

Nama: Ratin Kani
Kelas: Hacker Back-End 8

Run server

```
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!
```

Registrasi User

The screenshot displays the Postman interface for a REST client. The workspace is named "My Workspace" and contains a collection named "User Management API". The selected endpoint is "POST Register" with the URL "http://localhost:3000/register". The request body is a JSON object:

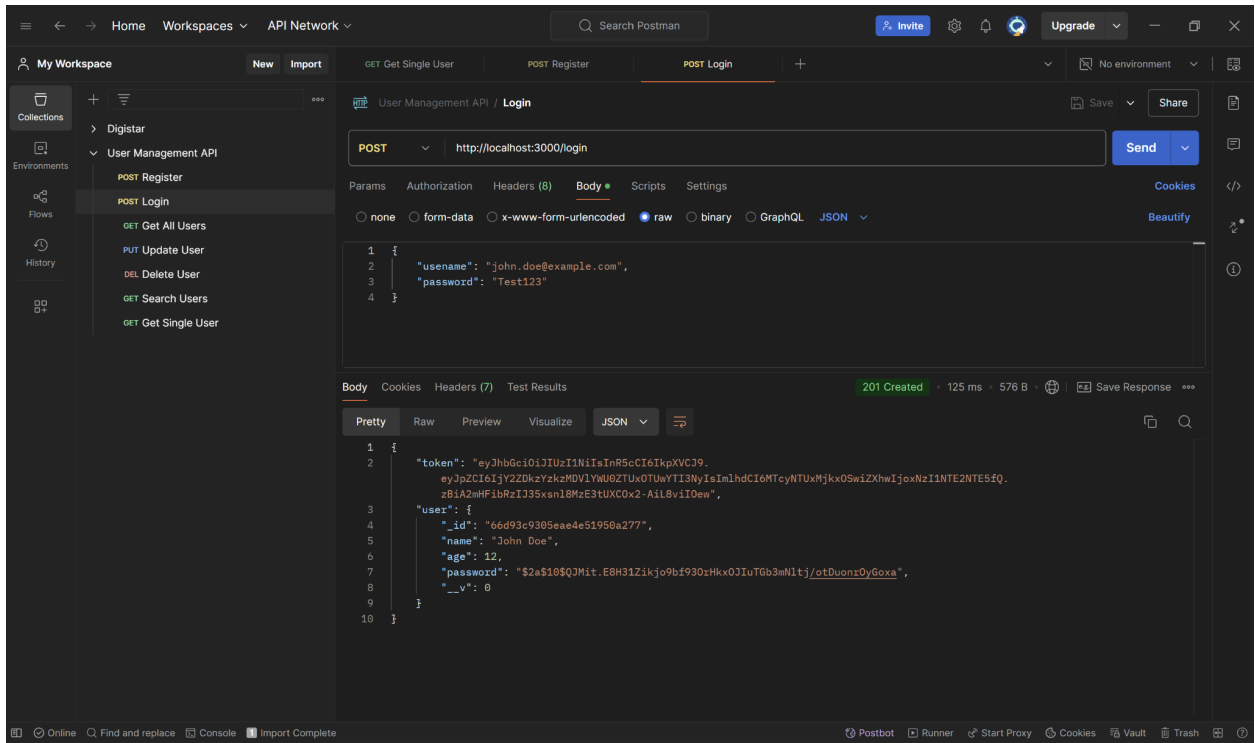
```
{
  "name": "John Doe",
  "age": 12,
  "username": "john.doe@example.com",
  "password": "Test123"
}
```

The response is a 201 Created status with a response time of 143 ms and a body size of 384 B. The response body is a JSON object:

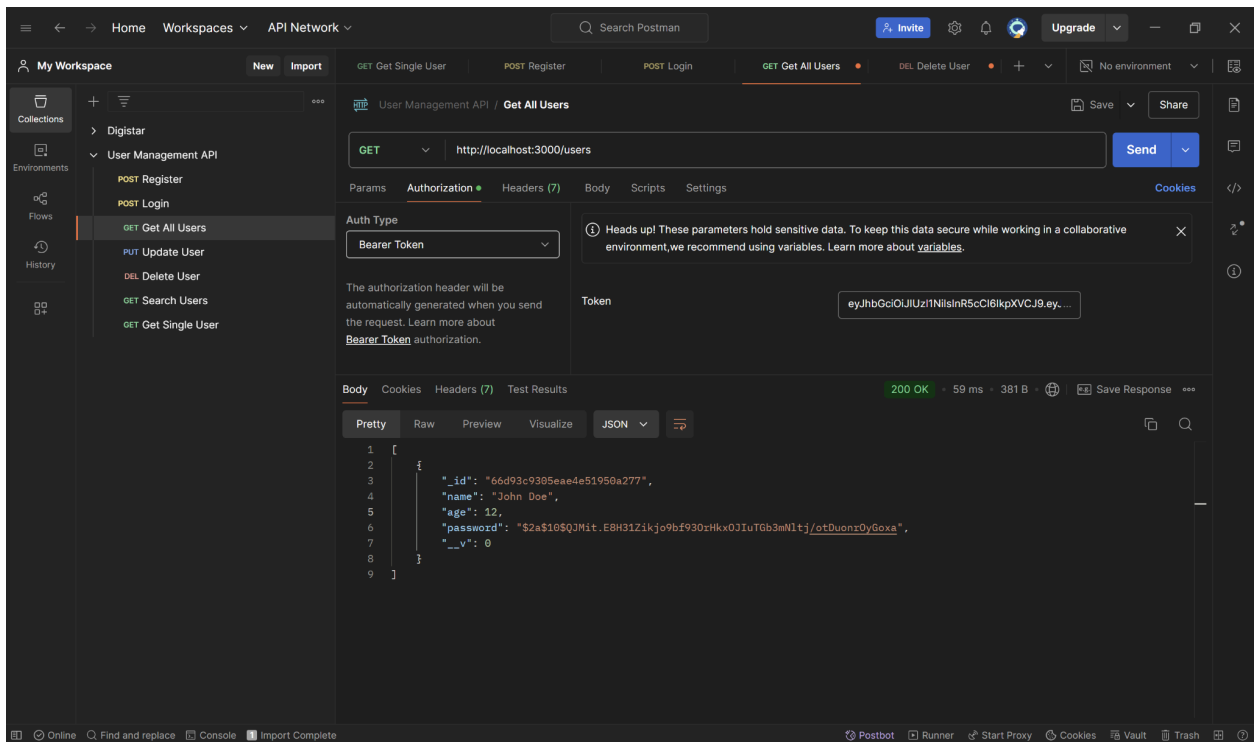
```
{
  "name": "John Doe",
  "age": 12,
  "password": "$2a$10$QJMit.E0H31Zitjo9bf930zHkx0JIuTgb3mNltj/otDuonx0yG0xa",
  "id": "66d93c9305eae4e51950a277",
  "_v": 0
}
```

The interface also shows a sidebar with a list of endpoints: "POST Register", "POST Login", "GET Get All Users", "PUT Update User", "DEL Delete User", "GET Search Users", and "GET Get Single User".

Login User dan copy token auth



Isi token auth dan get all user



Isi token auth dan get a single user

The screenshot shows the Postman interface for a 'User Management API'. The selected endpoint is 'GET Get Single User' with the URL 'http://localhost:3000/users/66d93c9305eae4e51950a277'. The 'Authorization' tab is active, showing 'Bearer Token' and a token value: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1b290a277IjoiIiwiaWF0IjoiMTY5MjY0MjY0In0'. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables.' The 'Body' tab shows a successful '200 OK' response with a status of 39 ms and 379 B. The response body is a JSON object:

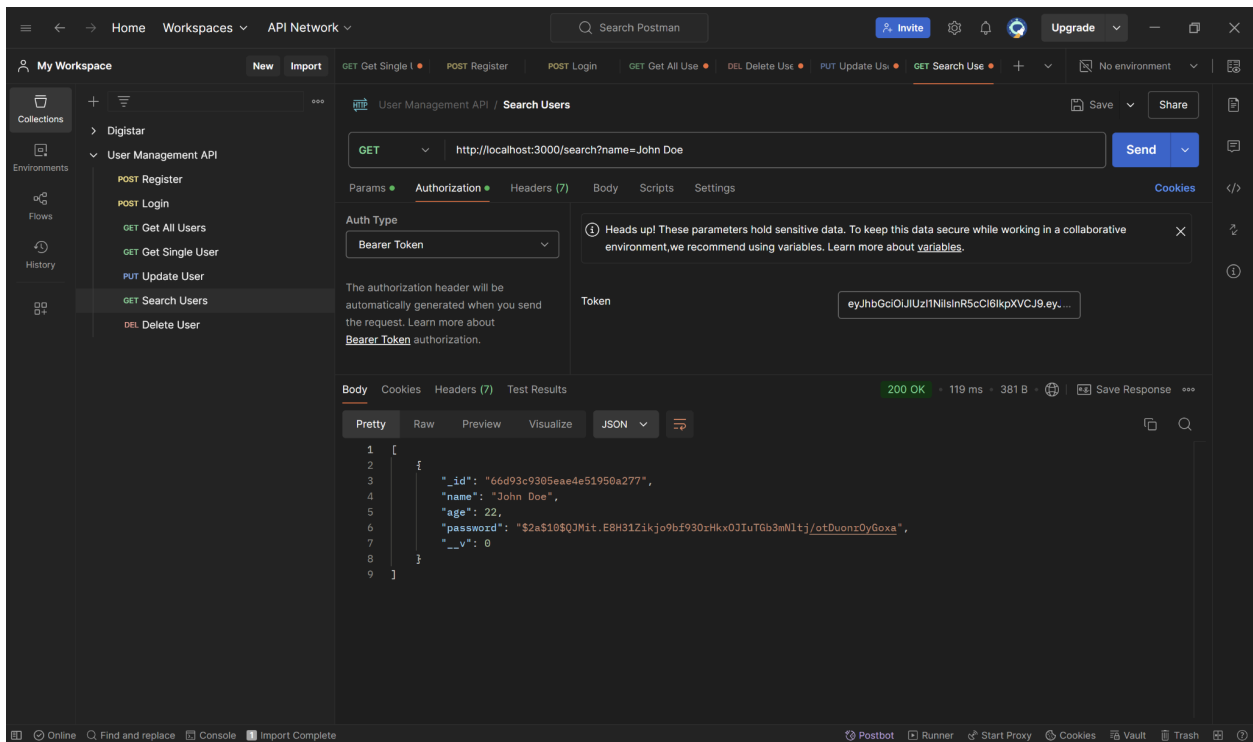
```
1 {
2   "id": "66d93c9305eae4e51950a277",
3   "name": "John Doe",
4   "age": 12,
5   "password": "$2a$10$QJMit.E8H31Zikjo9bf930zHkx0J1uT6b3mNltj/otDuonrOyGoxa",
6   "__v": 0
7 }
```

Isi token auth dan update age user menjadi 22

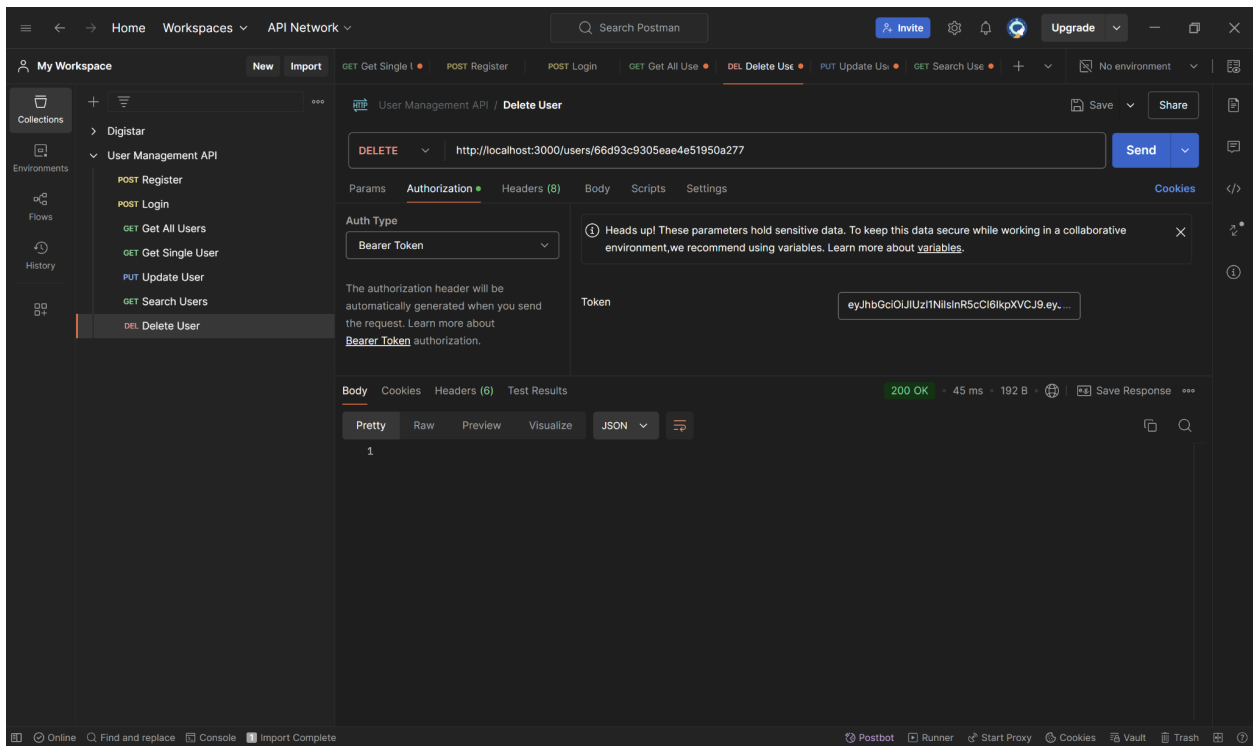
The screenshot shows the Postman interface for the 'User Management API'. The selected endpoint is 'PUT Update User' with the URL 'http://localhost:3000/users/66d93c9305eae4e51950a277'. The 'Authorization' tab is active, showing 'Bearer Token' and the same token value as the previous screenshot. The 'Body' tab is active, showing a request body: '1 { "name": "John Doe", "age": 22 }'. The 'Body' tab shows a successful '200 OK' response with a status of 69 ms and 379 B. The response body is a JSON object:

```
1 {
2   "id": "66d93c9305eae4e51950a277",
3   "name": "John Doe",
4   "age": 22,
5   "password": "$2a$10$QJMit.E8H31Zikjo9bf930zHkx0J1uT6b3mNltj/otDuonrOyGoxa",
6   "__v": 0
7 }
```

Isi token auth dan search user



Isi token auth dan hapus user



Hasil akhir

The screenshot displays the Postman API client interface. On the left sidebar, the 'My Workspace' is selected, showing a collection named 'User Management API' with several endpoints. The 'GET Get All Users' endpoint is highlighted. The main panel shows the details of this request: a GET method to the URL 'http://localhost:3000/users'. The 'Authorization' tab is active, showing 'Bearer Token' as the auth type. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables.' The 'Token' field contains the value 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...'. The 'Body' tab is selected, showing a '200 OK' status with a response time of 58 ms and a size of 235 B. The response body is displayed in JSON format as an empty array '[]'.

Home Workspaces API Network Search Postman Invite Upgrade

My Workspace New Import GET Get Single t POST Register POST Login GET Get All Use DEL Delete Use PUT Update Us GET Search Use

Collections + -

Environments

Flows

History

User Management API

POST Register

POST Login

GET Get All Users

GET Get Single User

PUT Update User

GET Search Users

DEL Delete User

User Management API / Get All Users

GET http://localhost:3000/users Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Auth Type

Bearer Token

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables.

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body Cookies Headers (7) Test Results 200 OK 58 ms 235 B Save Response

Pretty Raw Preview Visualize JSON

1 []

Online Find and replace Console Import Complete Postbot Runner Start Proxy Cookies Vault Trash