

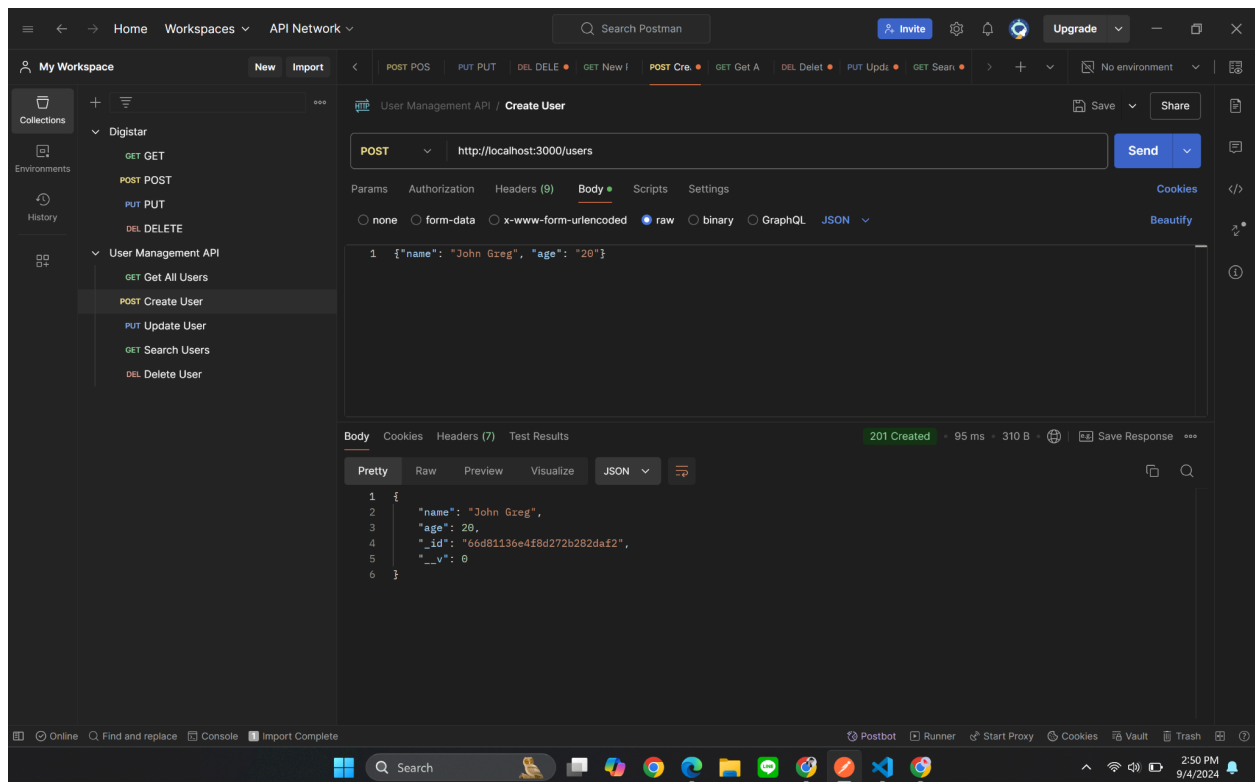
Nama: Ratin Kani
Kelas: Hacker Back End 8

Run express.js

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Kani\Downloads\NodeJS> node express.js
API server is running at http://localhost:3000
Pinged your deployment. You successfully connected to MongoDB!
```

Post data ke database



Postman interface showing a REST client request for "Create User" (POST) to "http://localhost:3000/users". The request body is raw JSON:

```
1 { "name": "John Nolan", "age": "54" }
```

. The response is 201 Created, 51 ms, 311 B, with a JSON body:

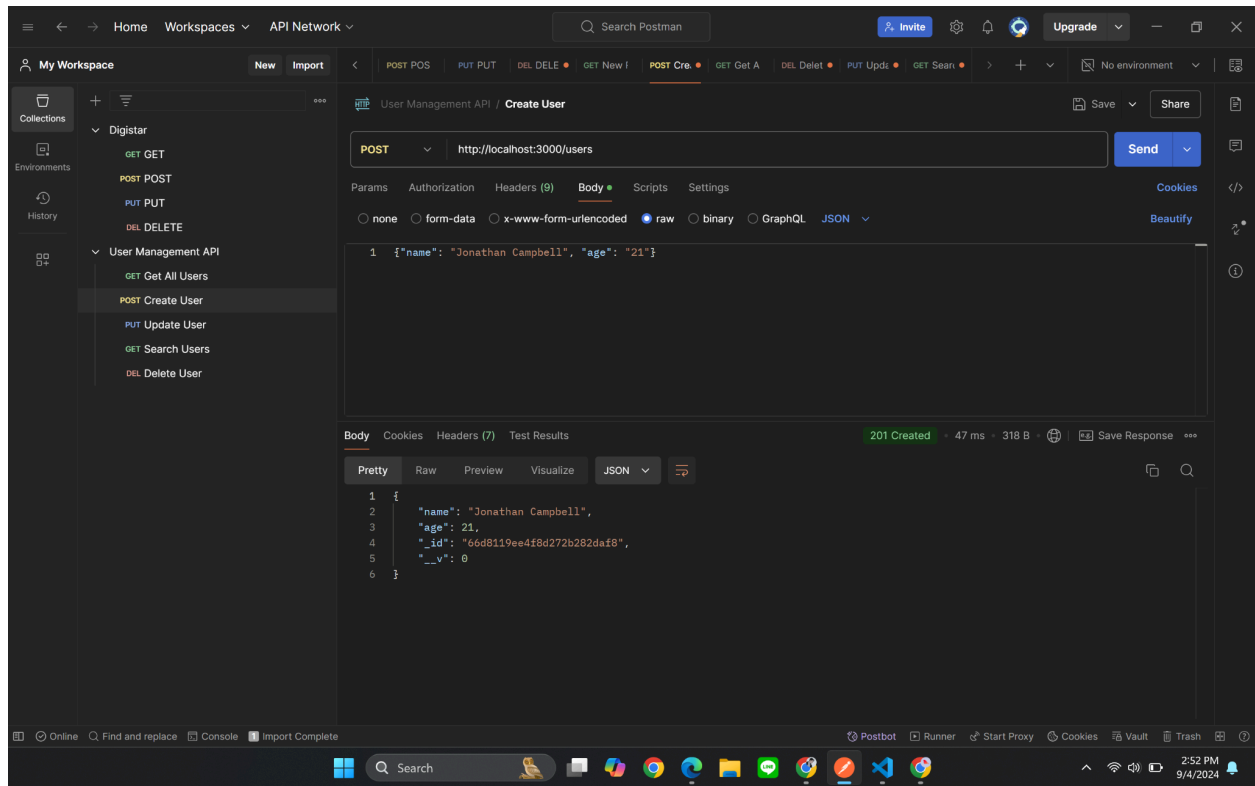
```
1 {
2   "name": "John Nolan",
3   "age": 54,
4   "id": "66d8115ce4f8d272b282daf4",
5   "__v": 0
6 }
```

Postman interface showing a REST client request for "Create User" (POST) to "http://localhost:3000/users". The request body is raw JSON:

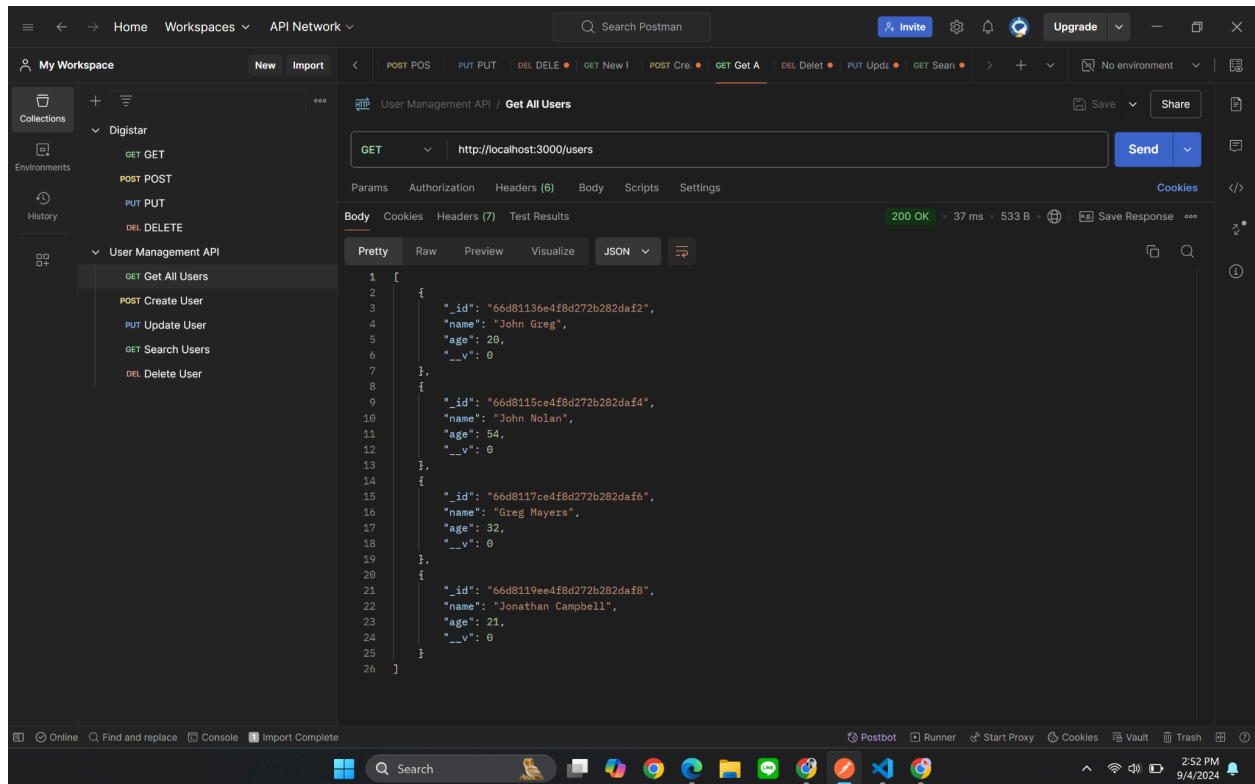
```
1 { "name": "Gzeg Mayers", "age": "32" }
```

. The response is 201 Created, 296 ms, 312 B, with a JSON body:

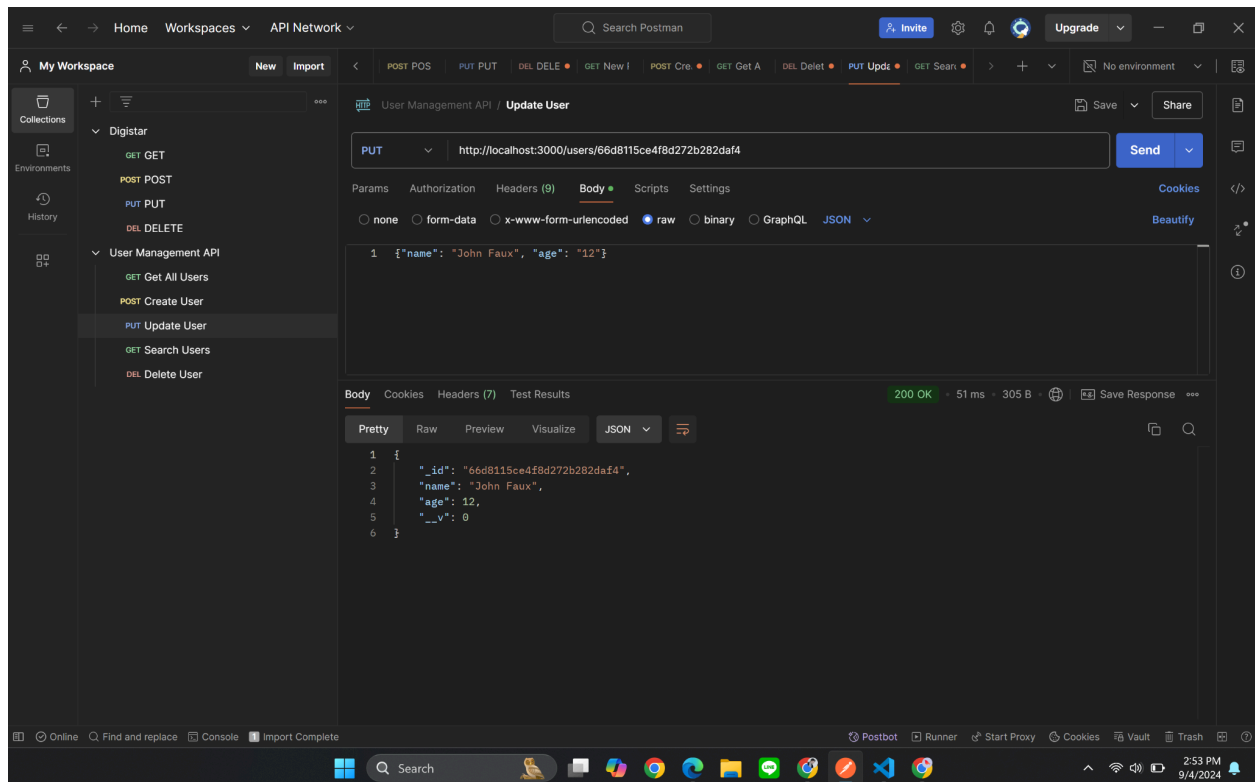
```
1 {
2   "name": "Gzeg Mayers",
3   "age": 32,
4   "id": "66d8117ce4f8d272b282daf6",
5   "__v": 0
6 }
```



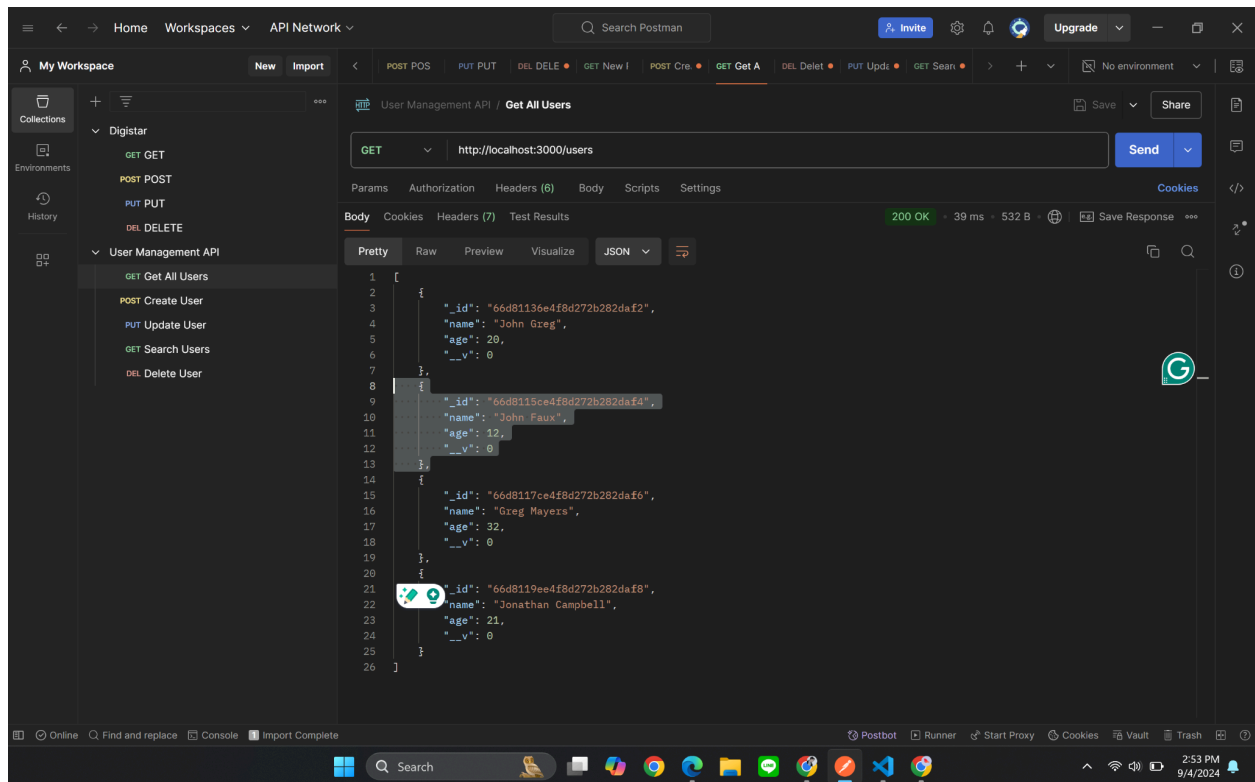
Hasil tambah data



Edit nama dan umur John Nolan - 54 menjadi John Faux - 12



Hasil edit



Search Greg Mayers

The screenshot shows the Postman interface with a workspace named "My Workspace". The "User Management API" collection is expanded, and the "Search Users" endpoint (GET) is selected. The request URL is `http://localhost:3000/search?name=Greg Mayers`. The response status is **200 OK** with a response time of 36 ms and a body size of 309 B. The response body is displayed in JSON format:

```
1 [
2   {
3     "_id": "66d8117ce4f8d272b282daf6",
4     "name": "Greg Mayers",
5     "age": 32,
6     "___v": 0
7   }
8 ]
```

Tambahkan data baru yaitu Greg Mayers 40

The screenshot shows the Postman interface with the same workspace. The "Create User" endpoint (POST) is selected. The request URL is `http://localhost:3000/users`. The request body is set to "raw" and contains the JSON: `{ "name": "Greg Mayers", "age": "40" }`. The response status is **201 Created** with a response time of 316 ms and a body size of 312 B. The response body is displayed in JSON format:

```
1 {
2   "name": "Greg Mayers",
3   "age": 40,
4   "_id": "66d8125d6ba8bef584e3bcb5",
5   "___v": 0
6 }
```

Search Greg Mayers akan menampilkan 2 orang

The screenshot shows the Postman application interface. On the left, the 'Collections' pane is open, showing a collection named 'User Management API' with several endpoints. The 'GET Search Users' endpoint is selected. The main pane displays the details of the selected endpoint, including the method (GET), the URL (http://localhost:3000/search?name=Greg Mayers), and the response body. The response body is a JSON array containing two user objects, both with the name 'Greg Mayers'.

Method: GET
URL: http://localhost:3000/search?name=Greg Mayers
Status: 200 OK
Response Body (JSON):

```
[{"_id": "66d8117ce4f8d272b282daf6", "name": "Greg Mayers", "age": 32, "__v": 0}, {"_id": "66d8125d6ba8bef584e3bcb5", "name": "Greg Mayers", "age": 40, "__v": 0}]
```

Delete Greg Mayers - 32

The screenshot shows the Postman application interface. On the left, the 'Collections' pane is open, showing a collection named 'User Management API' with several endpoints. The 'DEL Delete User' endpoint is selected. The main pane displays the details of the selected endpoint, including the method (DELETE), the URL (http://localhost:3000/users/66d8117ce4f8d272b282daf6), and the response body. The response body is a single character '1', indicating a successful deletion.

Method: DELETE
URL: http://localhost:3000/users/66d8117ce4f8d272b282daf6
Status: 200 OK
Response Body (JSON):

```
1
```

Hasil Akhir

