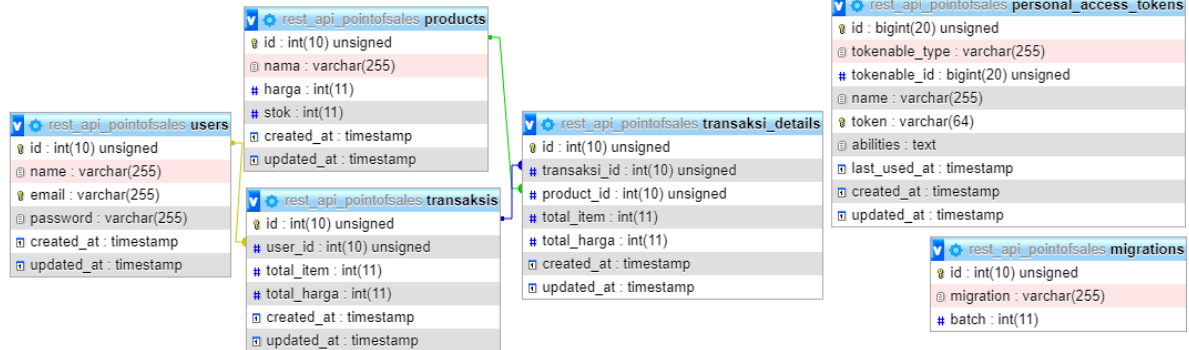


REST API Point of Sales dengan Laravel 8 dan MySQL

Adhi Dwi Saputro

Link github : <https://github.com/adisaputro17/rest-api-pointofsales-laravel8.git>

Desain database



Daftar route

\$ php artisan route:list					
Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	POST	api/login		App\Http\Controllers\UserController@login	api
	POST	api/logout		App\Http\Controllers\UserController@logout	api
	GET HEAD	api/product		App\Http\Controllers\ProductController@index	App\Http\Middleware\Authenticate:sanctum api
	POST	api/product		App\Http\Controllers\ProductController@store	App\Http\Middleware\Authenticate:sanctum api
	GET HEAD	api/product/{id}		App\Http\Controllers\ProductController@show	App\Http\Middleware\Authenticate:sanctum api
	PUT	api/product/{id}		App\Http\Controllers\ProductController@update	App\Http\Middleware\Authenticate:sanctum api
	DELETE	api/product/{id}		App\Http\Controllers\ProductController@delete	App\Http\Middleware\Authenticate:sanctum api
	POST	api/register		App\Http\Controllers\UserController@register	App\Http\Middleware\Authenticate:sanctum api
	GET HEAD	api/transaksi		App\Http\Controllers\TransaksiController@index	api
	POST	api/transaksi		App\Http\Controllers\TransaksiController@store	App\Http\Middleware\Authenticate:sanctum api
	GET HEAD	api/transaksi/{id}		App\Http\Controllers\TransaksiController@show	App\Http\Middleware\Authenticate:sanctum api
	DELETE	api/transaksi/{id}		App\Http\Controllers\TransaksiController@delete	App\Http\Middleware\Authenticate:sanctum api
	GET HEAD	api/user		App\Http\Controllers\UserController@user	App\Http\Middleware\Authenticate:sanctum api
	GET HEAD	sanctum/csrf-cookie		Laravel\Sanctum\Http\Controllers\CsrfCookieController@show	web

Register

Postman

File Edit View Help

+ New Import Runner My Workspace Invite

No Environment

POST localhost:8000/api/register

Untitled Request

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	kasir1	
<input checked="" type="checkbox"/> email	kasir1@gmail.com	
<input checked="" type="checkbox"/> password	kasir1	

Body Cookies Headers (10) Test Results

Status: 201 Created Time: 764 ms Size: 449 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "name": "kasir1",
3   "email": "kasir1@gmail.com",
4   "updated_at": "2021-04-20T04:02:13.000000Z",
5   "created_at": "2021-04-20T04:02:13.000000Z",
6   "id": 1
7 }
```

Find and Replace Console Bootcamp 11:02 AM 4/20/2021

Login

Postman

File Edit View Help

+ New Import Runner My Workspace Invite

No Environment

POST localhost:8000/api/login

Untitled Request

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> email	kasir1@gmail.com	
<input checked="" type="checkbox"/> password	kasir1	
Key	Value	Description

Body Cookies Headers (10) Test Results

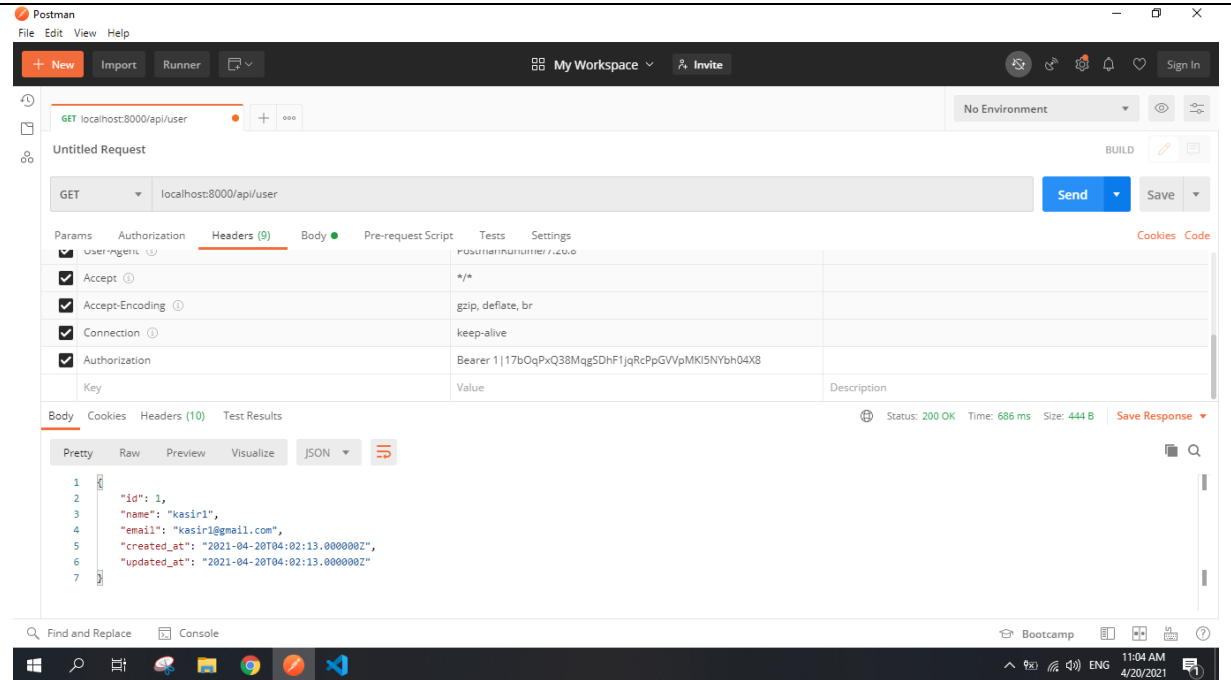
Status: 200 OK Time: 756 ms Size: 392 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Login successfully",
3   "token": "1|17b0qPxQ38HgSDnF1jRcPpGVVpWkISNybn04X8"
4 }
```

Find and Replace Console Bootcamp 11:02 AM 4/20/2021

Melihat data user yang sedang login

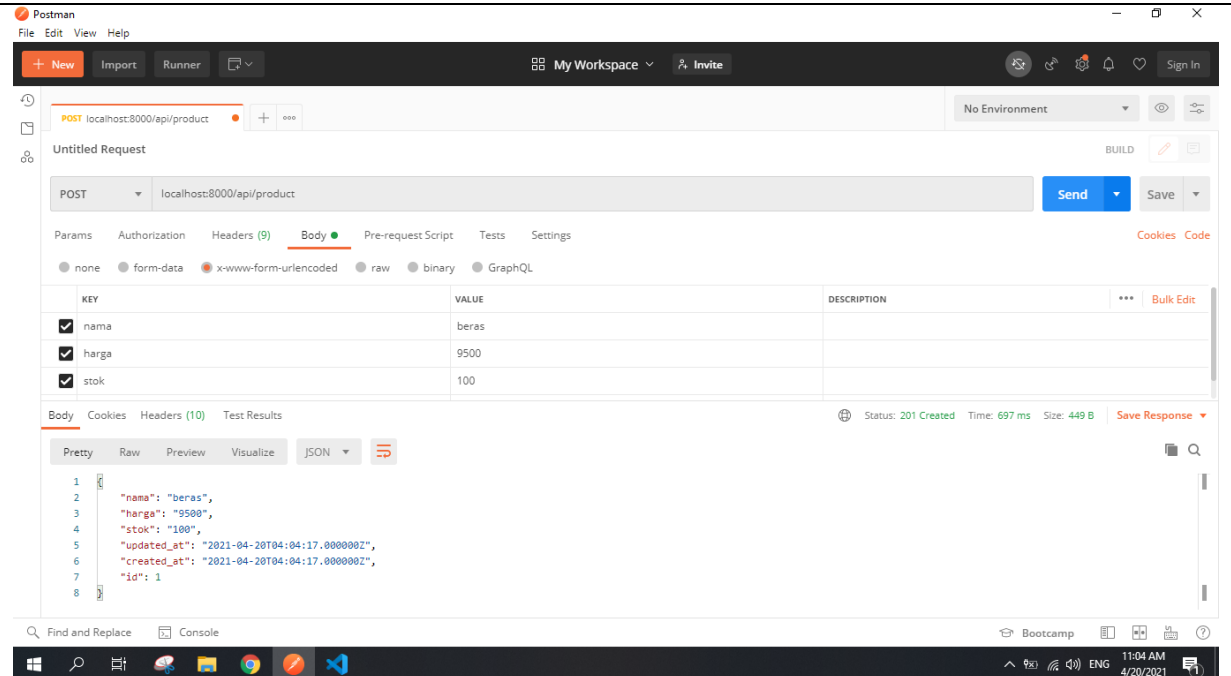


Postman interface showing a GET request to `localhost:8000/api/user`. The response is a JSON object with the following data:

```
{
  "id": 1,
  "name": "kasir1",
  "email": "kasir1@gmail.com",
  "created_at": "2021-04-20T04:02:13.000000Z",
  "updated_at": "2021-04-20T04:02:13.000000Z"
}
```

Status: 200 OK, Time: 686 ms, Size: 444 B

Menambahkan produk



Postman interface showing a POST request to `localhost:8000/api/product`. The request body is raw JSON:

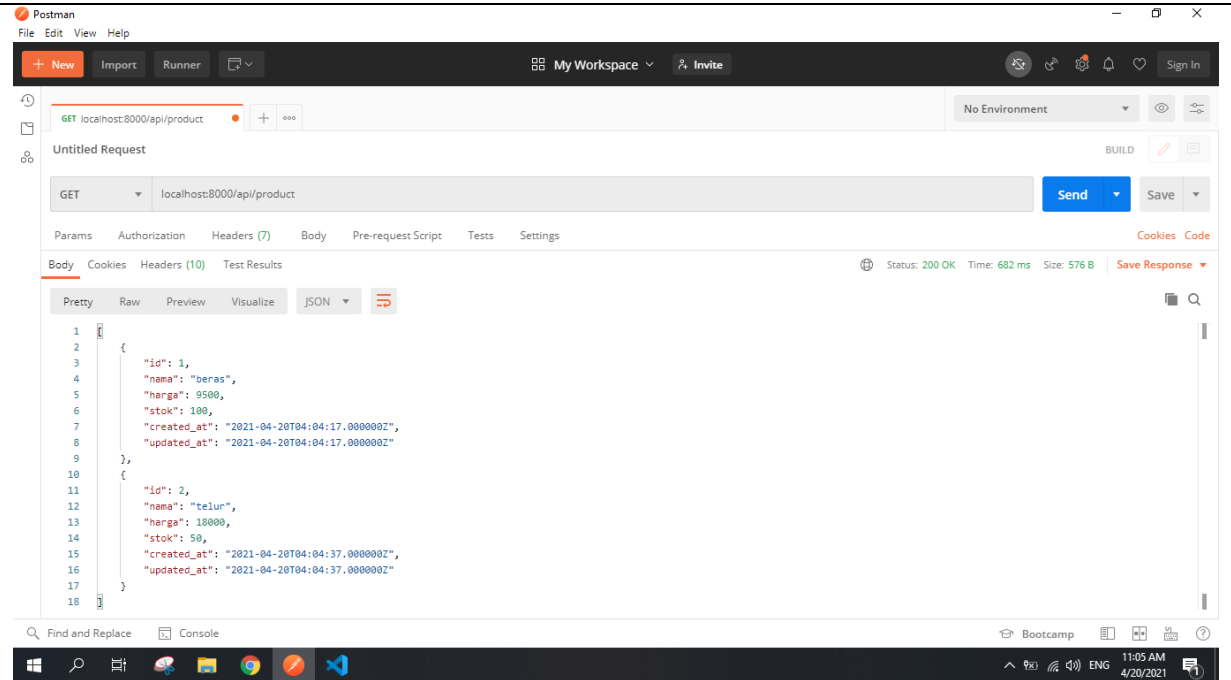
```
{
  "nama": "beras",
  "harga": "9500",
  "stok": "100",
  "updated_at": "2021-04-20T04:04:17.000000Z",
  "created_at": "2021-04-20T04:04:17.000000Z",
  "id": 1
}
```

The response is a JSON object with the following data:

```
{
  "nama": "beras",
  "harga": "9500",
  "stok": "100",
  "updated_at": "2021-04-20T04:04:17.000000Z",
  "created_at": "2021-04-20T04:04:17.000000Z",
  "id": 1
}
```

Status: 201 Created, Time: 697 ms, Size: 449 B

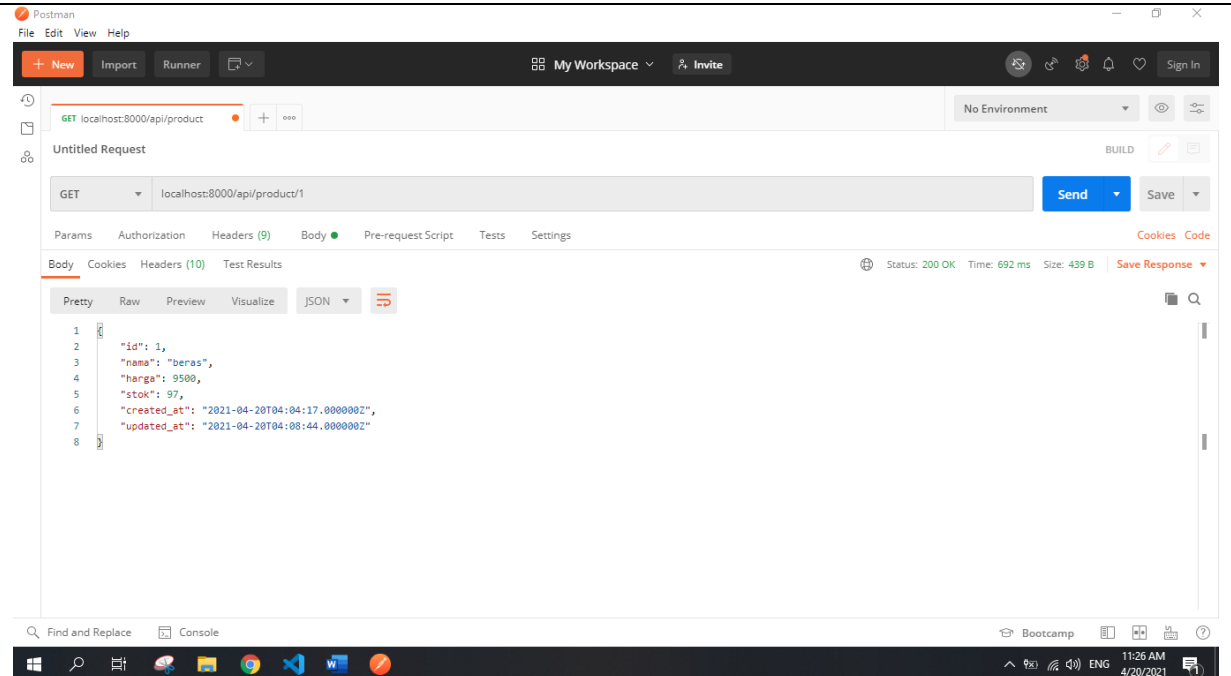
Melihat semua produk



Postman interface showing a GET request to `localhost:8000/api/product`. The response status is 200 OK, Time: 682 ms, Size: 576 B. The response body is a JSON array of two product objects.

```
1 {
2   "id": 1,
3   "nama": "beras",
4   "harga": 9500,
5   "stok": 100,
6   "created_at": "2021-04-20T04:04:17.000000Z",
7   "updated_at": "2021-04-20T04:04:17.000000Z"
8 },
9 {
10  "id": 2,
11  "nama": "telun",
12  "harga": 18000,
13  "stok": 50,
14  "created_at": "2021-04-20T04:04:37.000000Z",
15  "updated_at": "2021-04-20T04:04:37.000000Z"
16 }
17 }
18 }
```

Melihat satu produk



Postman interface showing a GET request to `localhost:8000/api/product/1`. The response status is 200 OK, Time: 692 ms, Size: 439 B. The response body is a JSON object representing a single product.

```
1 {
2   "id": 1,
3   "nama": "beras",
4   "harga": 9500,
5   "stok": 97,
6   "created_at": "2021-04-20T04:04:17.000000Z",
7   "updated_at": "2021-04-20T04:08:44.000000Z"
8 }
```

Mengedit produk

The screenshot shows the Postman application interface. At the top, the title bar says "Postman". Below it is a menu bar with "File", "Edit", and "View". The main toolbar includes buttons for "New", "Import", "Runner", and "My Workspace". The current request is a PUT request to "localhost:8000/api/product/2". The "Body" tab is selected, showing a JSON body with the following data:

```
{  "id": 2,  "nama": "telur",  "harga": "17500",  "stok": 50,  "created_at": "2021-04-20T04:04:37.000000Z",  "updated_at": "2021-04-20T04:06:00.000000Z"}
```

The status bar at the bottom indicates "Status: 200 OK", "Time: 699 ms", and "Size: 442 B".

Menghapus produk

The screenshot shows the Postman application interface. At the top, the title bar says "Postman". Below it is a menu bar with "File", "Edit", and "View". The main toolbar includes buttons for "New", "Import", "Runner", and "My Workspace". The current request is a DELETE request to "localhost:8000/api/product/3". The "Body" tab is selected, showing a JSON body with the following data:

```
{  "message": "Data berhasil dihapus"}
```

The status bar at the bottom indicates "Status: 200 OK", "Time: 686 ms", and "Size: 342 B".

Melakukan transaksi

The screenshot shows the Postman application with a POST request to `localhost:8000/api/transaksi`. The request body is a JSON object with the following structure:

```
1 {
2   "user_id": 1,
3   "products": [
4     {
5       "id": 1,
6       "total_item": 3
8     },
9     {
10      "id": 2,
11      "total_item": 2
12    }
13  ]
14 }
```

The response is a 200 OK status with a response time of 722 ms and a size of 413 B. The response body is a JSON object:

```
1 {
2   "message": "transaksi berhasil dilakukan",
3   "transaksi_id": 1,
4   "user_id": 1,
5   "total_item": 2,
6   "total_harga": 63500
7 }
```

Melihat semua transaksi yang telah dilakukan

The screenshot shows the Postman application with a GET request to `localhost:8000/api/transaksi`. The response is a 200 OK status with a response time of 712 ms and a size of 1.19 KB. The response body is a JSON object:

```
1 {
2   "id": 1,
3   "user_id": 1,
4   "total_item": 2,
5   "total_harga": 63500,
6   "created_at": "2021-04-20T04:08:44.000000Z",
7   "updated_at": "2021-04-20T04:08:44.000000Z",
8   "user": {
9     "id": 1,
10    "name": "kasir1",
11    "email": "kasir1@gmail.com",
12    "created_at": "2021-04-20T04:02:13.000000Z",
13    "updated_at": "2021-04-20T04:02:13.000000Z"
14  },
15  "details": [
16    {
17      "id": 1,
18    }
19  ]
20 }
```

Melihat satu transaksi yang telah dilakukan

The screenshot shows the Postman interface with a GET request to `localhost:8000/api/transaksi/1`. The response status is 200 OK, with a time of 682 ms and a size of 1.19 KB. The response body is displayed in JSON format:

```
1 {
2   "id": 1,
3   "user_id": 1,
4   "total_item": 2,
5   "total_harga": 63500,
6   "created_at": "2021-04-20T04:08:44.000000Z",
7   "updated_at": "2021-04-20T04:08:44.000000Z",
8   "user": {
9     "id": 1,
10    "name": "kasir1",
11    "email": "kasir1@gmail.com",
12    "created_at": "2021-04-20T04:02:13.000000Z",
13    "updated_at": "2021-04-20T04:02:13.000000Z"
14  },
15   "details": [
16     {
17       "id": 1,
18       "transaksi_id": 1,
```

Menghapus transaksi

The screenshot shows the Postman interface with a DELETE request to `localhost:8000/api/transaksi/1`. The response status is 200 OK, with a time of 689 ms and a size of 347 B. The response body is displayed in JSON format:

```
1 {
2   "message": "Transaksi berhasil dihapus"
3 }
```

Logout

Postman

File Edit View Help

New Import Runner

My Workspace

Invite

Sign In

POST localhost:8000/api/logout

No Environment

Build

Untitled Request

POST localhost:8000/api/logout

Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Cookies Code

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 707 ms Size: 340 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2  "message": "Logout successfully"
3
```

Find and Replace Console

Bootcamp

11:10 AM 4/20/2021