

## ACTIVITY PERTEMUAN 5

NAMA : Mikail Thoriq Kariemshah Banowo

NPM : 50422881

KELAS : 1IA19

MATERI : CRUD

MATA PRAKTIKUM : AP2B

---

```
app.py | data.json | PROBLEMS | TERMINAL | DEBUG CONSOLE | py + - | ... | X
```

```
app.py > save_data
11 from flask import Flask, request, jsonify
12 import json
13
14 app = Flask(__name__)
15
16 DATA_FILE = 'data.json'
17
18 def load_data():
19     with open(DATA_FILE, 'r') as f:
20         return json.load(f)
21
22 def save_data(data):
23     with open(DATA_FILE, 'w') as f:
24         json.dump(data, f, indent=4)
25
26 @app.route('/')
27 def home():
28     return "API Ready"
29
30 @app.route('/stores', methods=['GET'])
31 def get_stores():
32     stores = load_data()
33
34     return jsonify(stores)
35
36 @app.route('/stores/<int:store_id>', methods=['GET'])
37 def get_store(store_id):
38     stores = load_data()
39
40     for store in stores:
41         if store['id'] == store_id:
42             return jsonify(store)
43
44     return jsonify({'error': 'Store not found'}), 404
45
46 @app.route('/stores', methods=['POST'])
47 def add_store():
48     stores = load_data()
49     data = request.get_json()
50
51     required_keys = {'name', 'address', 'phone'}
52
53     if not required_keys.issubset(data.keys()):
54         missing_keys = required_keys - data.keys()
55         return jsonify({'error': f'Missing required keys: {missing_keys}'}), 400
56
57     new_store = {
58         'id': max(store['id'] for store in stores) + 1,
59         'name': data['name'],
60         'address': data['address'],
61         'phone': data['phone']
62     }
63
64     stores.append(new_store)
65     save_data(stores)
66
67     return jsonify(new_store), 201
68
69 @app.route('/stores/<int:store_id>', methods=['PUT'])
70 def update_store(store_id):
71     stores = load_data()
72
73     for store in stores:
74         if store['id'] == store_id:
75             data = request.get_json()
76             store['name'] = data['name']
77             store['address'] = data['address']
78             store['phone'] = data['phone']
79
80     save_data(stores)
81
82     return jsonify(store), 200
83
84 @app.route('/stores/<int:store_id>', methods=['DELETE'])
85 def delete_store(store_id):
86     stores = load_data()
87
88     for store in stores:
89         if store['id'] == store_id:
90             stores.remove(store)
91
92     save_data(stores)
93
94     return jsonify({'message': 'Store deleted'}), 200
```

```
~\Documents\Programming\praktikum-ti\Pertemuan5 v3.11.2
14:26 py app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 246-769-587
```

```
app.py > save_data
13
14 @app.route('/stores/<int:store_id>', methods=['GET'])
15 def get_store(store_id):
16     stores = load_data()
17
18     for store in stores:
19         if store['id'] == store_id:
20             return jsonify(store)
21
22     return jsonify({'error': 'Store not found'}), 404
23
24 @app.route('/stores', methods=['POST'])
25 def add_store():
26     stores = load_data()
27     data = request.get_json()
28
29     required_keys = {'name', 'address', 'phone'}
30
31     if not required_keys.issubset(data.keys()):
32         missing_keys = required_keys - data.keys()
33         return jsonify({'error': f'Missing required keys: {missing_keys}'}), 400
34
35     new_store = {
36         'id': max(store['id'] for store in stores) + 1,
37         'name': data['name'],
38         'address': data['address'],
39         'phone': data['phone']
40     }
41
42     stores.append(new_store)
43     save_data(stores)
44
45     return jsonify(new_store), 201
46
47 @app.route('/stores/<int:store_id>', methods=['DELETE'])
48 def delete_store(store_id):
49     stores = load_data()
50
51     for store in stores:
52         if store['id'] == store_id:
53             stores.remove(store)
54
55     save_data(stores)
56
57     return jsonify({'message': 'Store deleted'}), 200
```

```
~\Documents\Programming\praktikum-ti\Pertemuan5 v3.11.2
14:26 py app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 246-769-587
```

```
app.py
31 def save_data:
32     if not required_keys.issubset(data.keys()):
33         missing_keys = required_keys - data.keys()
34         return jsonify({'error': f'Missing required keys: {missing_keys}'})
35
36     new_store = {
37         'id': max(store['id'] for store in stores) + 1,
38         'name': data['name'],
39         'address': data['address'],
40         'phone': data['phone'],
41     }
42
43     stores.append(new_store)
44     save_data(stores)
45
46     return jsonify(new_store), 201
47
48 if __name__ == '__main__':
49     app.run(debug = True)
```

```
data.json
1 [{"id": 4,
2   "name": "Toko Tanaman",
3   "address": "Solo",
4   "phone": "0812912930123"},
5
6  {"id": 5,
7   "name": "Toko Kayu",
8   "address": "Banyuwangi",
9   "phone": "0812912930123"},
10
11 {"id": 6,
12  "name": "Toko Batu",
13  "address": "Jakarta",
14  "phone": "0812912930123"},
15
16 {"id": 7,
17  "name": "Toko Mainan",
18  "address": "Depok",
19  "phone": "0128812931"},
20
21 {"id": 8,
22  "name": "Toko Bukunq"}]
```

```
Terminal
~\Documents\Programming\praktikum-ti\Pertemuan5 v3.11.2
14:26 py app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 246-769-587
```

## Get All Stores

Postman interface showing a GET request to `http://localhost:5000/stores`. The response is a JSON array of store data.

Key	Value	Description
Key	Value	Description

```
Body
Pretty
1 {
2   "address": "Solo",
3   "id": 4,
4   "name": "Toko Tanaman",
5   "phone": "0812912930123"
6 },
7 {
8   "address": "Banyuwangi",
9   "id": 5,
10  "name": "Toko Kayu",
11  "phone": "0812912930123"
12 },
13 }
```

## Show Specific Store

The screenshot shows the Postman interface with a workspace named 'Pertemuan5'. The active collection is 'Pertemuan5', and the selected request is 'Show Specific Store'. The request is a GET method to the endpoint `{{url}}/stores/5`. The response is a 200 OK status with a response time of 21 ms and a body size of 257 B. The response body is displayed in JSON format:

```
1 {
2   "address": "Banyuwangi",
3   "id": 5,
4   "name": "Toko Kayu",
5   "phone": "0812912938123"
6 }
```

The screenshot shows the Postman interface with a workspace named 'Pertemuan5'. The active collection is 'Pertemuan5', and the selected request is 'Add New Store'. The request is a POST method to the endpoint `{{url}}/stores/9`. The response is a 200 OK status with a response time of 17 ms and a body size of 270 B. The response body is displayed in JSON format:

```
1 {
2   "address": "Depok",
3   "id": 9,
4   "name": "Mikail Thoziq Kaziemshah Banowo",
5   "phone": "68422881"
6 }
```

Add New  
Stores

