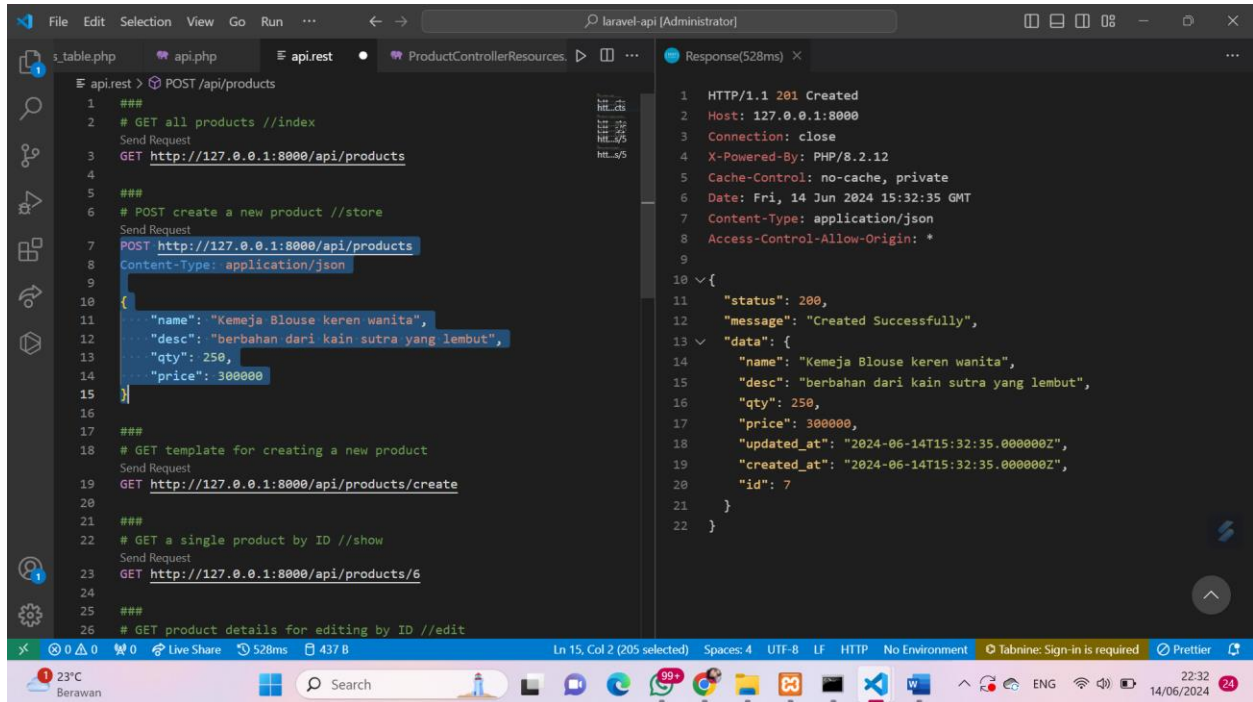


## Output Tugas Laravel-api (Menambahkan fungsi create, edit, update, dan delete)

### ➤ Menambahkan data database memakai fungsi store

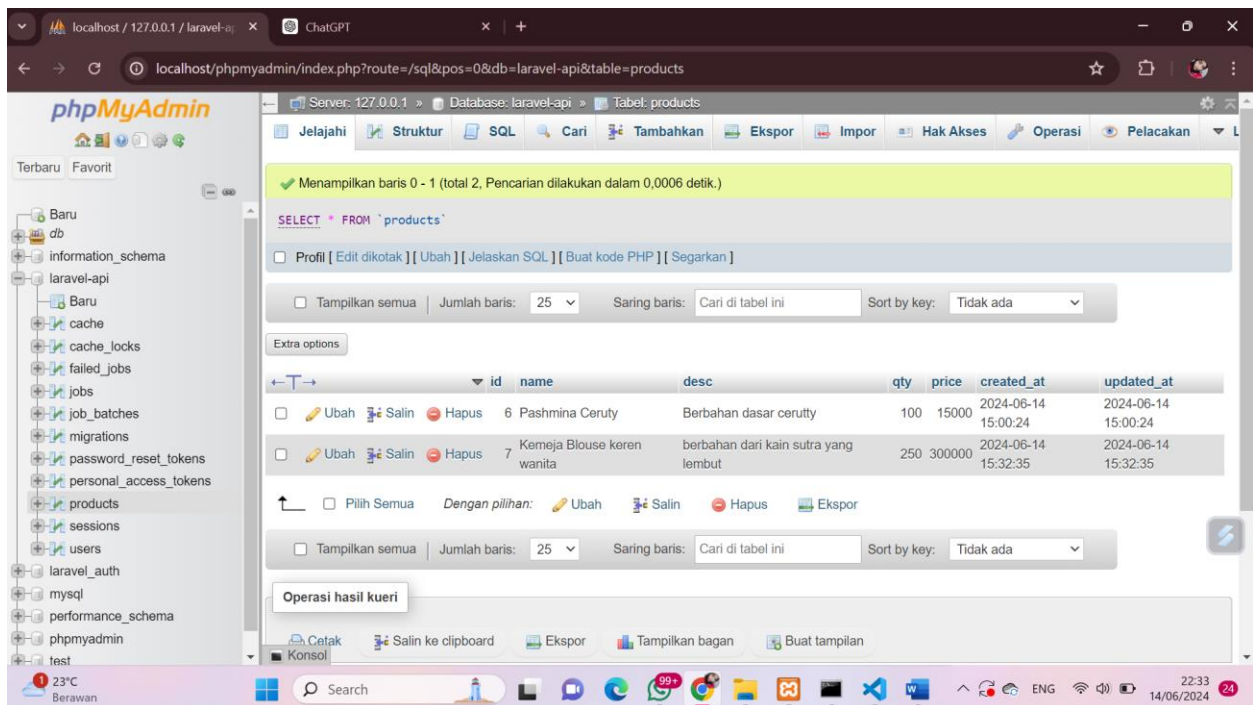


The screenshot shows a REST client interface in VS Code. The left pane displays a collection of API requests for a Laravel API. The selected request is a POST to `/api/products` with a JSON body. The right pane shows the response, which is a 200 status with a success message and product details.

```
###
1 # GET all products //index
2 Send Request
3 GET http://127.0.0.1:8000/api/products
4 ###
5 # POST create a new product //store
6 Send Request
7 POST http://127.0.0.1:8000/api/products
8 Content-Type: application/json
9 {
10   "name": "Kemeja Blouse keren wanita",
11   "desc": "berbahan dari kain sutra yang lembut",
12   "qty": 250,
13   "price": 300000
14 }
15 ###
16 # GET template for creating a new product
17 Send Request
18 GET http://127.0.0.1:8000/api/products/create
19 ###
20 # GET a single product by ID //show
21 Send Request
22 GET http://127.0.0.1:8000/api/products/6
23 ###
24 # GET product details for editing by ID //edit
```

```
1 HTTP/1.1 201 Created
2 Host: 127.0.0.1:8000
3 Connection: close
4 X-Powered-By: PHP/8.2.12
5 Cache-Control: no-cache, private
6 Date: Fri, 14 Jun 2024 15:32:35 GMT
7 Content-Type: application/json
8 Access-Control-Allow-Origin: *
9
10 {
11   "status": 200,
12   "message": "Created Successfully",
13   "data": {
14     "name": "Kemeja Blouse keren wanita",
15     "desc": "berbahan dari kain sutra yang lembut",
16     "qty": 250,
17     "price": 300000,
18     "updated_at": "2024-06-14T15:32:35.000000Z",
19     "created_at": "2024-06-14T15:32:35.000000Z",
20     "id": 7
21   }
22 }
```

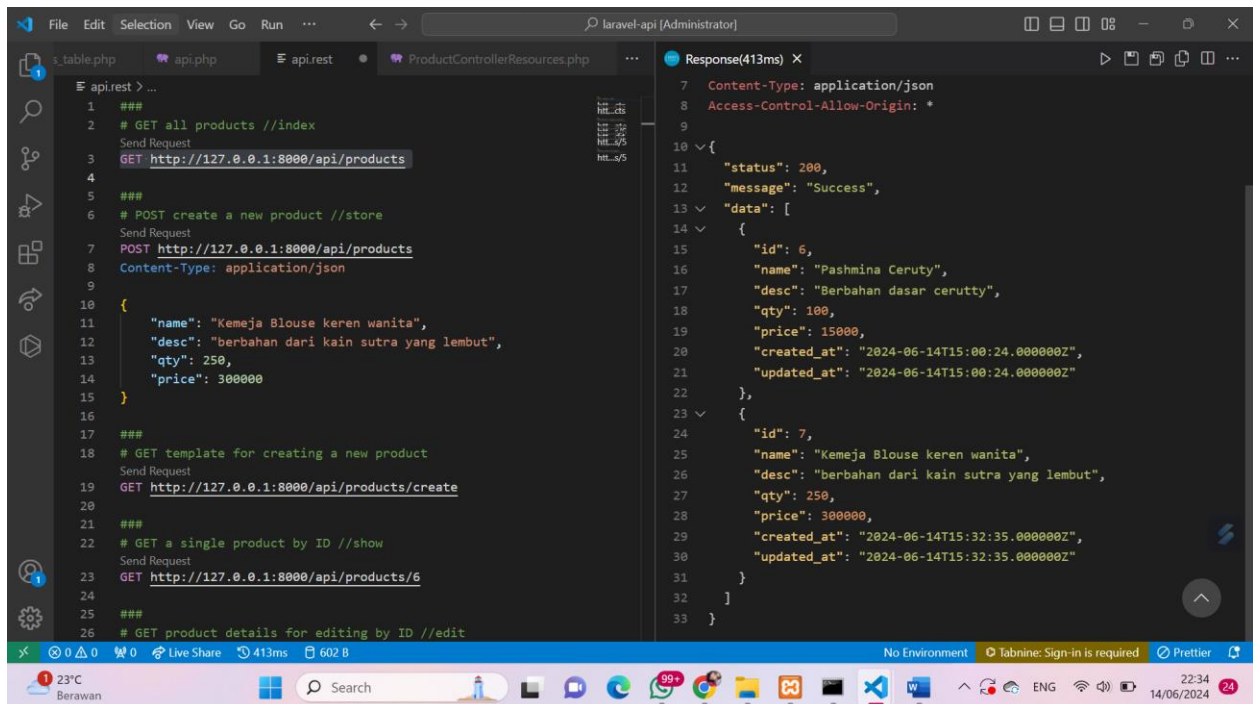
### Berhasil menambahkan ke database



The screenshot shows the phpMyAdmin interface for the `laravel-api` database. The `products` table is selected, and the data is displayed in a table format. The table contains two rows of product data.

	id	name	desc	qty	price	created_at	updated_at
<input type="checkbox"/>	6	Pashmina Ceruty	Berbahan dasar ceruty	100	15000	2024-06-14 15:00:24	2024-06-14 15:00:24
<input type="checkbox"/>	7	Kemeja Blouse keren wanita	berbahan dari kain sutra yang lembut	250	300000	2024-06-14 15:32:35	2024-06-14 15:32:35

## ➤ Menampilkan semua data database memakai fungsi index



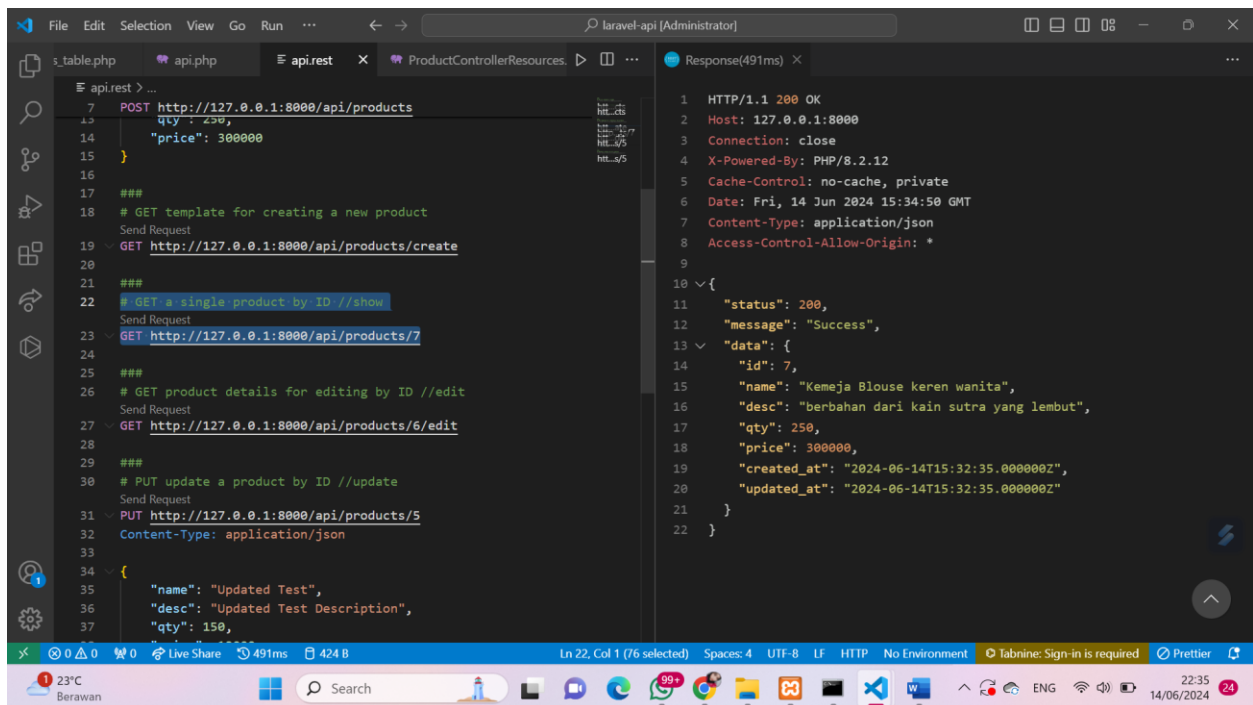
The screenshot shows a REST client interface in VS Code. The left pane contains a REST client file named `api.rest` with the following content:

```
1 ###
2 # GET all products //index
3 GET http://127.0.0.1:8000/api/products
4
5 ###
6 # POST create a new product //store
7 POST http://127.0.0.1:8000/api/products
8 Content-Type: application/json
9
10 {
11   "name": "Kemeja Blouse keren wanita",
12   "desc": "berbahan dari kain sutra yang lembut",
13   "qty": 250,
14   "price": 300000
15 }
16
17 ###
18 # GET template for creating a new product
19 GET http://127.0.0.1:8000/api/products/create
20
21 ###
22 # GET a single product by ID //show
23 GET http://127.0.0.1:8000/api/products/6
24
25 ###
26 # GET product details for editing by ID //edit
```

The right pane shows the response for the GET request to `http://127.0.0.1:8000/api/products`. The response is a JSON array with two product objects:

```
7 Content-Type: application/json
8 Access-Control-Allow-Origin: *
9
10 [
11   {
12     "status": 200,
13     "message": "Success",
14     "data": [
15       {
16         "id": 6,
17         "name": "Pashmina Ceruty",
18         "desc": "Berbahan dasar ceruty",
19         "qty": 100,
20         "price": 15000,
21         "created_at": "2024-06-14T15:00:24.000000Z",
22         "updated_at": "2024-06-14T15:00:24.000000Z"
23       },
24       {
25         "id": 7,
26         "name": "Kemeja Blouse keren wanita",
27         "desc": "berbahan dari kain sutra yang lembut",
28         "qty": 250,
29         "price": 300000,
30         "created_at": "2024-06-14T15:32:35.000000Z",
31         "updated_at": "2024-06-14T15:32:35.000000Z"
32       }
33     ]
34   }
35 ]
```

## ➤ Menampilkan data persatuan dari database menggunakan fungsi show



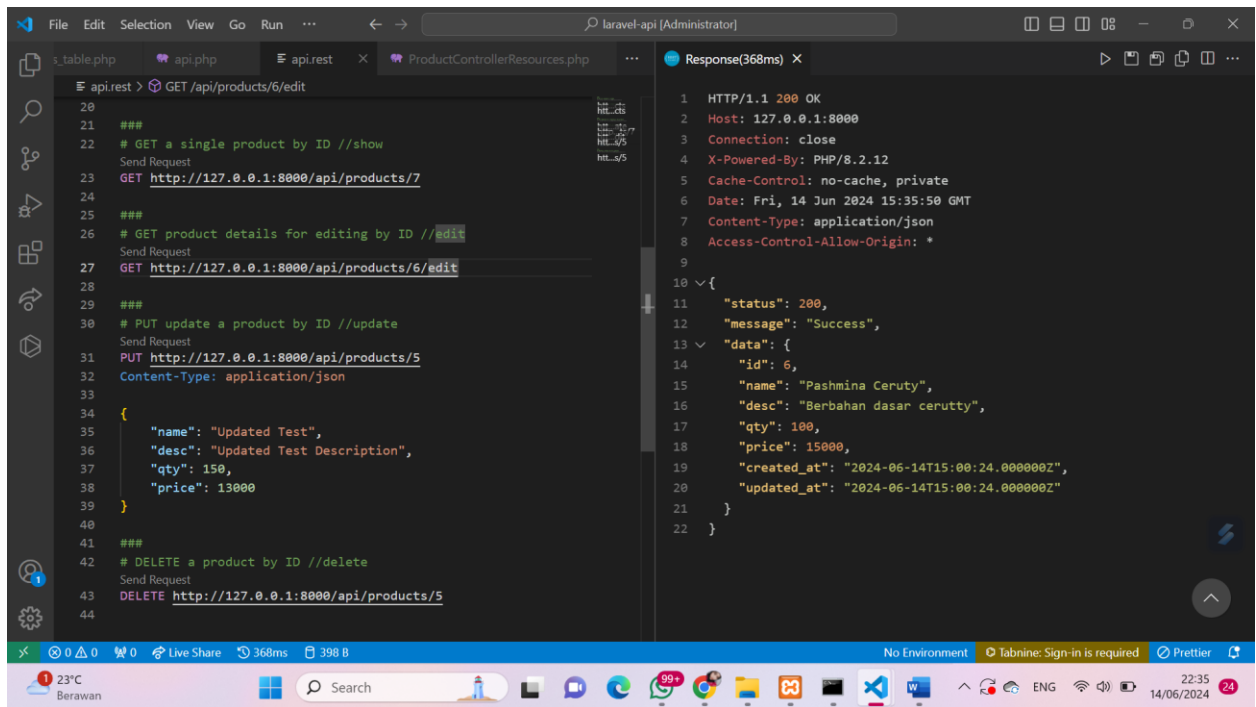
The screenshot shows a REST client interface in VS Code. The left pane contains a REST client file named `api.rest` with the following content:

```
7 POST http://127.0.0.1:8000/api/products
8 {
9   "name": "Kemeja Blouse keren wanita",
10  "desc": "berbahan dari kain sutra yang lembut",
11  "qty": 250,
12  "price": 300000
13 }
14
15 ###
16 # GET template for creating a new product
17 GET http://127.0.0.1:8000/api/products/create
18
19 ###
20 # GET a single product by ID //show
21 GET http://127.0.0.1:8000/api/products/7
22
23 ###
24 # GET product details for editing by ID //edit
25 GET http://127.0.0.1:8000/api/products/6/edit
26
27 ###
28 # PUT update a product by ID //update
29 PUT http://127.0.0.1:8000/api/products/5
30 Content-Type: application/json
31 {
32   "name": "Updated Test",
33   "desc": "Updated Test Description",
34   "qty": 150,
35   "price": 300000
36 }
37
```

The right pane shows the response for the GET request to `http://127.0.0.1:8000/api/products/7`. The response is a JSON object representing a single product:

```
1 HTTP/1.1 200 OK
2 Host: 127.0.0.1:8000
3 Connection: close
4 X-Powered-By: PHP/8.2.12
5 Cache-Control: no-cache, private
6 Date: Fri, 14 Jun 2024 15:34:50 GMT
7 Content-Type: application/json
8 Access-Control-Allow-Origin: *
9
10 {
11   "status": 200,
12   "message": "Success",
13   "data": {
14     "id": 7,
15     "name": "Kemeja Blouse keren wanita",
16     "desc": "berbahan dari kain sutra yang lembut",
17     "qty": 250,
18     "price": 300000,
19     "created_at": "2024-06-14T15:32:35.000000Z",
20     "updated_at": "2024-06-14T15:32:35.000000Z"
21   }
22 }
```

## ➤ Mengedit data



The screenshot shows a VS Code editor with a REST client file named `api.rest`. The file contains several API requests. The selected request is a GET request to `http://127.0.0.1:8000/api/products/6/edit`. The response is a 200 OK status with a JSON body containing product details.

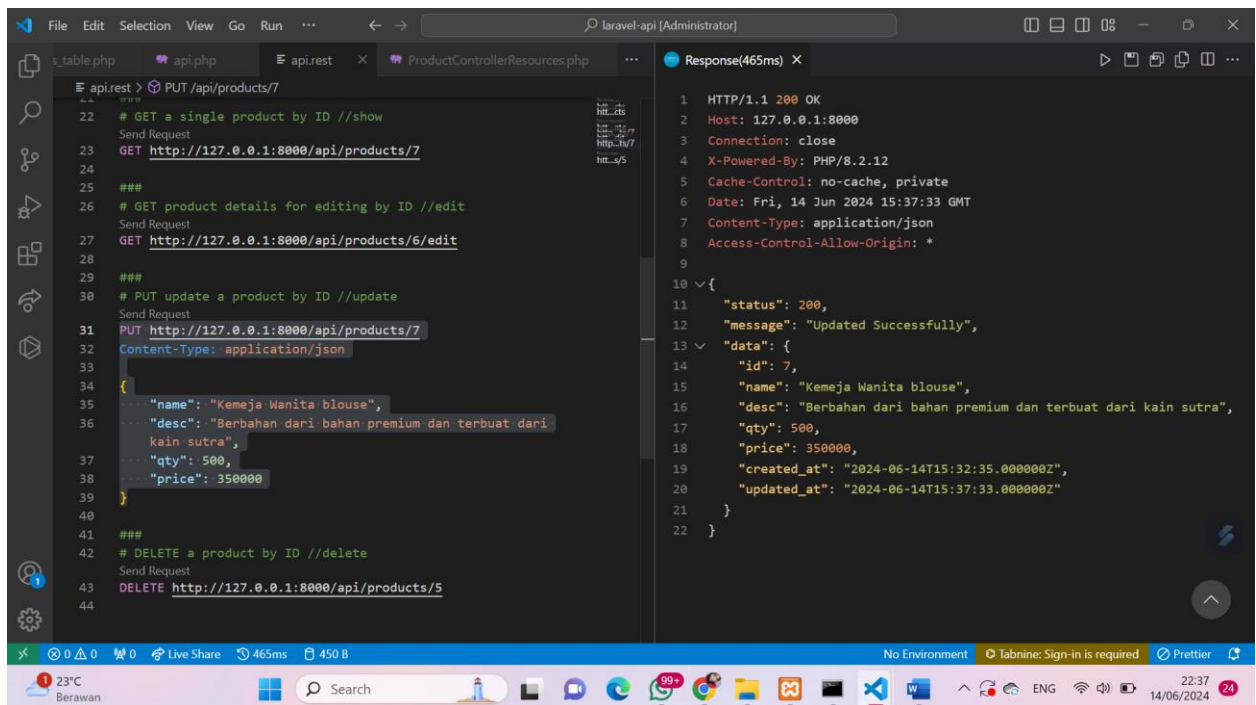
```
20
21
22 # GET a single product by ID //show
23 GET http://127.0.0.1:8000/api/products/7
24
25 ###
26 # GET product details for editing by ID //edit
27 GET http://127.0.0.1:8000/api/products/6/edit
28
29 ###
30 # PUT update a product by ID //update
31 PUT http://127.0.0.1:8000/api/products/5
32 Content-Type: application/json
33 {
34   "name": "Updated Test",
35   "desc": "Updated Test Description",
36   "qty": 150,
37   "price": 13000
38 }
39
40 ###
41 # DELETE a product by ID //delete
42 DELETE http://127.0.0.1:8000/api/products/5
```

The response (368ms) is as follows:

```
1 HTTP/1.1 200 OK
2 Host: 127.0.0.1:8000
3 Connection: close
4 X-Powered-By: PHP/8.2.12
5 Cache-Control: no-cache, private
6 Date: Fri, 14 Jun 2024 15:35:50 GMT
7 Content-Type: application/json
8 Access-Control-Allow-Origin: *
```

```
10 {
11   "status": 200,
12   "message": "Success",
13   "data": {
14     "id": 6,
15     "name": "Pashmina Ceruty",
16     "desc": "Berbahan dasar ceruty",
17     "qty": 100,
18     "price": 15000,
19     "created_at": "2024-06-14T15:00:24.000000Z",
20     "updated_at": "2024-06-14T15:00:24.000000Z"
21   }
22 }
```

## ➤ Mengupdate data database menggunakan fungsi update



The screenshot shows a VS Code editor with a REST client file named `api.rest`. The selected request is a PUT request to `http://127.0.0.1:8000/api/products/7`. The response is a 200 OK status with a JSON body indicating a successful update.

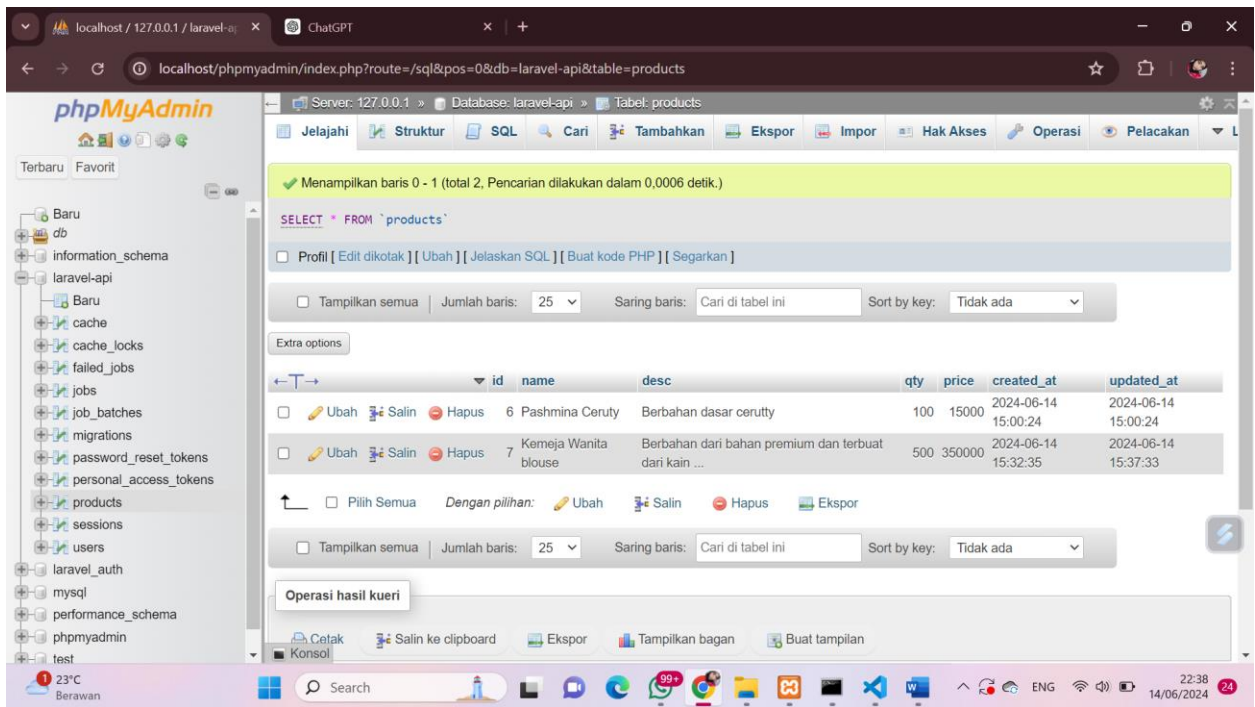
```
22 # GET a single product by ID //show
23 GET http://127.0.0.1:8000/api/products/7
24
25 ###
26 # GET product details for editing by ID //edit
27 GET http://127.0.0.1:8000/api/products/6/edit
28
29 ###
30 # PUT update a product by ID //update
31 PUT http://127.0.0.1:8000/api/products/7
32 Content-Type: application/json
33 {
34   "name": "Kemeja Wanita blouse",
35   "desc": "Berbahan dari bahan premium dan terbuat dari kain sutra",
36   "qty": 500,
37   "price": 35000
38 }
39
40 ###
41 # DELETE a product by ID //delete
42 DELETE http://127.0.0.1:8000/api/products/5
```

The response (465ms) is as follows:

```
1 HTTP/1.1 200 OK
2 Host: 127.0.0.1:8000
3 Connection: close
4 X-Powered-By: PHP/8.2.12
5 Cache-Control: no-cache, private
6 Date: Fri, 14 Jun 2024 15:37:33 GMT
7 Content-Type: application/json
8 Access-Control-Allow-Origin: *
```

```
10 {
11   "status": 200,
12   "message": "Updated Successfully",
13   "data": {
14     "id": 7,
15     "name": "Kemeja Wanita blouse",
16     "desc": "Berbahan dari bahan premium dan terbuat dari kain sutra",
17     "qty": 500,
18     "price": 35000,
19     "created_at": "2024-06-14T15:32:35.000000Z",
20     "updated_at": "2024-06-14T15:37:33.000000Z"
21   }
22 }
```

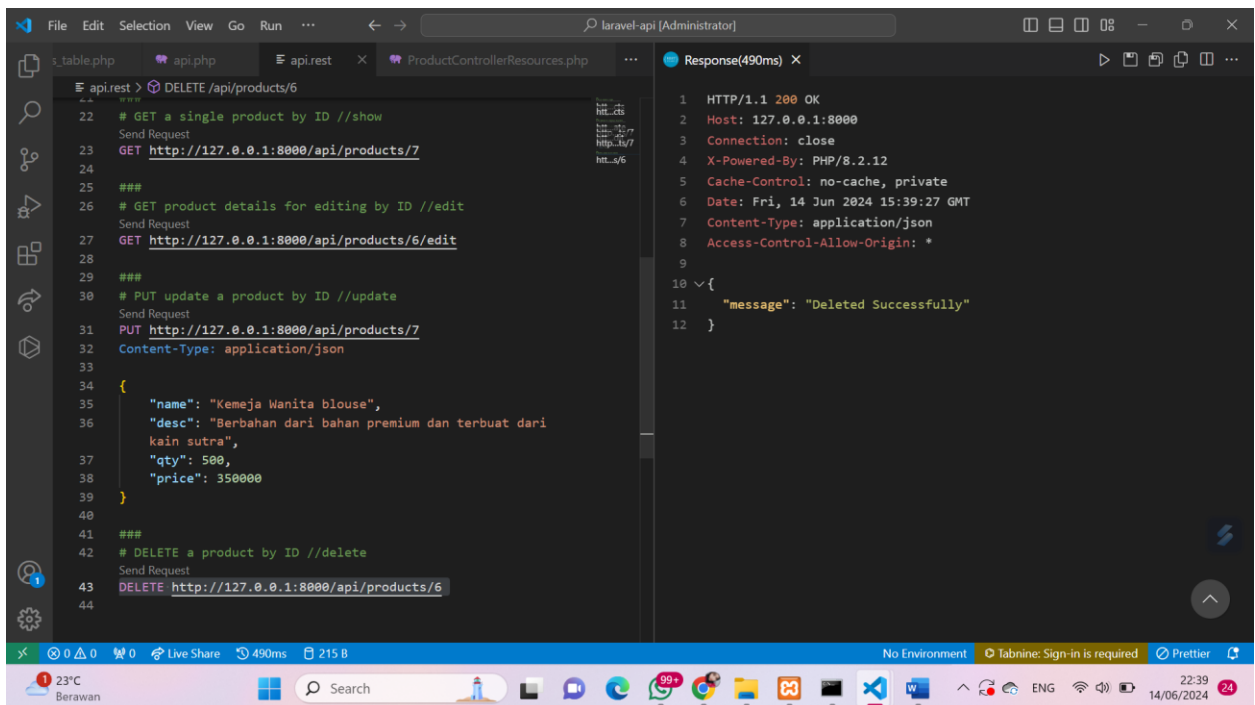
## Berhasil terupdate data id=7



The screenshot shows the phpMyAdmin interface for the 'laravel-api' database. The 'products' table is selected, and the SQL query 'SELECT \* FROM `products`' is executed. The table contains two rows:

	id	name	desc	qty	price	created_at	updated_at
<input type="checkbox"/>	6	Pashmina Ceruty	Berbahan dasar cerutty	100	15000	2024-06-14 15:00:24	2024-06-14 15:00:24
<input type="checkbox"/>	7	Kemeja Wanita blouse	Berbahan dari bahan premium dan terbuat dari kain ...	500	350000	2024-06-14 15:32:35	2024-06-14 15:37:33

- Menghapus data database menggunakan fungsi delete (disini saya menghapus data pada id=6)



The screenshot shows the VS Code editor with the REST Client and API Explorer. The REST Client is open, showing a DELETE request to 'http://127.0.0.1:8000/api/products/6'. The API Explorer shows the response for the DELETE request:

```
DELETE http://127.0.0.1:8000/api/products/6
```

```
1 HTTP/1.1 200 OK
2 Host: 127.0.0.1:8000
3 Connection: close
4 X-Powered-By: PHP/8.2.12
5 Cache-Control: no-cache, private
6 Date: Fri, 14 Jun 2024 15:39:27 GMT
7 Content-Type: application/json
8 Access-Control-Allow-Origin: *
9
10 {
11   "message": "Deleted Successfully"
12 }
```

## Berhasil terhapus data id=6 di database

The screenshot shows the phpMyAdmin web interface in a browser. The URL is `localhost/phpmyadmin/index.php?route=/sql&pos=0&db=laravel-api&table=products`. The interface displays the 'products' table with the following data:

id	name	desc	qty	price	created_at	updated_at
7	Kemeja Wanita blouse	Berbahan dari bahan premium dan terbuat dari kain ...	500	350000	2024-06-14 15:32:35	2024-06-14 15:37:33

The interface also shows a SQL query editor with the query `SELECT * FROM `products`` and a console window at the bottom.