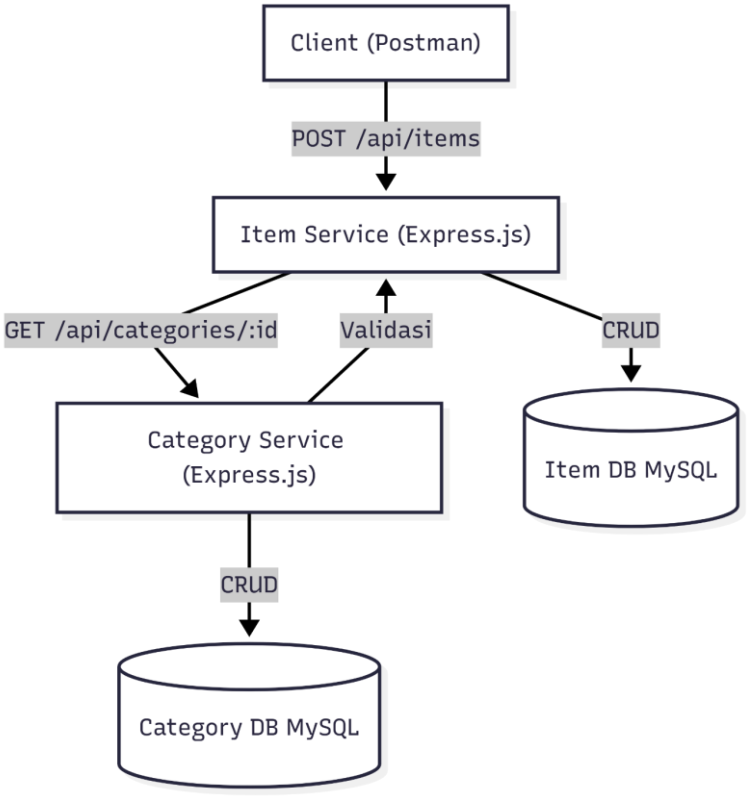
2. Diagram Arsitektur Microservice

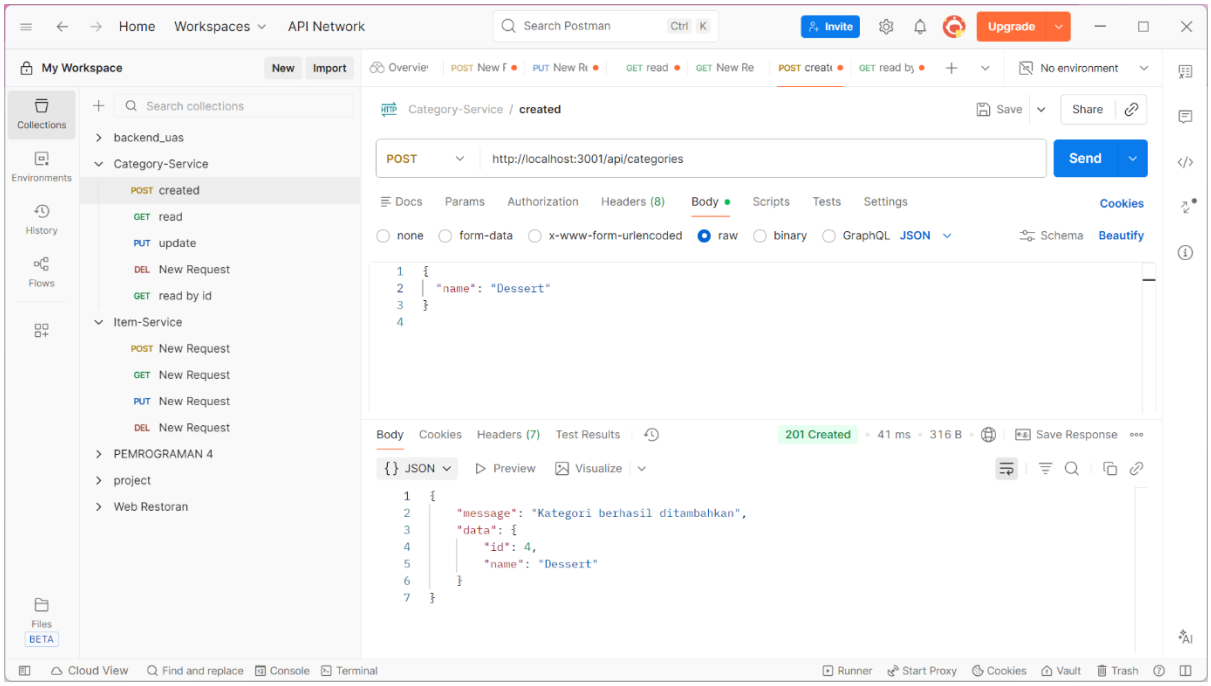


1. **Client → Category Service**
   Client dapat mengirimkan request langsung ke Category Service untuk melakukan operasi CRUD data kategori, seperti menambah, menampilkan, mengubah, dan menghapus kategori. Category Service akan memproses request tersebut dan berinteraksi langsung dengan Category Database.
2. **Client → Item Service**
   Client juga dapat mengirimkan request ke Item Service untuk melakukan operasi CRUD data item. Item Service bertanggung jawab untuk mengelola data item dan menyimpannya ke Item Database.
3. **Item Service → Category Service (Komunikasi Antar Service)**
   Pada saat proses pembuatan atau pembaruan data item, Item Service melakukan komunikasi ke Category Service melalui REST API untuk memvalidasi data kategori berdasarkan category_id. Hal ini memastikan bahwa item hanya dapat dibuat atau diperbarui apabila kategori yang digunakan benar-benar tersedia.
4. **Service → Database**
   Setiap service memiliki database masing-masing dan hanya service tersebut yang memiliki akses langsung ke databasenya.
   - Category Service ↔ Category Database
   - Item Service ↔ Item Database
5. **Response ke Client**
   Setelah proses selesai, masing-masing service akan mengembalikan response dalam format JSON kepada client, baik berupa data hasil proses maupun pesan status.

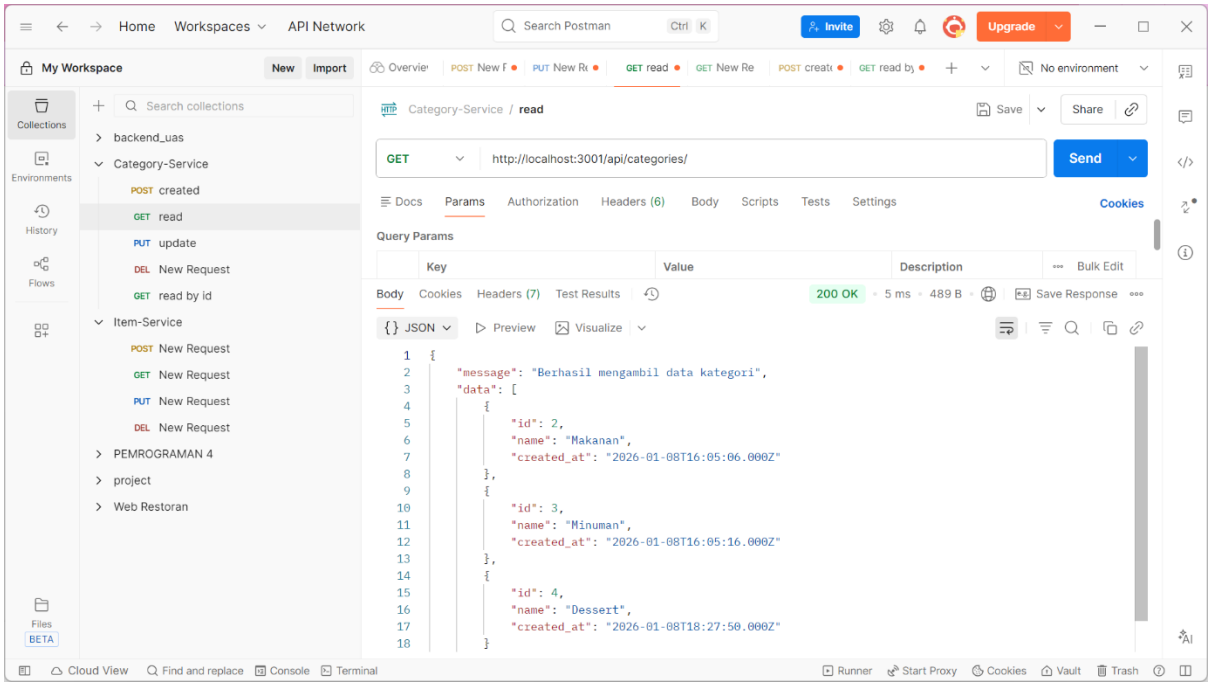
6. Screenshot hasil pengujian

Category Service

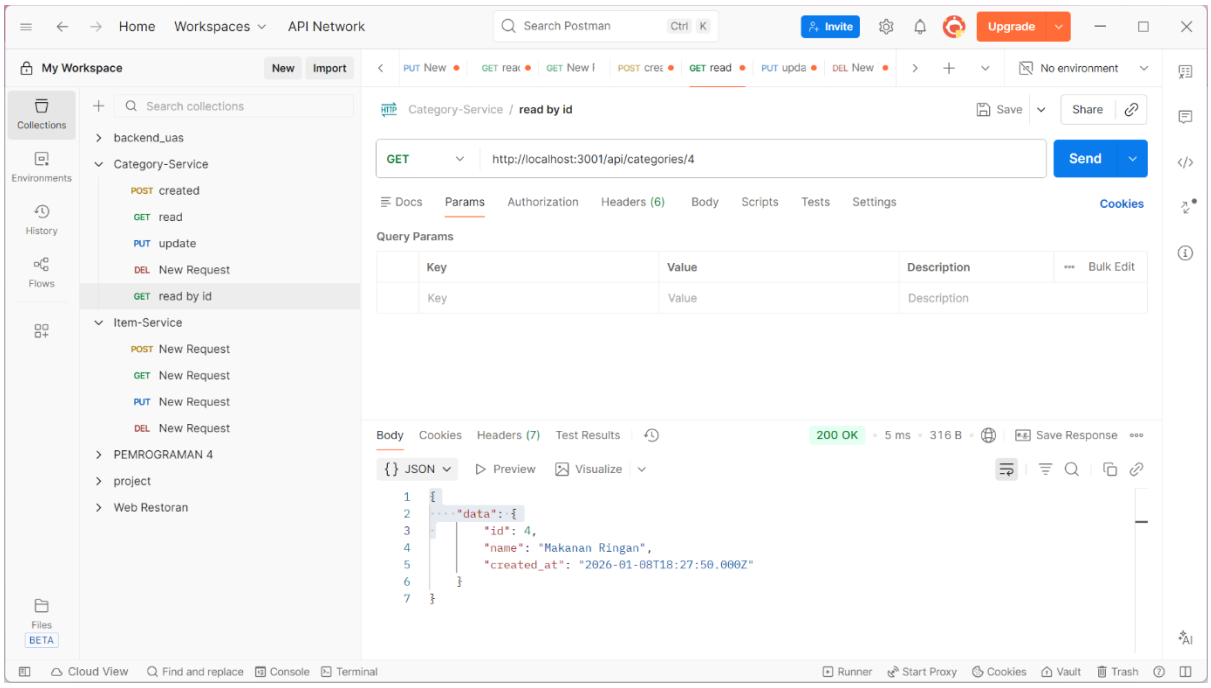
- **Create**



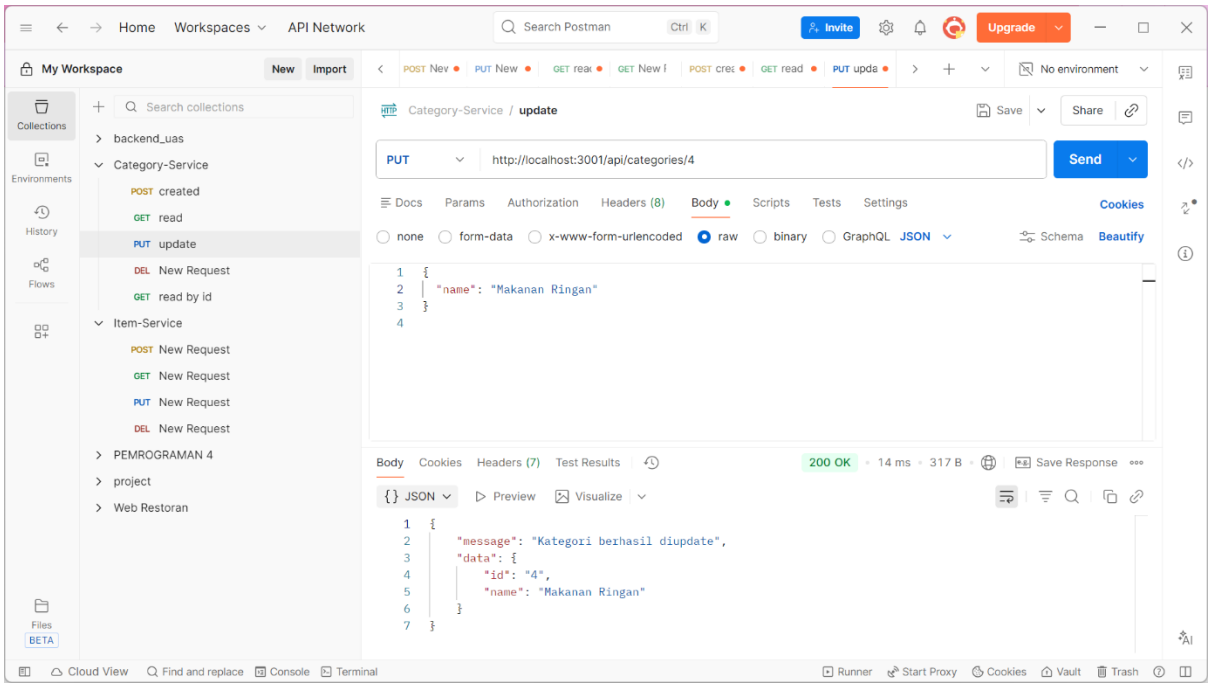
- **Read**



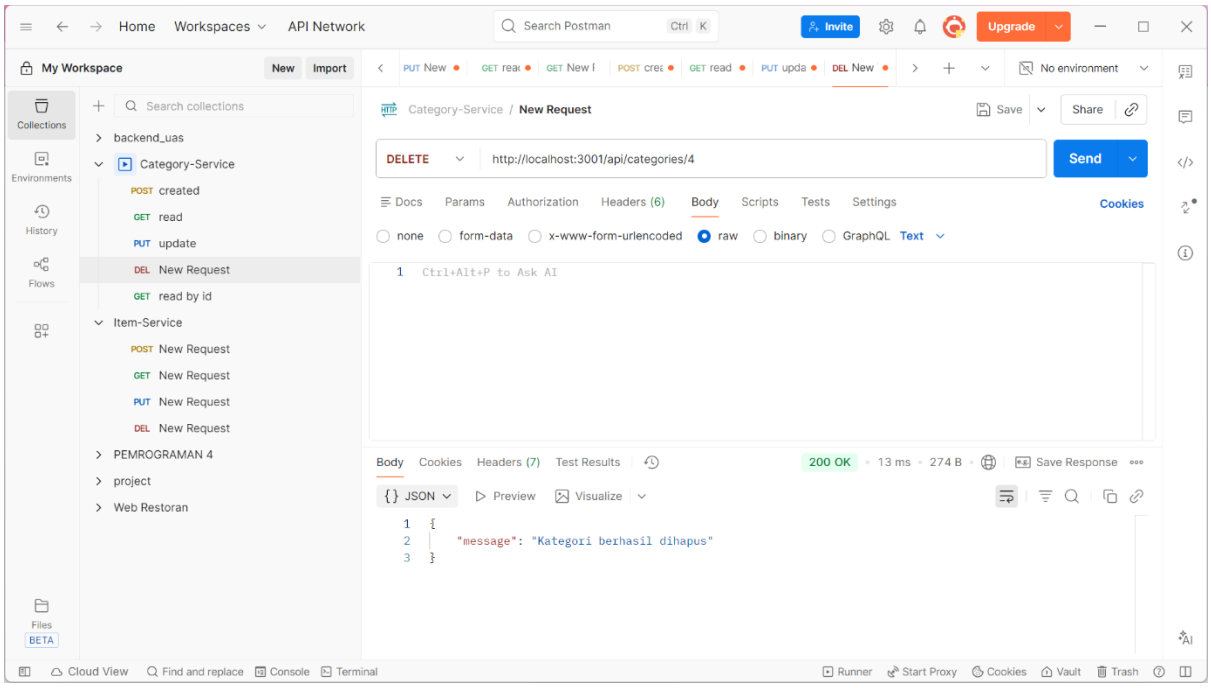
- **Read by ID**



- **Update**

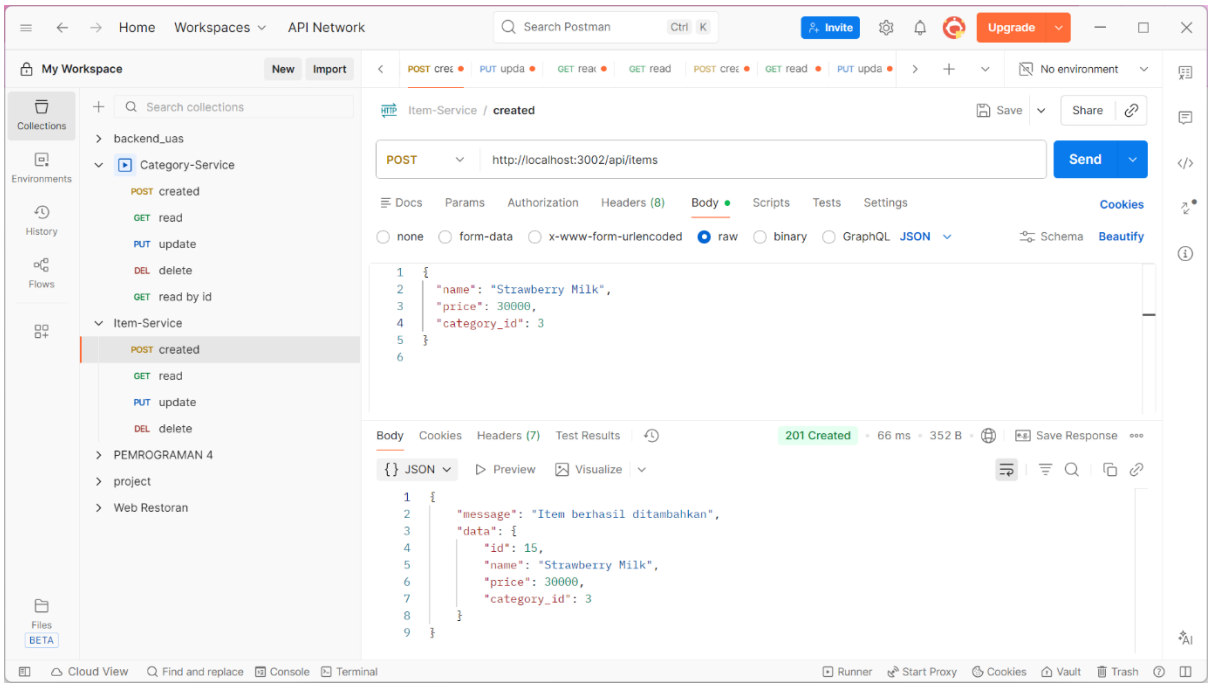


- **Delete**

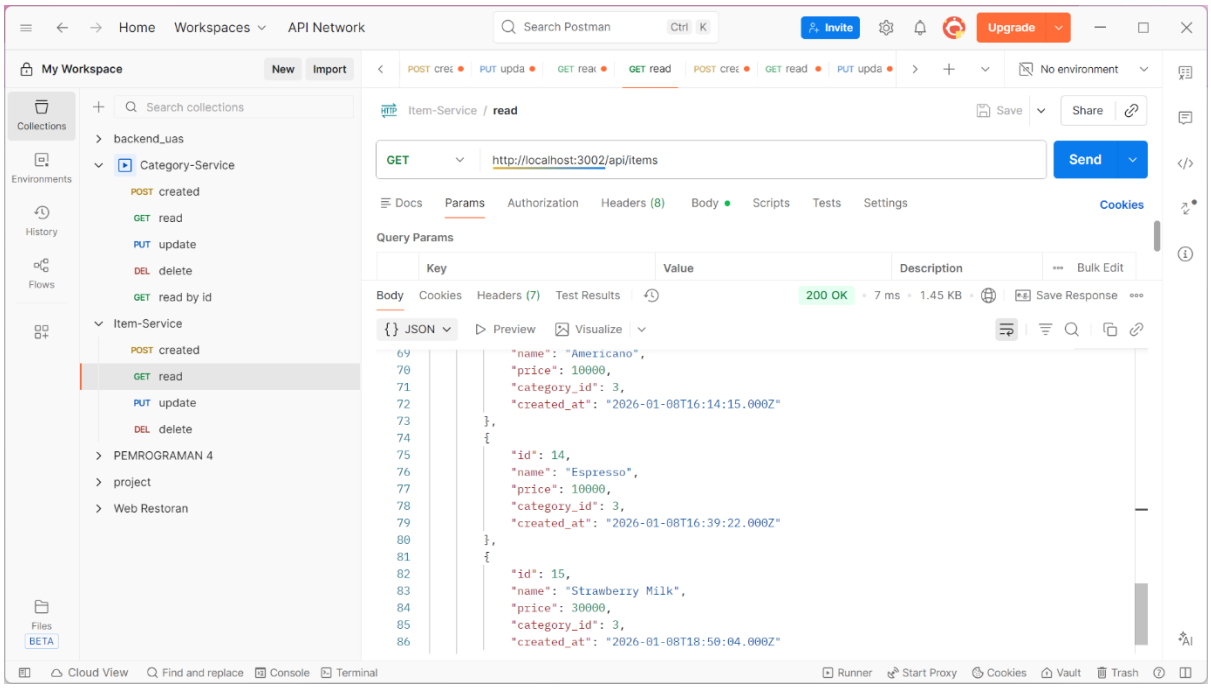


Item Service

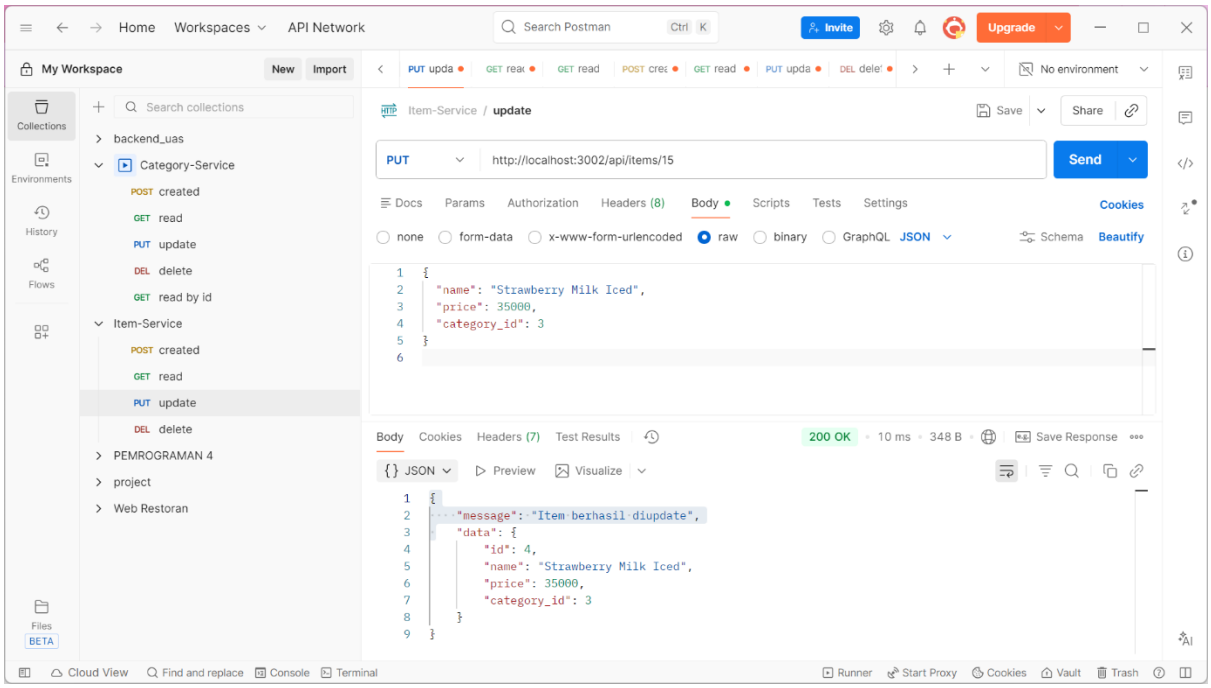
- **Create**



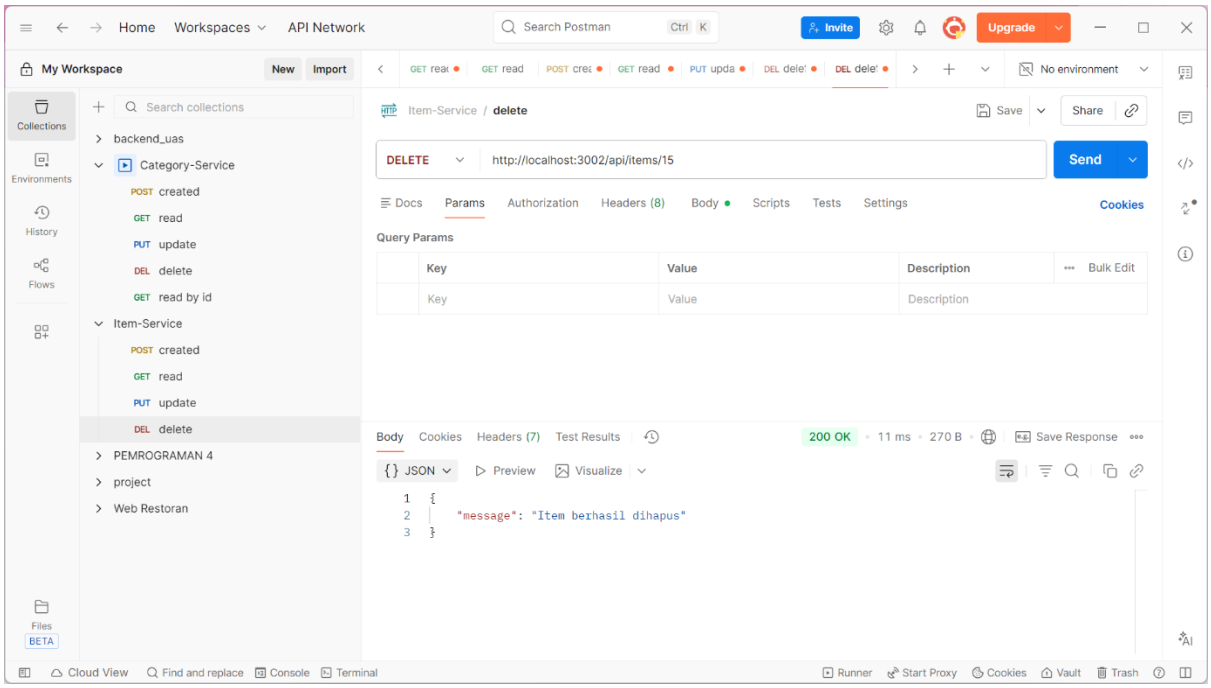
- **Read**


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
No environment
Collections
Environments
History
Flows
Files
BETA
Search collections
backend_uas
Category-Service
POST created
GET read
PUT update
DEL delete
GET read by id
Item-Service
POST created
GET read
PUT update
DEL delete
PEMROGRAMAN 4
project
Web Restoran
Item-Service / read
Save
Share
GET
http://localhost:3002/api/items
Send
Docs
Params
Authorization
Headers (8)
Body
Scripts
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (7)
Test Results
200 OK
7 ms
1.45 KB
Save Response
JSON
Preview
Visualize
"name": "Americano",
"price": 10000,
"category_id": 3,
"created_at": "2026-01-08T16:14:15.000Z"
},
{
"id": 14,
"name": "Espresso",
"price": 10000,
"category_id": 3,
"created_at": "2026-01-08T16:39:22.000Z"
},
{
"id": 15,
"name": "Strawberry Milk",
"price": 30000,
"category_id": 3,
"created_at": "2026-01-08T18:50:04.000Z"
Cloud View
Find and replace
Console
Terminal
Runner
Start Proxy
Cookies
Vault
Trash

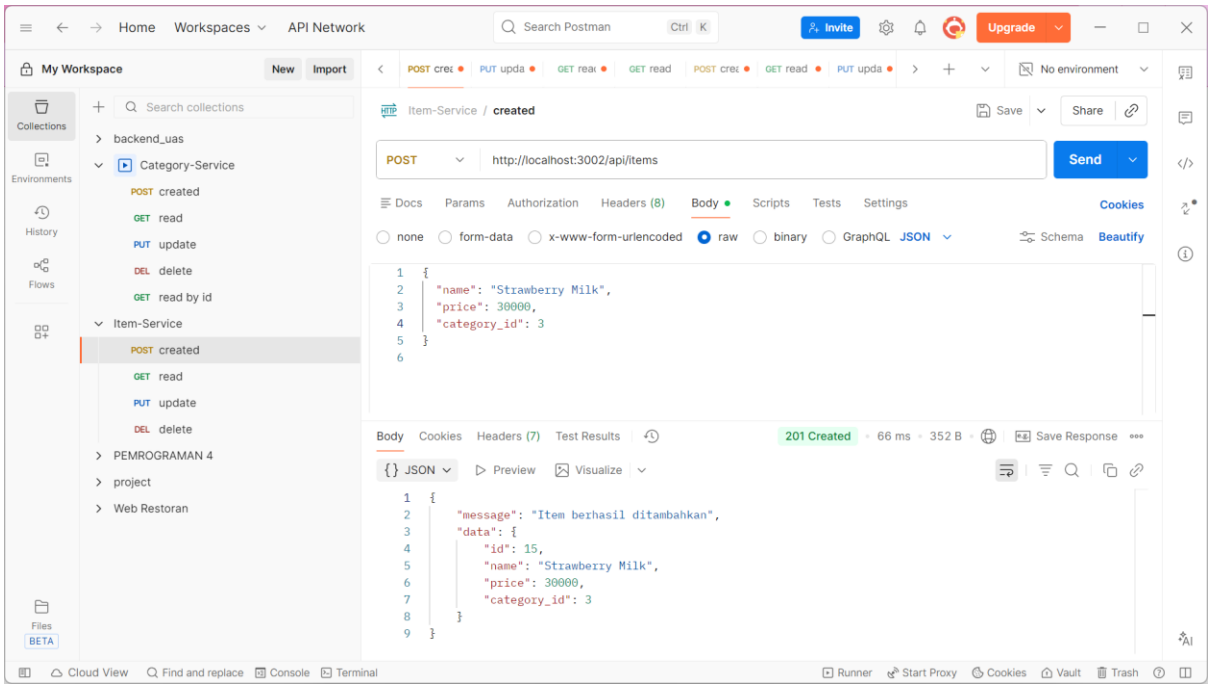
- **Update**


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
No environment
Collections
Environments
History
Flows
Files
BETA
Search collections
backend_uas
Category-Service
POST created
GET read
PUT update
DEL delete
GET read by id
Item-Service
POST created
GET read
PUT update
DEL delete
PEMROGRAMAN 4
project
Web Restoran
Item-Service / update
Save
Share
PUT
http://localhost:3002/api/items/15
Send
Docs
Params
Authorization
Headers (8)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Schema
Beautify
{
"name": "Strawberry Milk Iced",
"price": 35000,
"category_id": 3
}
Body
Cookies
Headers (7)
Test Results
200 OK
10 ms
348 B
Save Response
JSON
Preview
Visualize
{
"message": "Item berhasil diupdate",
"data": {
"id": 4,
"name": "Strawberry Milk Iced",
"price": 35000,
"category_id": 3
}
}
Cloud View
Find and replace
Console
Terminal
Runner
Start Proxy
Cookies
Vault
Trash

- **Delete**



- **Komunikasi antar service**



Pada proses pembuatan item, Item Service melakukan request ke Category Service untuk memvalidasi kategori berdasarkan category_id.