "nama_lengkap": "Testerrrrr",
6       "username": "testerr",
7       "password": "12345678",
8       "created_at": "2023-05-11T15:05:23.000000Z",
9       "updated_at": "2023-05-11T15:58:09.000000Z"
10    },
11    {
12      "id": 7,
13      "nama_lengkap": "Admin1",
14      "username": "admin1",
15      "password": "admin1",
16      "created_at": "2023-05-31T00:49:41.000000Z",
17      "updated_at": "2023-05-31T00:49:41.000000Z"
18    }
19  ]
20 }

```

Pengambilan data Semua data user admin di dapat dari API, code
 Route::post('user_admin', [UserAdminController::class, 'buat_user']);

- Menghapus User Admin



DELETE http://127.0.0.1:8000/api/user_admin/7 Send

Status: 200 OK Size: 54 Bytes Time: 1.04 s

Query Headers² Auth Body Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

Form Fields ☐ Files

field name value

```

1 {
2   "pesan": "User Admin Berhasil Dihapus",
3   "pesan_b": true
4 }

```

Penghapusan data user admin di dapat dari API, code
 Route::delete('user_admin/{parameter}', [UserAdminController::class, 'hapus_user']);

- Mengupdate User Admin

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

- Method:** POST
- URL:** http://127.0.0.1:8000/api/user_admin_update
- Body:** Form (selected)
- Form Fields:**
 - email: admin2
 - password: admin2
 - id: 7
 - nama: Admin2
- Status:** 200 OK
- Size:** 53 Bytes
- Time:** 1.03 s
- Response:**

```
{
  "pesan": "User Admin Berhasil Diubah",
  "pesan_b": true
}
```

Pengupdatean Data User Admin di dapat dari API, Code
 Route::post('user_admin_update', [UserAdminController::class, 'ubah_user']);

USER PINJAM

- Login Peminjam

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

- Method:** POST
- URL:** http://127.0.0.1:8000/api/login_peminjam
- Body:** Form (selected)
- Form Fields:**
 - email: wilyandi
 - password: wil
- Status:** 200 OK
- Size:** 193 Bytes
- Time:** 7.26 ms
- Response:**

```
{
  "pesan": "Anda Berhasil Login",
  "pesan_b": true,
  "data": {
    "attributes": {},
    "request": {},
    "query": {},
    "server": {},
    "files": {},
    "cookies": {},
    "headers": {}
  },
  "id_user": "b684bda8-355a-4ecc-97bc-81f2cad8a076"
}
```

Login user pinjam untuk mengetahui siapa saja yang meminjam yang didapat dari API, Code

Route::post('user_peminjam', [UserPinjamController::class, 'buat_user']);

- Membuat User Peminjam

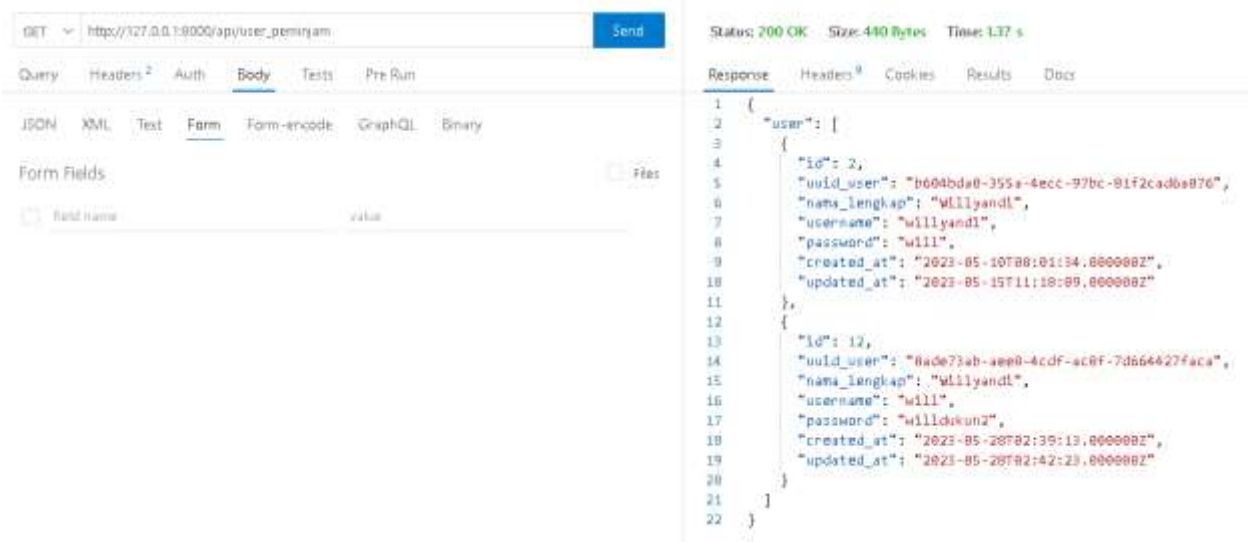
The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:8000/api/user_peminjam
- Body Type:** Form
- Form Fields:**
 - ☒ email: manman
 - ☒ password: suwar
 - ☒ nama: Suwarman Adi Wardoyok
 - ☐ field name: value
- Status:** 201 Created
- Size:** 58 Bytes
- Time:** 1.13 s
- Response:**

```
1 {
2   "pesan": "User Peminjam Berhasil Di Input",
3   "pesan_b": true
4 }
```

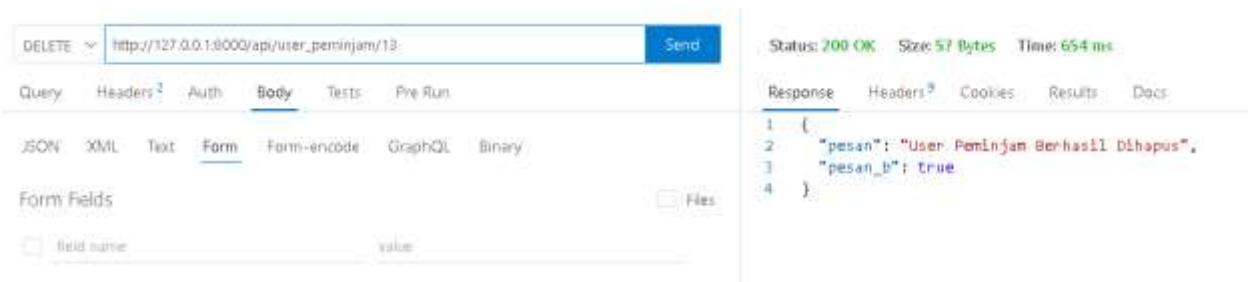
Untuk membuat User Peminjam di dapat Dari API, code
Route::post('user_peminjam', [UserPinjamController::class, 'buat_user']);

- Mengambil Semua Data User Peminjam



Pengambilan Semua data user pinjam di dapat dari API, code
`Route::get('user_peminjam', [UserPinjamController::class, 'semua_data_user']);`

- Menghapus User Peminjam



Menghapus user peminjam yang didapat dari routes API Code,
`Route::delete('user_peminjam/{parameter}', [UserPinjamController::class, 'hapus_user']);`

- Mengupdate User Peminjam

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8000/api/user_peminjam_update`. The request body is in Form format with the following fields:

Field	Value
email	manman
password	suwarman
nama	Suwarman Adi Wardoyo
id	13
uuid_user	d5f027f3-d475-420f-991c-d5ca2b1fba1
field name	value

The response status is 200 OK, with a size of 56 bytes and a time of 1.01 s. The response body is in JSON format:

```
{
  "pesan": "User Peminjam Berhasil Diubah",
  "pesan_b": true
}
```

Mengupdate User peminjam yang didapat dari routes API, Code
 Route::post('user_peminjam_update', [UserPinjamController::class, 'ubah_user']);

BUKU

- Membuat Buku

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8000/api/buku`. The request body is in Form format with the following fields:

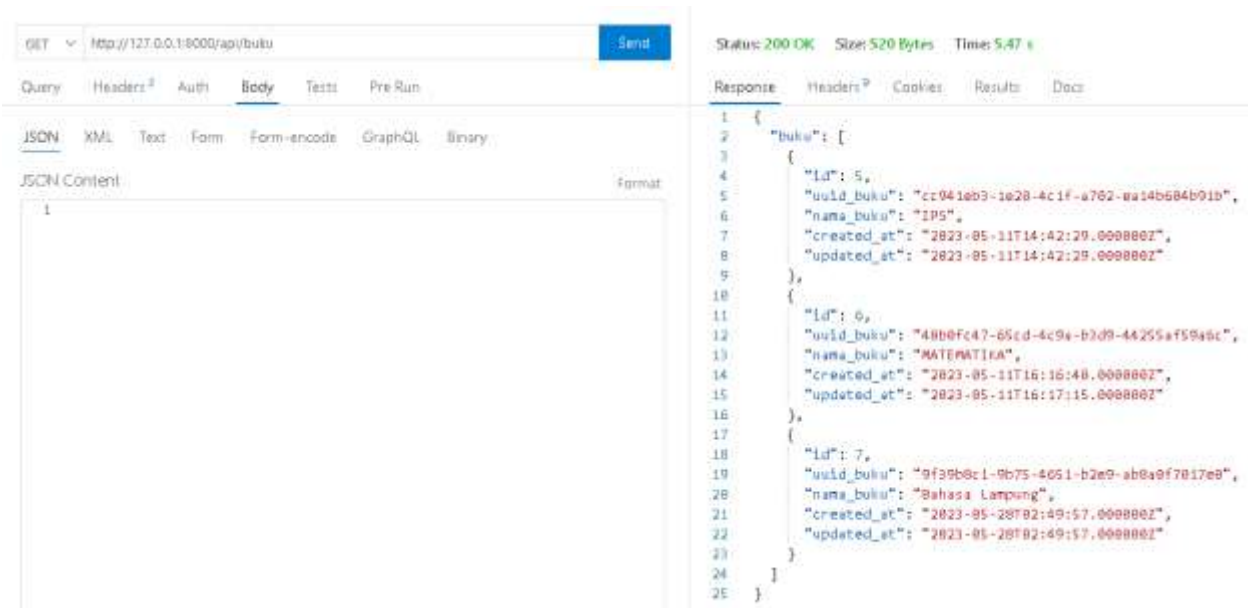
Field	Value
buku	Bahasa Indonesia
field name	value

The response status is 201 Created, with a size of 49 bytes and a time of 3.65 s. The response body is in JSON format:

```
{
  "pesan": "Buku Berhasil Di Input",
  "pesan_b": true
}
```

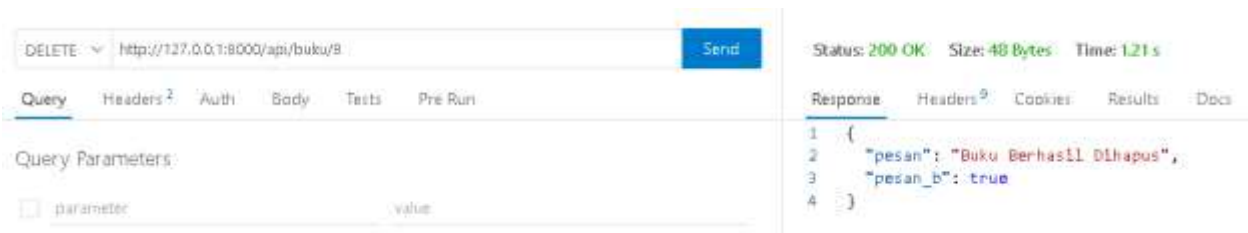
Membuat buku, buku apa saja yang akan di masukkan yang di dapat dari routes
 API, Code Route::post('buku', [BukuController::class, 'buat_buku']);

- Mengambil Semua Data Buku



Mengambil semua data buku untuk memeriksa buku yang di pinjam didapat dari routes API, code `Route::get('buku', [BukuController::class, 'semua_data_buku'])`;

- Menghapus Data Buku



Mengapus Data Buku di dapat dari routes API, Code `Route::delete('buku/{parameter}', [BukuController::class, 'hapus_buku'])`;

- Mengupdate Data Buku

POST `http://127.0.0.1:8000/api/buku_update` Send

Query Headers² Auth **Body¹** Tests Pre Run

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

Form Fields Files

<input checked="" type="checkbox"/>	buku	Bahasa Inggris
<input checked="" type="checkbox"/>	id	8
<input type="checkbox"/>	field name	value

Status: **200 OK** Size: 47 Bytes Time: 1.05 s

Response Headers⁵ Cookies Results Docs

```

1 {
2   "pesan": "Buku Berhasil Diubah",
3   "pesan_b": true
4 }

```

Untuk mengupdate buku didapat dari routes API, Code Route::post('buku_update', [BukuController::class, 'ubah_buku']);

PINJAM BUKU

- Mengambil Semua Buku Yang Dipinjam Berdasarkan User

POST `http://127.0.0.1:8000/api/peminjam_token` Send

Query Headers² Auth **Body¹** Tests Pre Run

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

Form Encoded

<input checked="" type="checkbox"/>	id	8ade73ab-aea8-4cdf-ac9f-7d664427faca
<input type="checkbox"/>	name	value

Status: **200 OK** Size: 84 Bytes Time: 1.06 s

Response Headers⁹ Cookies Results Docs

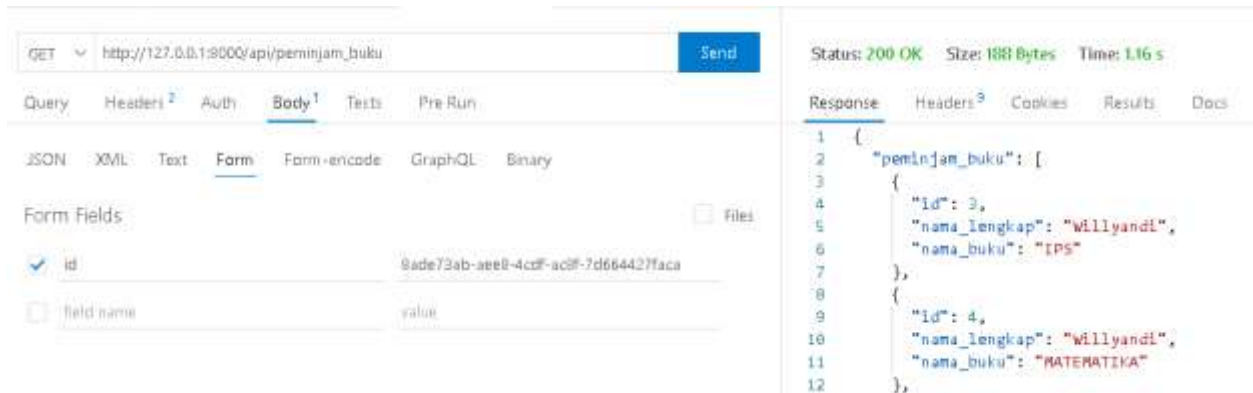
```

1 {
2   "peminjam_buku": [
3     {
4       "id": 4,
5       "nama_lengkap": "Willyandi",
6       "nama_buku": "Bahasa Lampung"
7     }
8   ]
9 }

```

Mengambil Semua buku yang dipinjam berdasarkan user yang di dapat dari routes API, code Route::post('peminjam_buku', [PinjamBukuController::class, 'buat_pinjam_buku']);

- Mengambil Semua Buku Yang Dipinjam



GET http://127.0.0.1:8000/api/peminjam_buku Send

Status: 200 OK Size: 188 Bytes Time: 1.16 s

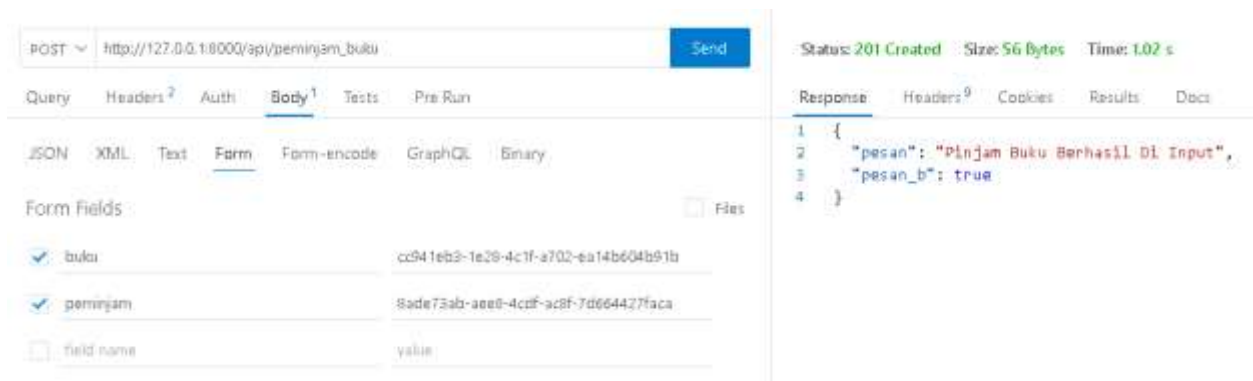
Response Headers Cookies Results Docs

```

1 {
2   "peminjam_buku": [
3     {
4       "id": 3,
5       "nama_lengkap": "Willyandi",
6       "nama_buku": "IPS"
7     },
8     {
9       "id": 4,
10      "nama_lengkap": "Willyandi",
11      "nama_buku": "MATEMATIKA"
12    }
  ]
}
```

Mengambil semua buku yang dipinjam di dapat dari routes API code
 Route::get('peminjam_buku', [PinjamBukuController::class,
 'semua_data_peminjam_buku']);

- Meminjam Buku



POST http://127.0.0.1:8000/api/peminjam_buku Send

Status: 201 Created Size: 56 Bytes Time: 1.02 s

Response Headers Cookies Results Docs

```

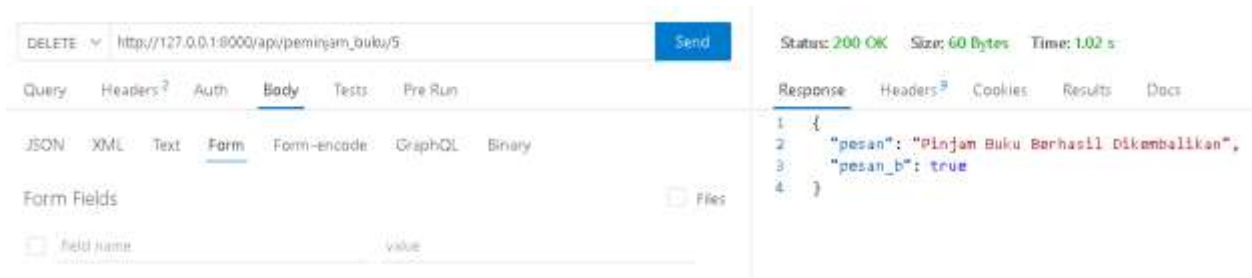
1 {
2   "pesan": "Pinjam Buku Berhasil Di Input",
3   "pesan_b": true
4 }
```

Form Fields

Field Name	Value
buku	cc941eb3-1e28-4c1f-a702-ea14b604b91b
peminjam	8ade73ab-aae8-4cdf-ac8f-7d664427faca

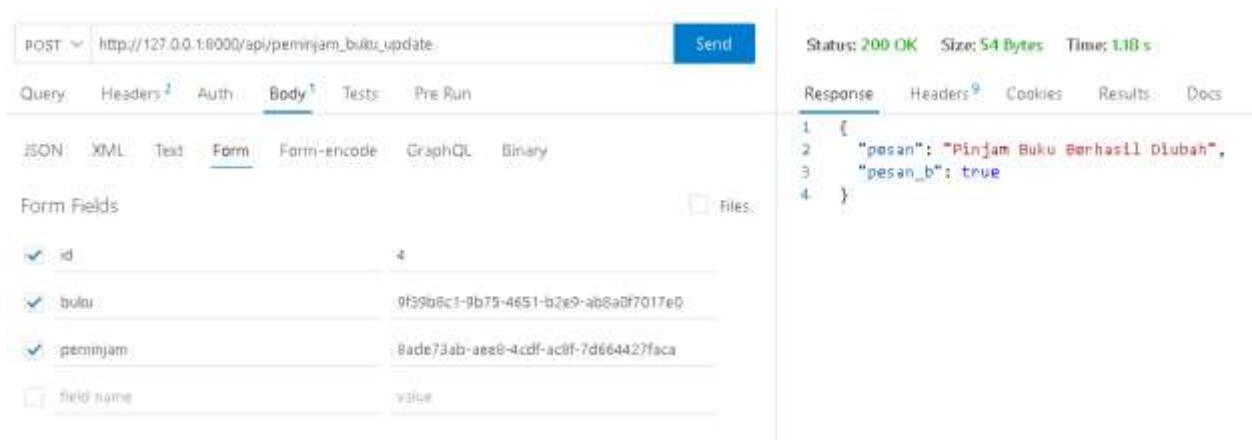
Meminjam buku di dapat dari routes API, Code Route::post('peminjam_buku',
 [PinjamBukuController::class, 'buat_pinjam_buku']);

- Mengembalikan Buku



Mengembalikan buku untuk buku apa yang dikembalikan, didapat dari routes API, code

- Mengupdate Buku Yang Di Pinjam



Mengupdate buku yang di pinjam di dapat dari routes API, Code
Route::post('peminjam_buku_update', [PinjamBukuController::class, 'ubah_pinjam_buku']);