

# **Laporan Praktek Topik Khusus 2**

Menjalankan CRUD API dengan Golang, MySQL, dan Redis



**SEMESTER VI**

**DISUSUN OLEH :**

**KURNIAWAN ALEXANDER**

**2211083030**

**DOSEN PENGAMPU :**

**YULHERNIWATI, S.Kom.,MT**

**YUNUS SUPRIADI WIJAYA**

**PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI PADANG**

**2025**

## **A. Dasar Teori**

Dalam pengembangan aplikasi web, penggunaan database dan caching berperan penting dalam meningkatkan kinerja sistem. Tutorial ini membahas cara mengintegrasikan MySQL sebagai database utama dengan Redis sebagai caching layer untuk mempercepat akses data. MySQL merupakan sistem manajemen basis data relasional (RDBMS) yang menggunakan SQL (Structured Query Language) untuk mengelola data. Keandalannya dalam menangani transaksi yang aman, struktur data yang solid, serta dukungan untuk aplikasi berskala besar menjadikannya pilihan populer. Namun, seiring meningkatnya jumlah permintaan data, query langsung ke MySQL dapat menyebabkan waktu respons yang lebih lama.

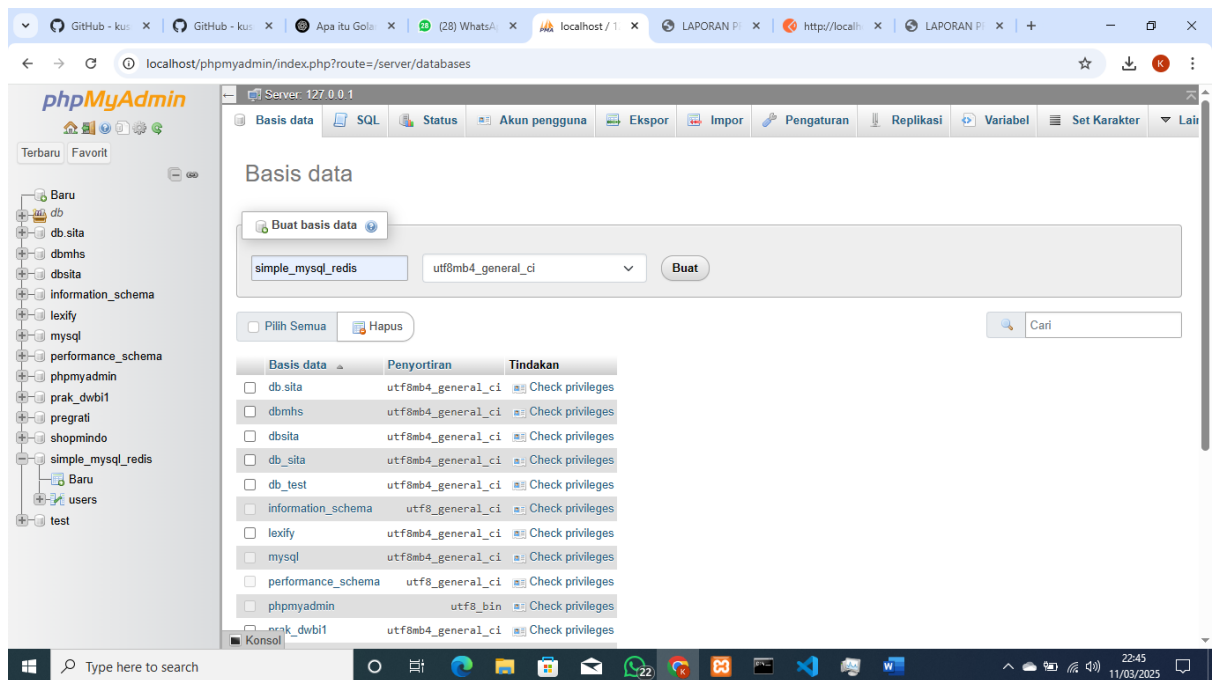
Untuk mengatasi hal tersebut, Redis digunakan sebagai sistem caching berbasis key-value yang berjalan di dalam memori (RAM). Dengan Redis, data yang sering diakses dapat disimpan sementara, sehingga permintaan berikutnya dapat diproses lebih cepat tanpa harus mengakses MySQL secara langsung. Dalam tutorial ini, Redis dan MySQL diintegrasikan menggunakan teknik caching. Sistem akan terlebih dahulu memeriksa apakah data sudah tersedia di Redis. Jika tidak ditemukan, data diambil dari MySQL dan kemudian disimpan di Redis untuk mempercepat permintaan berikutnya serta mengurangi beban pada server database. Pendekatan ini memungkinkan aplikasi merespons lebih cepat dan bekerja lebih efisien.

## **B. Tools**

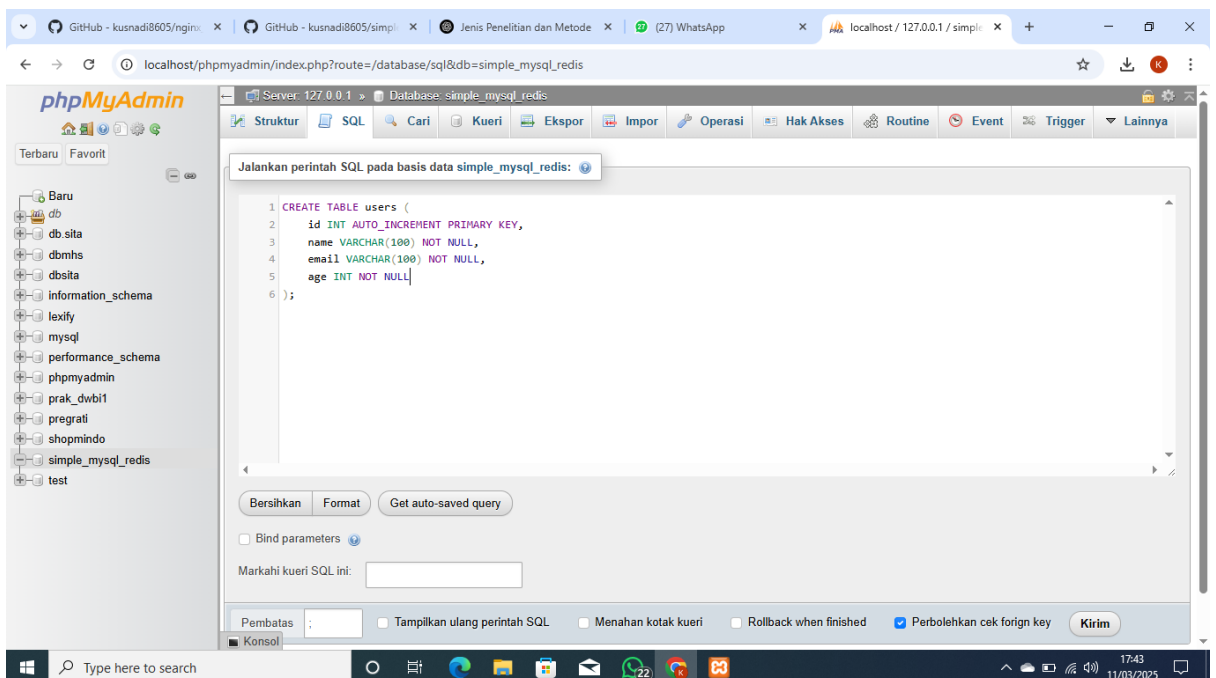
- Visual Studio Code
- MySQL ( XAMPP )
- Golang
- Command Prompt (CMD)
- Redis
- Postman

## C. Langkah Kerja

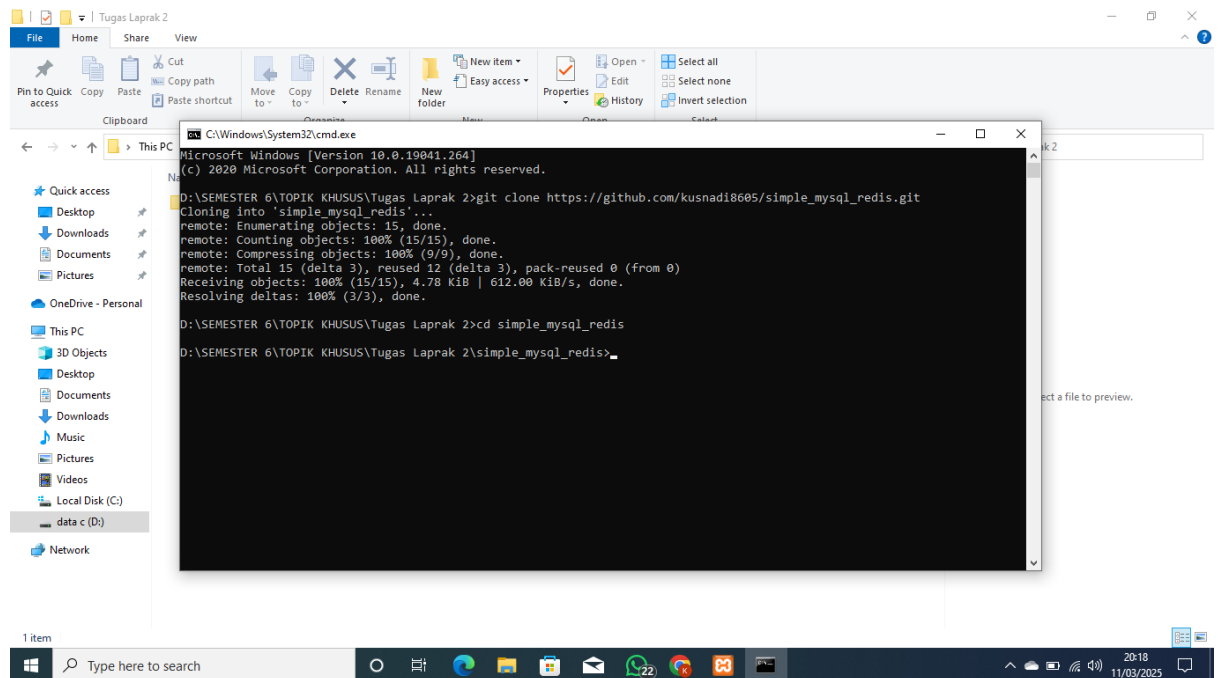
### 1. Buat Database dengan nama simple\_mysql\_redis



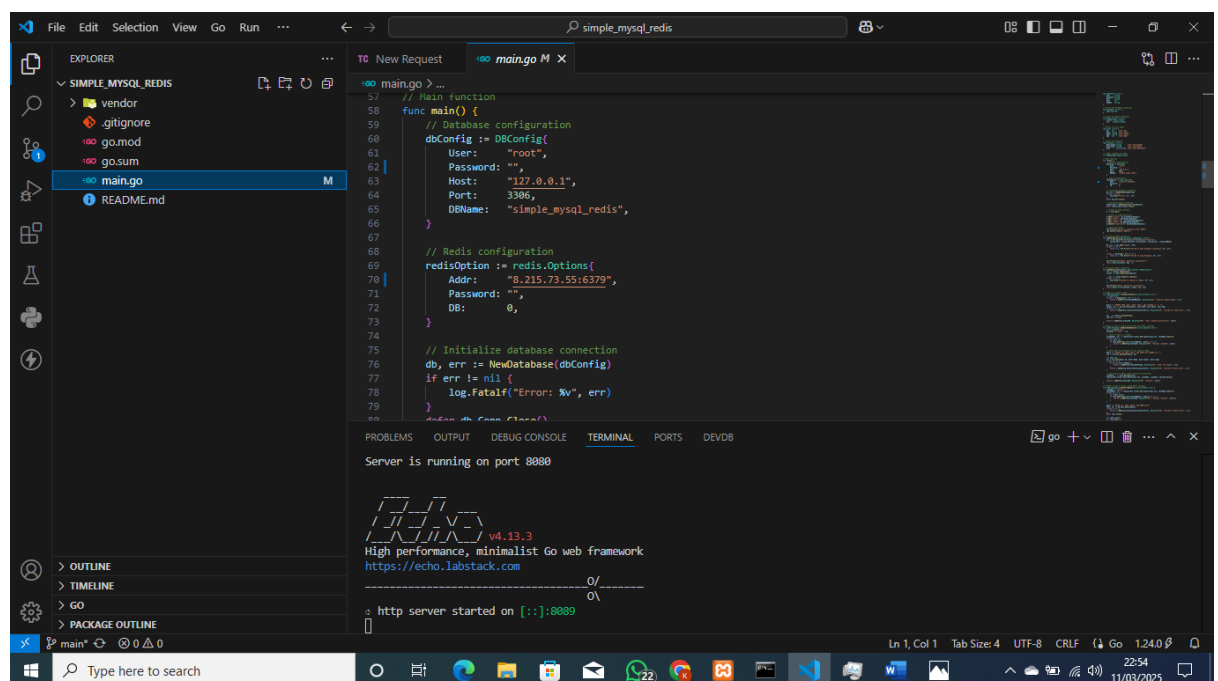
### 2. Buat Tabel User



### 3. Lakukan klon Repository dari github



### 4. Konfigurasi database dan redis agar bias saling terhubung pada file main.go. Pada dbConfig hapus atau kosongkan password. Pada konfigurasi Redis gunakan IP Onlinenya yaitu 8.215.73.55:6379 menggantikan localhost.



## 5. Instal depedensi yang diperlukan

```
C:\Windows\System32\cmd.exe - go mod vendor
Microsoft Windows [Version 10.0.19041.264]
(c) 2020 Microsoft Corporation. All rights reserved.

D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2>git clone https://github.com/kusnadi8605/simple_mysql_redis.git
Cloning into 'simple_mysql_redis'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 15 (delta 3), reused 12 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 4.78 KiB | 612.00 KiB/s, done.
Resolving deltas: 100% (3/3), done.

D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2>cd simple_mysql_redis

D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2\simple_mysql_redis>go mod init simple_mysql_redis
go: creating new go.mod: module simple_mysql_redis
go: to add module requirements and sums:
    go mod tidy

D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2\simple_mysql_redis>go mod tidy
go: finding module for package github.com/redis/go-redis/v9
go: finding module for package github.com/go-sql-driver/mysql
go: finding module for package github.com/labstack/echo/v4
go: downloading github.com/go-sql-driver/mysql v1.9.0
go: downloading github.com/redis/go-redis/v9 v9.7.1
go: downloading github.com/redis/go-redis v6.15.9+incompatible
go: found github.com/go-sql-driver/mysql in github.com/go-sql-driver/mysql v1.9.0
go: found github.com/labstack/echo/v4 in github.com/labstack/echo/v4 v4.13.3
go: found github.com/redis/go-redis/v9 in github.com/redis/go-redis/v9 v9.7.1
go: downloading filippo.io/edwards25519 v1.1.0
go: downloading github.com/cespare/xxhash/v2 v2.2.0
go: downloading github.com/dgryski/go-rendezvous v0.0.0-20200823014737-9f7001d12a5f
go: downloading github.com/bsm/ginkgo/v2 v2.12.0
go: downloading github.com/bsm/gomega v1.27.10

D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2\simple_mysql_redis>go mod vendor
```

```
File Edit Selection View Go Run ...
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will
editors. To keep them

D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2>simple_mysql_redis>go mod init simple_mysql_redis
go: creating new go.mod: module simple_mysql_redis
go: to add module requirements and sums:
    go mod tidy

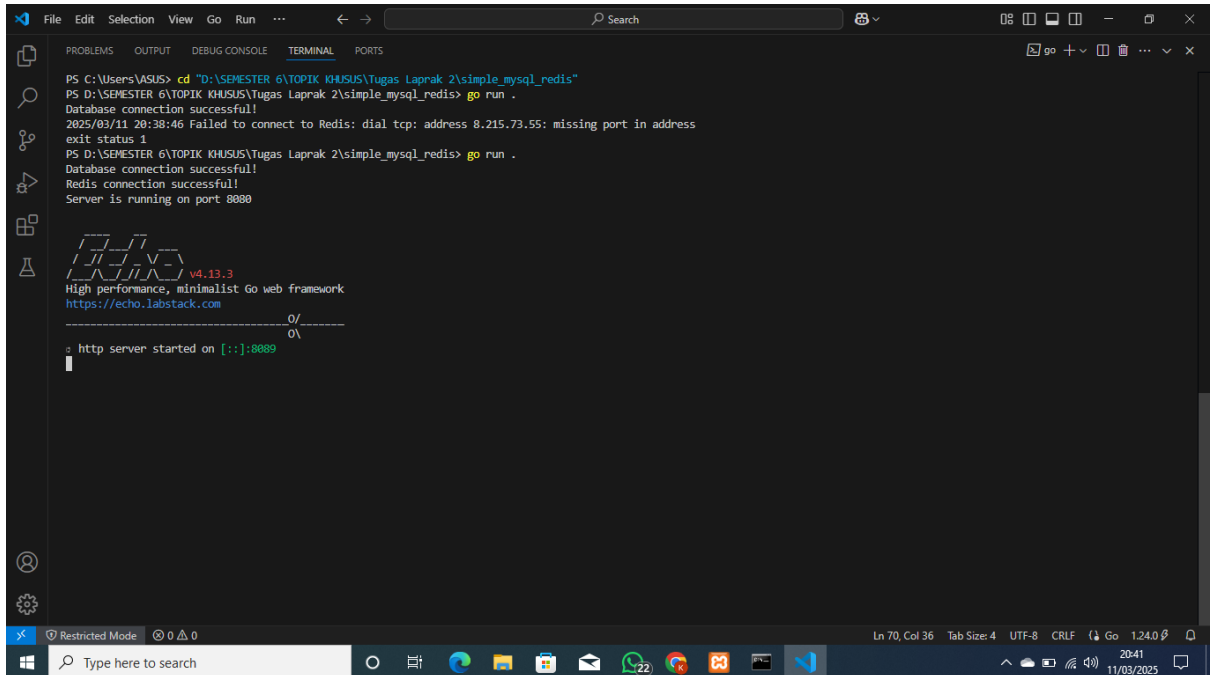
D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2\simple_mysql_redis>go mod tidy
go: finding module for package github.com/redis/go-redis/v9
go: finding module for package github.com/go-sql-driver/mysql
go: finding module for package github.com/labstack/echo/v4
go: downloading github.com/go-sql-driver/mysql v1.9.0
go: downloading github.com/redis/go-redis/v9 v9.7.1
go: downloading github.com/redis/go-redis v6.15.9+incompatible
go: found github.com/go-sql-driver/mysql in github.com/go-sql-driver/mysql v1.9.0
go: found github.com/labstack/echo/v4 in github.com/labstack/echo/v4 v4.13.3
go: found github.com/redis/go-redis/v9 in github.com/redis/go-redis/v9 v9.7.1
go: downloading filippo.io/edwards25519 v1.1.0
go: downloading github.com/cespare/xxhash/v2 v2.2.0
go: downloading github.com/dgryski/go-rendezvous v0.0.0-20200823014737-9f7001d12a5f
go: downloading github.com/bsm/ginkgo/v2 v2.12.0
go: downloading github.com/bsm/gomega v1.27.10

D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2\simple_mysql_redis>go mod vendor

D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2\simple_mysql_redis>

// Routes for CRUD operations
e.POST("/users", db.CreateUserHandler)
e.GET("/users", db.GetUsersHandler)
e.GET("/users/:id", db.GetUserByIdHandler)
```

## 6. Jalankan Server



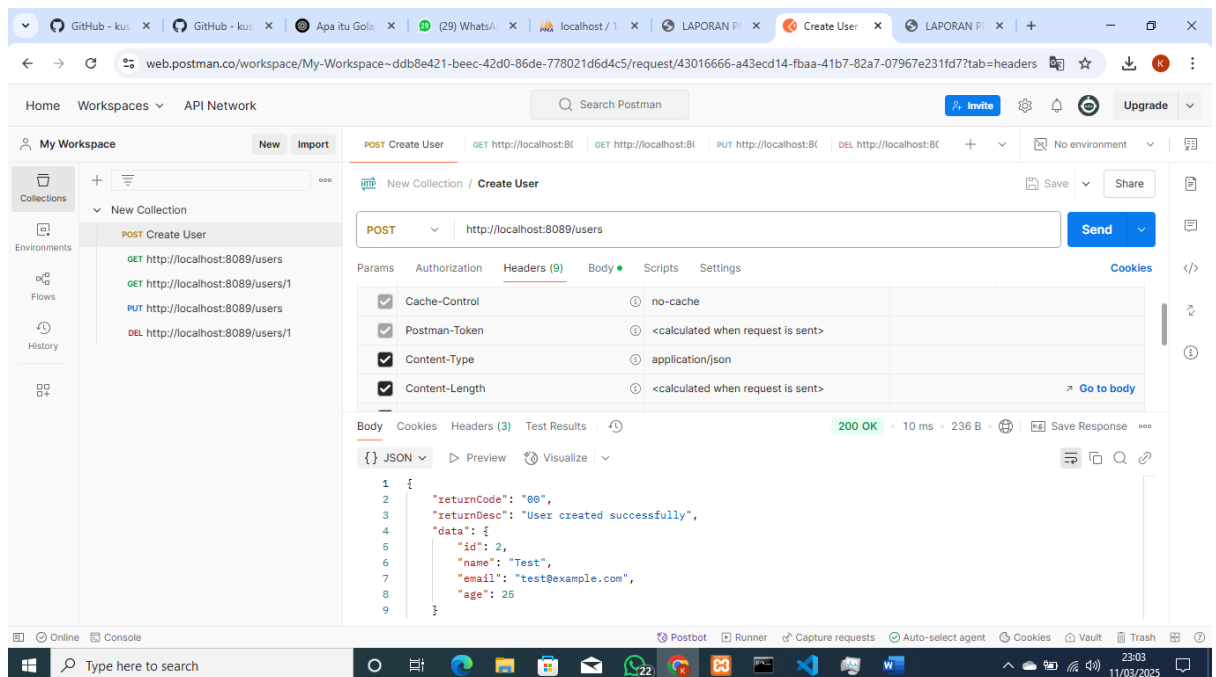
```
PS C:\Users\ASUS> cd "D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2\simple_mysql_redis"
PS D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2\simple_mysql_redis> go run .
Database connection successful!
2025/03/11 20:38:46 Failed to connect to Redis: dial tcp: address 8.215.73.55: missing port in address
exit status 1
PS D:\SEMESTER 6\TOPIK KHUSUS\Tugas Laprak 2\simple_mysql_redis> go run .
Database connection successful!
Redis connection successful!
Server is running on port 8089

v4.13.3
High performance, minimalist Go web framework
https://echo.labstack.com

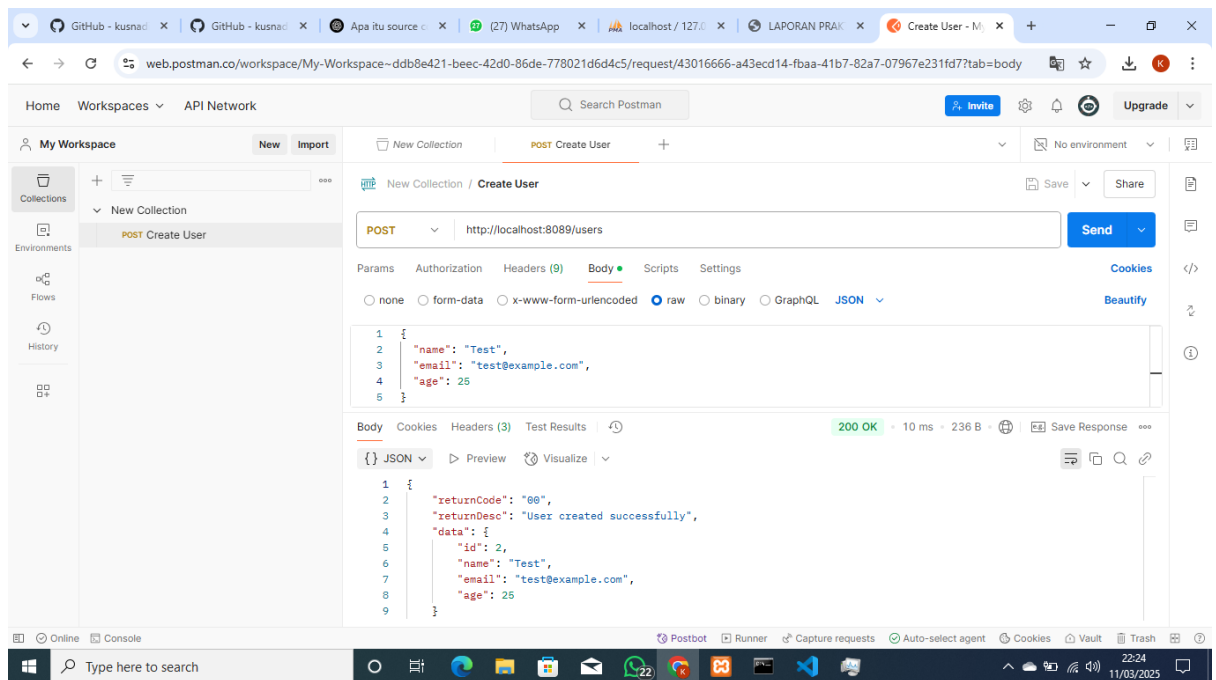
: http server started on [::]:8089
```

## 7. Selanjutnya, menjalankan CRUD API dengan Postman

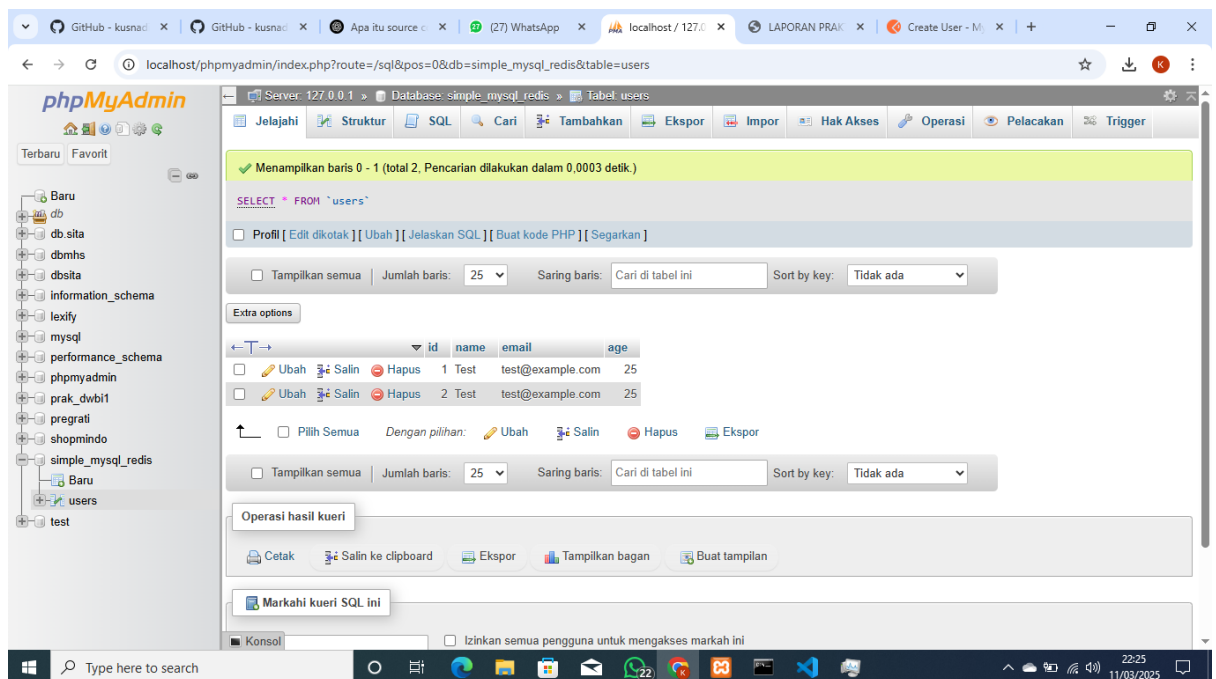
8. Pertama Create User, lakukan POST pada postman dengan url <http://localhost:8089/users>. Pada Header isi Key dengan Content-Type dan Values dengan application/json



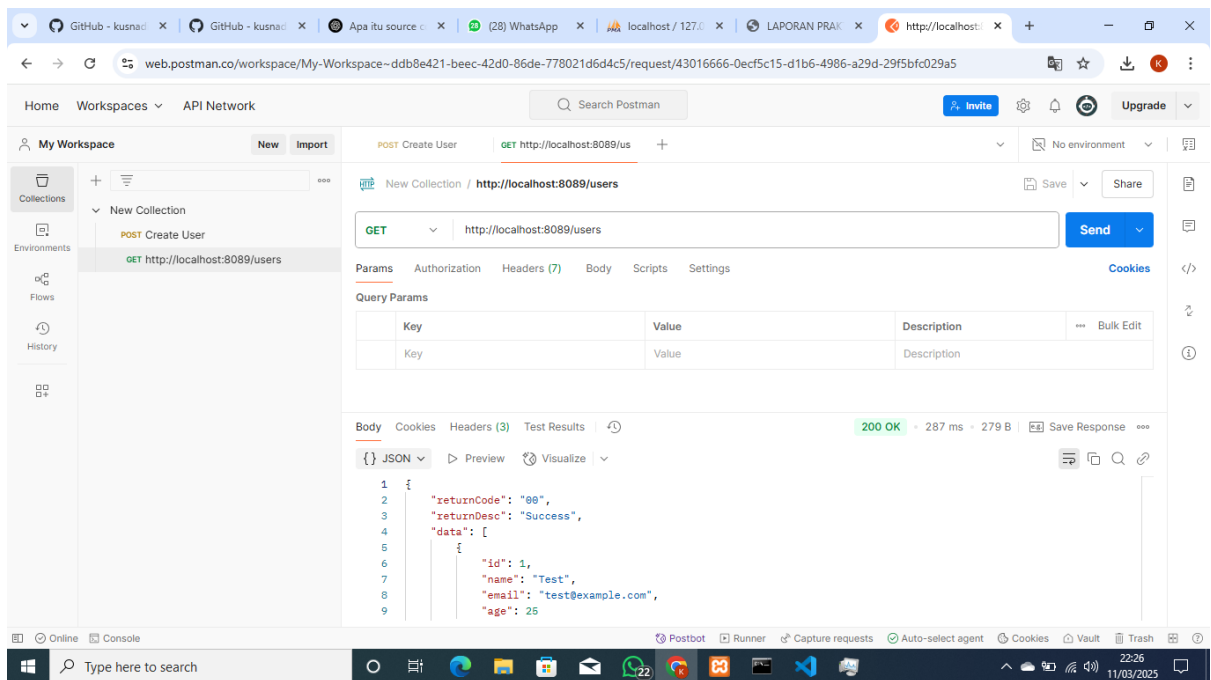
Selanjutnya, pada bagian Body pilih raw dan ketikkan json yang ingin diisikan baik nama, email, age lalu lakukan send untuk melakukan POST API. Jika berhasil maka muncul response JSON dari server.



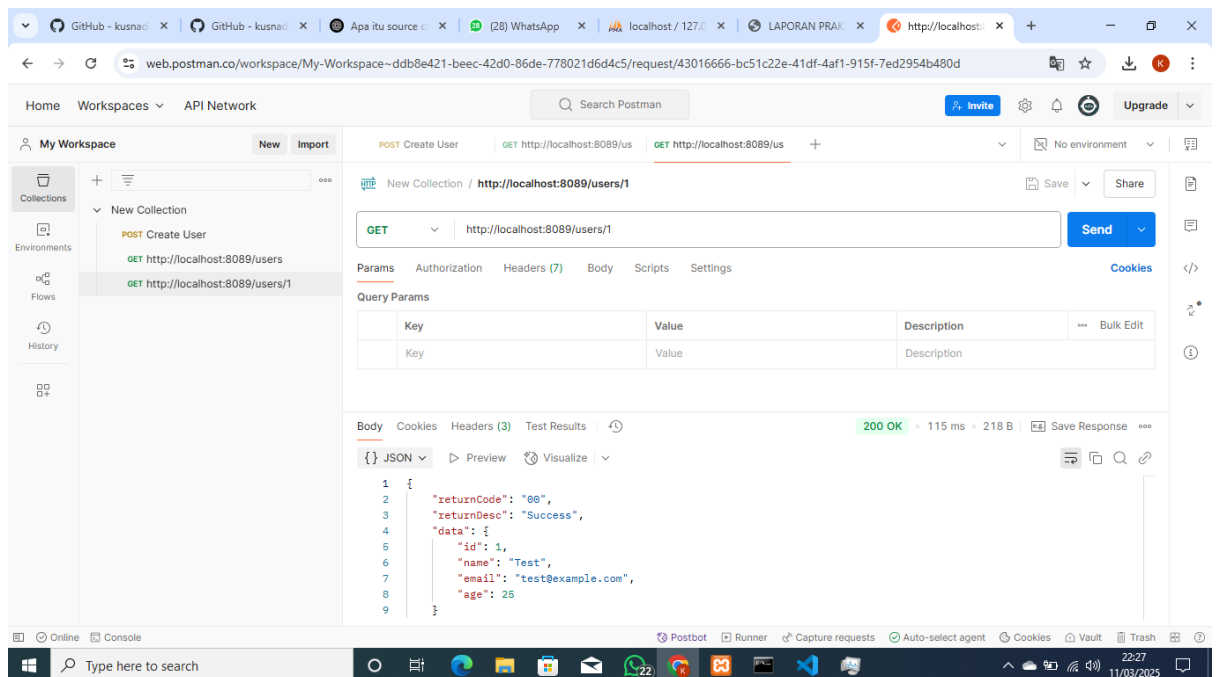
Data berhasil masuk dalam table



9. Kedua Gel All Users, lakukan GET pada postman dengan url <http://localhost:8089/users> untuk GET semua data yang ada. Jika berhasil maka muncul response.

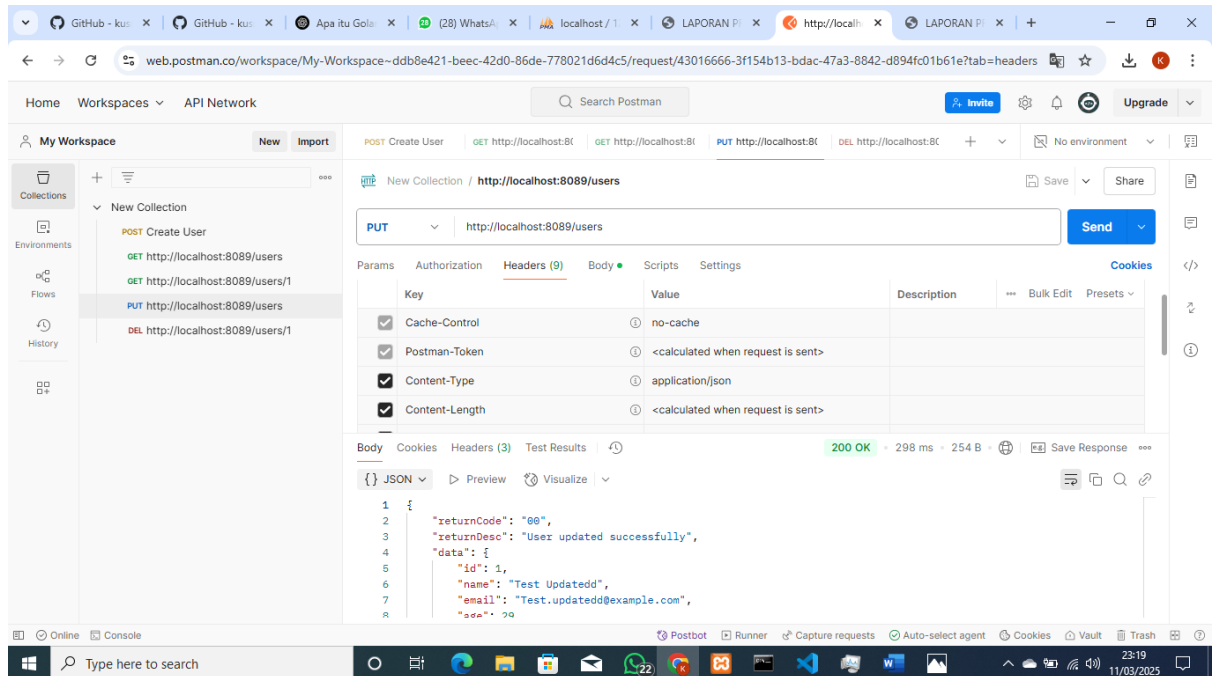


10. Ketiga Get Single, lakukan GET pada postman dengan url <http://localhost:8089/users/1> . Lalu Send untuk GET salah satu data berdasarkan id. Jika berhasil maka akan muncul response.

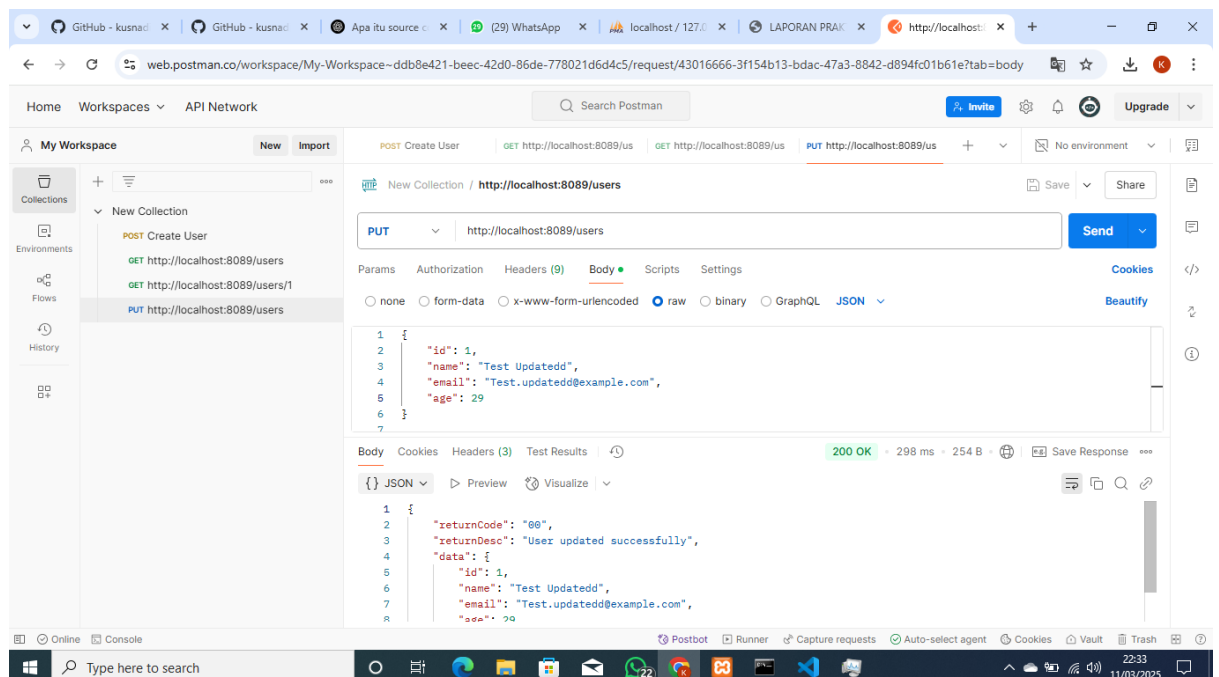




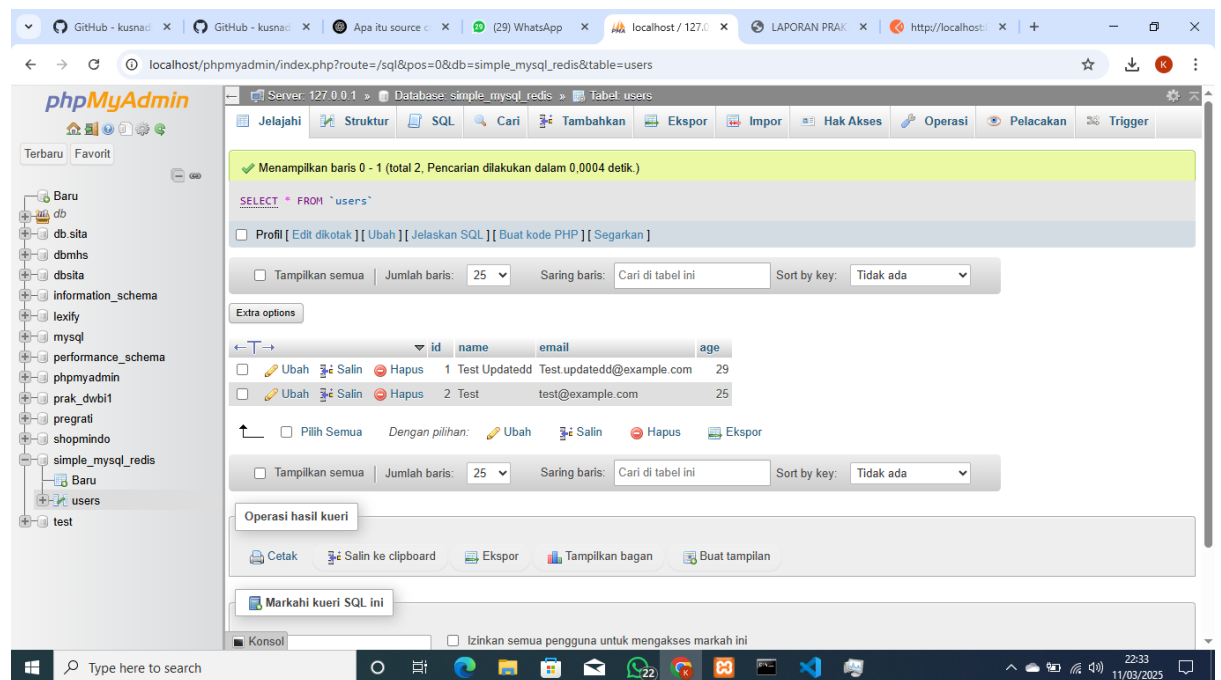
11. Keempat Update User, lakukan PUT pada Postman dengan url <http://localhost:8089/users>. Pada Header isi Key dengan Content-Type dan Values dengan application/json.



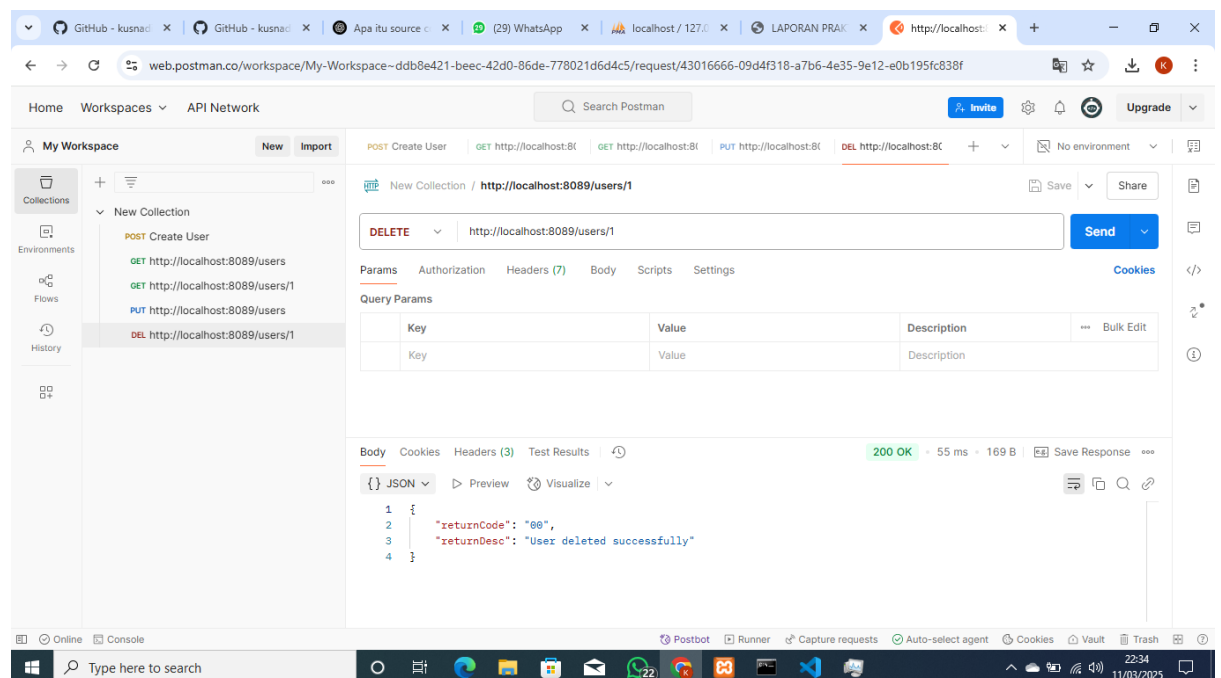
Selanjutnya pada bagian Body pilih raw dan ketikkan json yang ingin diisi dengan baik id, nama, email, dan age yang mau diubah atau diperbaharui. Jika sudah selesai penginputan lakukan send untuk melakukan Update data (PUT). Jika berhasil maka muncul response Json dari server



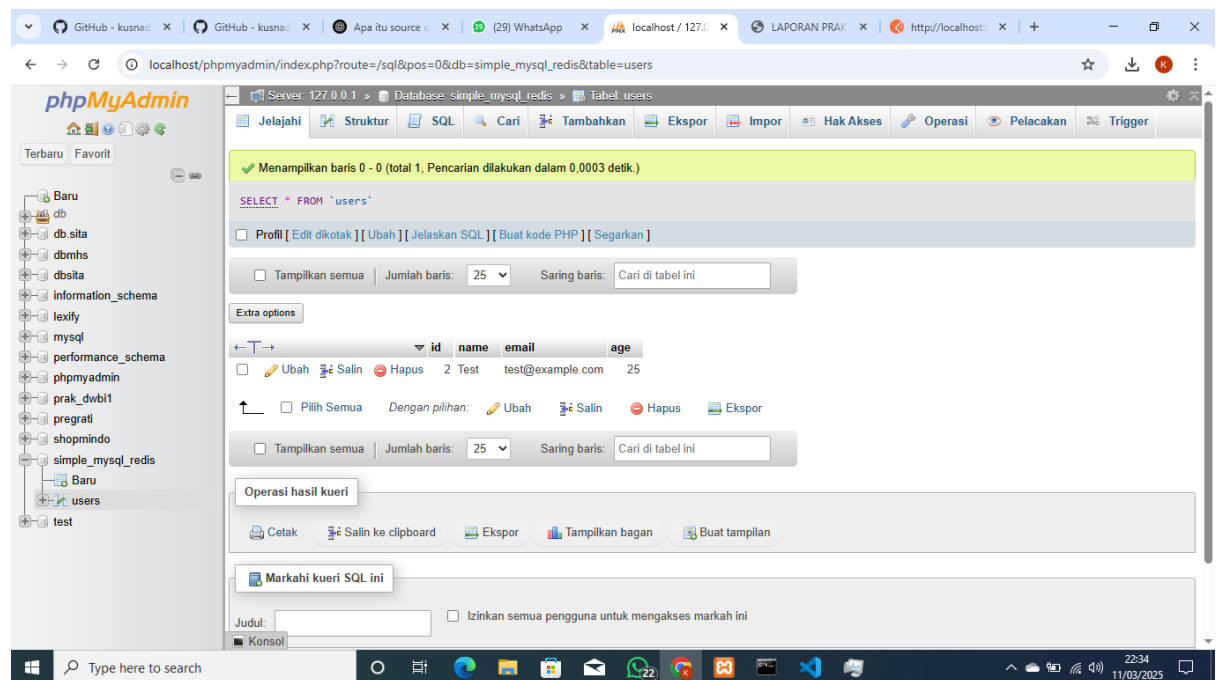
## Data berhasil berubah



12. Selanjutnya Delete User, lakukan Delete pada Postman dengan url <http://localhost:8089/users/1>. Lalu Send untuk menghapus salah satu data berdasarkan id. Jika berhasil maka akan muncul response.



Data berhasil dihapus dari table



13.Semua proses API CRUD berhasil dilakukan

## D. Kesimpulan

Memanfaatkan Redis sebagai caching untuk MySQL dapat secara signifikan meningkatkan efisiensi aplikasi. Dengan mengurangi jumlah query langsung ke database, aplikasi dapat berjalan lebih cepat dan lebih ringan. Pendekatan ini sangat ideal untuk sistem yang memiliki banyak permintaan pembacaan data, seperti dashboard, API dengan traffic tinggi, atau platform analitik.

Namun, pengelolaan cache yang baik tetap diperlukan untuk mencegah terjadinya data usang (stale data). Dengan implementasi yang tepat, integrasi MySQL dan Redis dapat meningkatkan skalabilitas serta kinerja aplikasi secara keseluruhan.