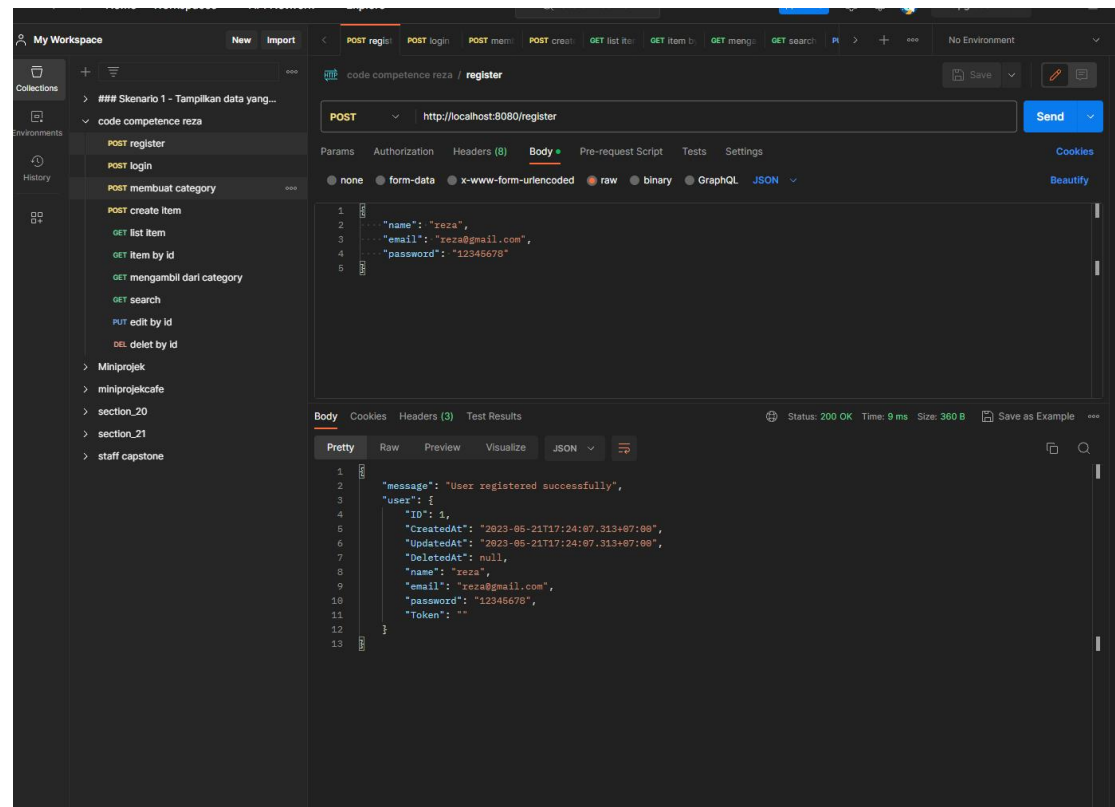


Code competence

1.Register

http://localhost:8080/register

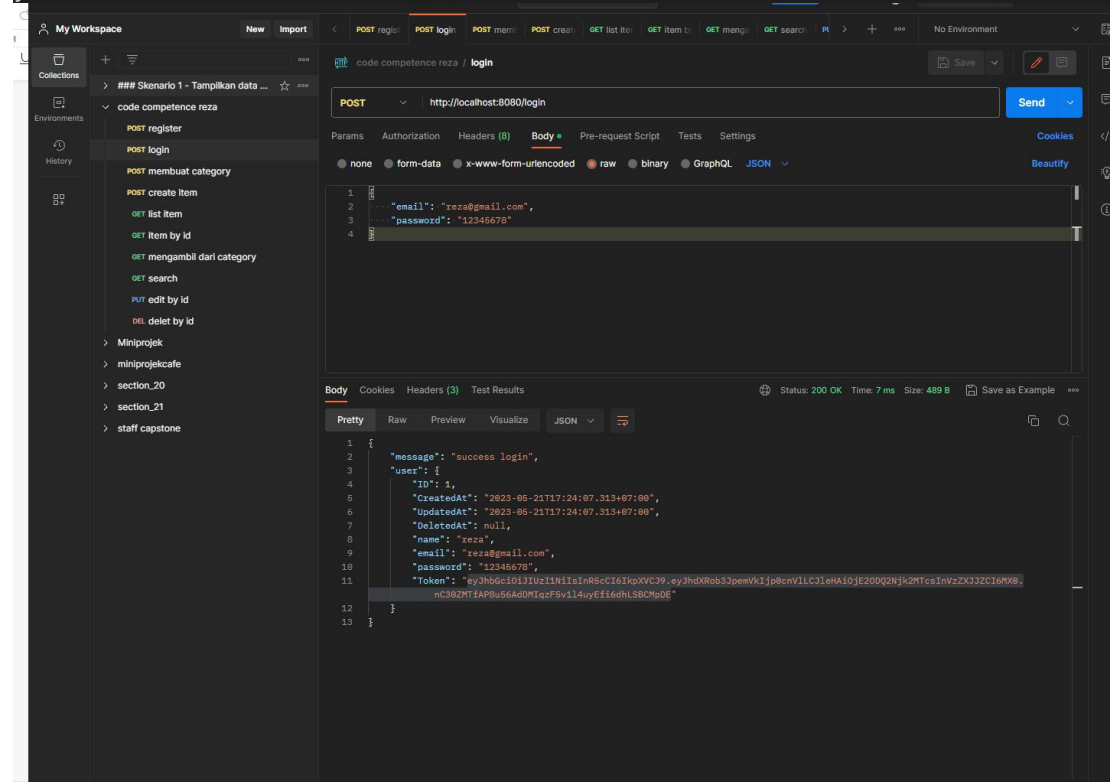
```
{
  "name": "reza",
  "email": "reza@gmail.com",
  "password": "12345678"
}
```



2.Login

http://localhost:8080/login

```
{
  "email": "reza@gmail.com",
  "password": "12345678"
}
```



3. Create Category

```
http://localhost:8080/items/category
```

```
{  
  "name": "laptop"  
}
```

The screenshot displays the Postman application interface. On the left, the 'My Workspace' sidebar shows a collection named 'code competence reza' with a sub-collection 'membuat category'. The main panel shows a POST request to 'http://localhost:8080/items/category'. The request body is a JSON object: `{ "name": "laptop" }`. The response is shown in the 'Body' tab, indicating a successful status (200 OK) with a response time of 4 ms and a size of 197 B. The response body is a JSON object: `{ "category": { "id": 1, "name": "laptop" }, "message": "Success Create Category" }`.

POST http://localhost:8080/items/category

Body

```
{  
  "name": "laptop"  
}
```

Body

```
{  
  "category": {  
    "id": 1,  
    "name": "laptop"  
  },  
  "message": "Success Create Category"  
}
```

Status: 200 OK Time: 4 ms Size: 197 B

4. Create Item

```
http://localhost:8080/items
{
  "name": "asus tuff a15",
  "description": "laptop murah ryzen 7",
  "stock": 1,
  "price": 500000,
  "category_id": 1
}
```

The screenshot displays a REST client interface with a sidebar on the left containing a 'My Workspace' section with various API endpoints. The main panel shows a POST request to 'http://localhost:8080/items' with a JSON body. The 'Body' tab is active, showing the request body and the response body. The response status is '201 Created' with a time of '7 ms' and a size of '317 B'. The response body is a JSON object containing an 'items' array with one object representing the created item.

Request Body:

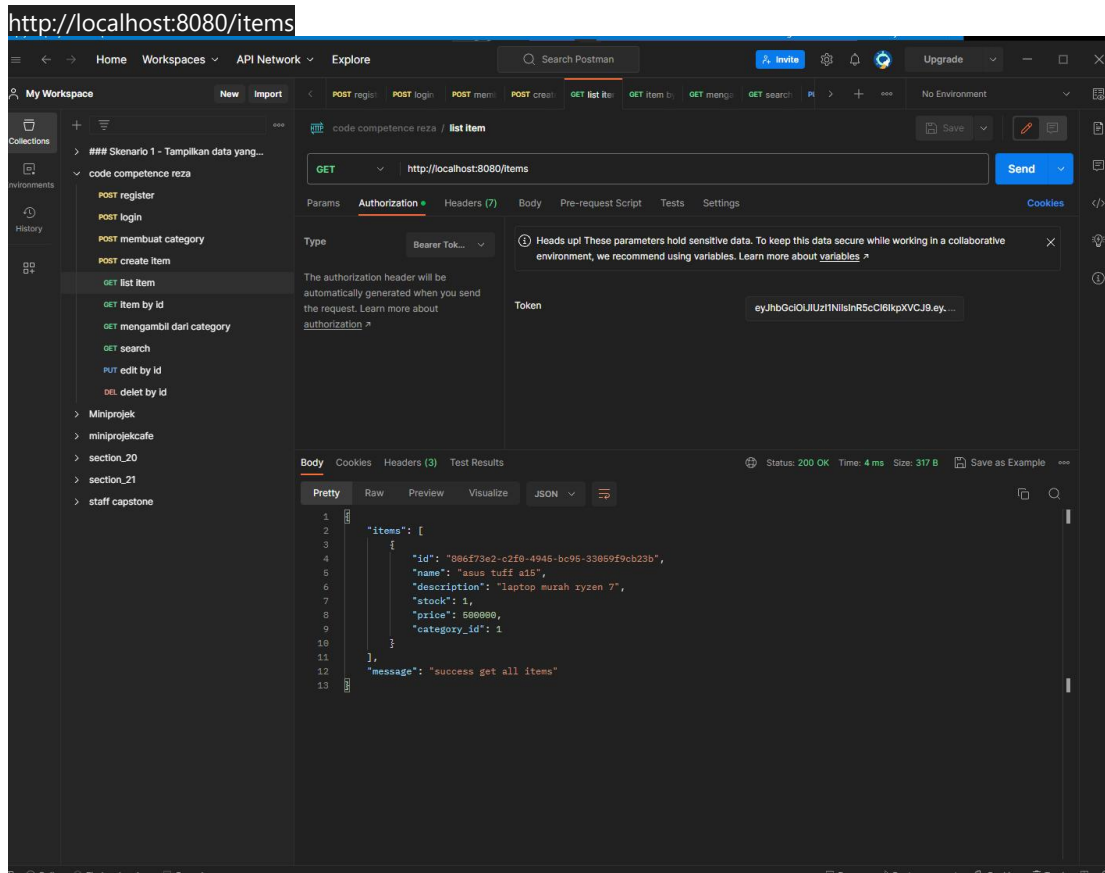
```
{
  "name": "asus tuff a15",
  "description": "laptop murah ryzen 7",
  "stock": 1,
  "price": 500000,
  "category_id": 1
}
```

Response Body:

```
{
  "items": [
    {
      "id": "986f73e2-c2f8-4946-bc95-33859f9cb23b",
      "name": "asus tuff a15",
      "description": "laptop murah ryzen 7",
      "stock": 1,
      "price": 500000,
      "category_id": 1
    }
  ],
  "message": "Success Create Item"
}
```

5. List item

http://localhost:8080/items



The screenshot shows the Postman interface with a GET request to `http://localhost:8080/items`. The request is configured with Bearer Token authentication. The response status is 200 OK, and the response body is a JSON array of items.

Request:

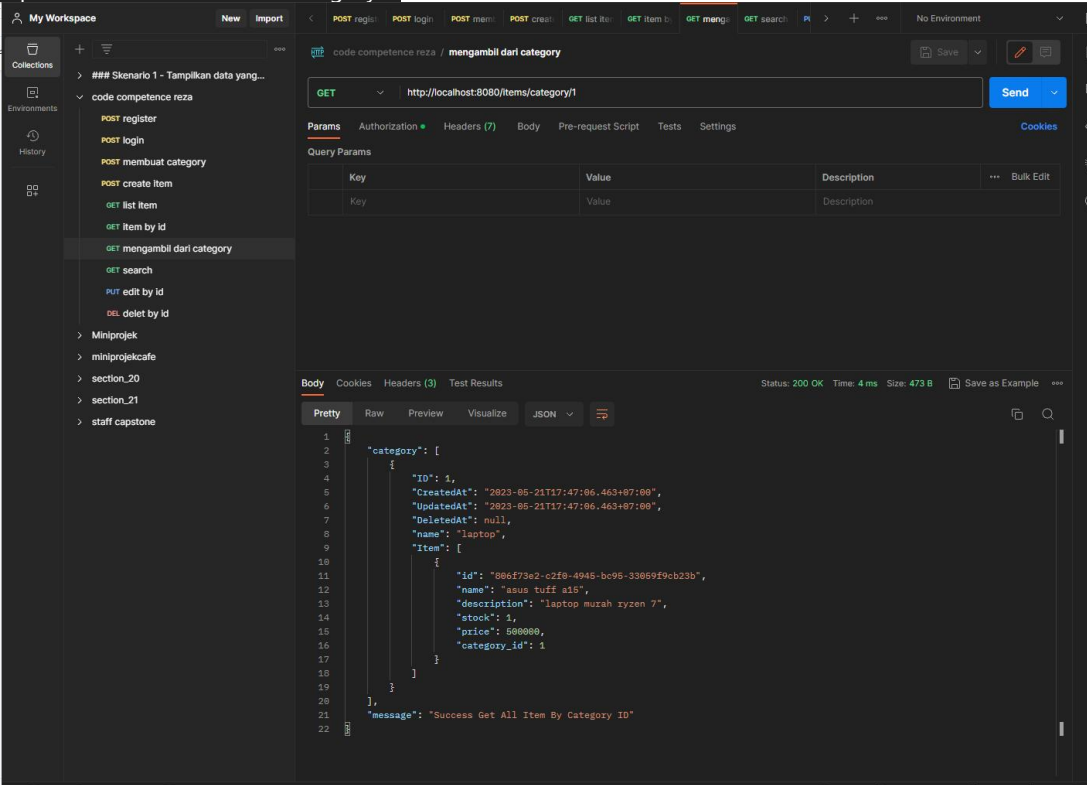
- Method: GET
- URL: `http://localhost:8080/items`
- Authorization: Bearer Token (Token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`)

Response:

```
1  {
2    "items": [
3      {
4        "id": "886f73e2-c2fe-4945-bc95-33869f9cb23b",
5        "name": "asus tuff a15",
6        "description": "laptop murah zyzen 7",
7        "stock": 1,
8        "price": 8888888,
9        "category_id": 1
10     }
11   ],
12   "message": "success get all items"
13 }
```

6. List category

http://localhost:8080/items/category/1



The screenshot displays a REST client interface with a sidebar on the left containing a 'Collections' pane and an 'Environments' pane. The 'Collections' pane shows a collection named 'code competence reza' with several endpoints, including 'GET mengambil dari category'. The 'Environments' pane shows a collection named 'code competence reza' with several endpoints, including 'GET mengambil dari category'. The main area shows a GET request to 'http://localhost:8080/items/category/1'. The response is a JSON object with a status of 200 OK, time of 4 ms, and size of 473 B. The response body is displayed in a 'Pretty' format, showing a list of items under the category 'laptop'.

```
1 {
2   "category": [
3     {
4       "ID": 1,
5       "CreatedAt": "2023-05-21T17:47:06.463+07:00",
6       "UpdatedAt": "2023-05-21T17:47:06.463+07:00",
7       "DeletedAt": null,
8       "name": "laptop",
9       "Item": [
10        {
11          "id": "86ef73e2-c2f8-4945-bc95-33659f9cb23b",
12          "name": "masu tuff a15",
13          "description": "laptop murah ryzen 7",
14          "stock": 1,
15          "price": 5000000,
16          "category_id": 1
17        }
18      ]
19    }
20  ],
21  "message": "Success Get All Item By Category ID"
22 }
```

7. search

http://localhost:8080/items/name?name=asus

The screenshot shows a REST client interface with a sidebar on the left containing a list of collections and environments. The main area displays a GET request to `http://localhost:8080/items/name?name=asus`. The request parameters are defined in a table:

Key	Value	Description
<input checked="" type="checkbox"/> name	asus	

The response body is shown in JSON format:

```
1 {
2   "item": [
3     {
4       "id": "886f73e2-c2f8-4946-bc98-33859f9cb23b",
5       "name": "asus tuff a15",
6       "description": "laptop murah ryzen 7",
7       "stock": 1,
8       "price": 500000,
9       "category_id": 1
10    }
11  ],
12  "message": "Success Get Items By Name"
13 }
```

The status bar at the bottom indicates a 201 Created status, a response time of 3 ms, and a size of 325 B.

8. Edit item

http://localhost:8080/items/id/806f73e2-c2f0-4945-bc95-33059f9cb23b

The screenshot shows a REST client interface with a sidebar on the left containing a list of API endpoints. The main panel displays a PUT request to the URL `http://localhost:8080/items/id/806f73e2-c2f0-4945-bc95-33059f9cb23b`. The request body is a JSON object representing an item to be updated.

Query Params

Key	Value	Description
Key	Value	Description

Body

```
1  {
2    "Item": {
3      "id": "806f73e2-c2f0-4945-bc95-33059f9cb23b",
4      "name": "laptop tuff",
5      "description": "laptop murmer",
6      "stock": 20,
7      "price": 120000,
8      "category_id": 1
9    },
10   "message": "Success Update Item By ID"
11 }
```

Status: 200 OK Time: 6 ms Size: 310 B

9. Delete Item

http://localhost:8080/items/id/806f73e2-c2f0-4945-bc95-33059f9cb23b

The screenshot shows a REST client interface with a workspace named "code competence reza". The selected endpoint is "delete by id" with a URL of "http://localhost:8080/items/id/806f73e2-c2f0-4945-bc95-33059f9cb23b". The request method is "DELETE". The response status is "200 OK" with a time of "3 ms" and a size of "157 B". The response body is a JSON object: {"message": "Success Delete Item"}.

My Workspace New Import < POST memi POST creat GET list ite GET item b GET mangi GET search PUT edit by DEL delet b > + ... No Environment

code competence reza / delete by id Save Send

DELETE http://localhost:8080/items/id/806f73e2-c2f0-4945-bc95-33059f9cb23b

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (3) Test Results Status: 200 OK Time: 3 ms Size: 157 B Save as Example

Pretty Raw Preview Visualize JSON ...

```
1  [
2    "message": "Success Delete Item"
3  ]
```

Online Find and replace Console Runner Capture requests Cookies Trash