

LAPORAN
Menjalankan CRUD API dengan Golang, MySQL, dan Redis



SEMESTER VI

DISUSUN OLEH:
MUHAMMAD FAHRI

2211083035

DOSEN PENGAMPU:
Yulherniwati, S.Kom.,MT
Yunus Supriadi Wijaya

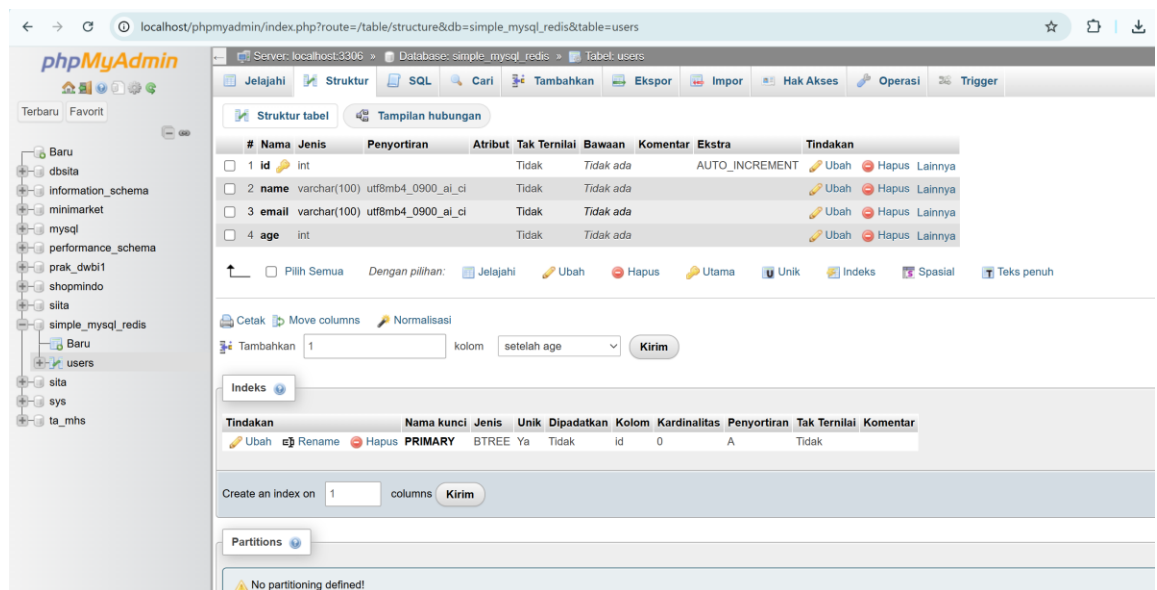
JURUSAN TEKNOLOGI INFORMASI
PRODI TEKNOLOGI REKAYASA PERANGKAT LUNAK
POLITEKNIK NEGERI PADANG
TAHUN 2025

A. Alat

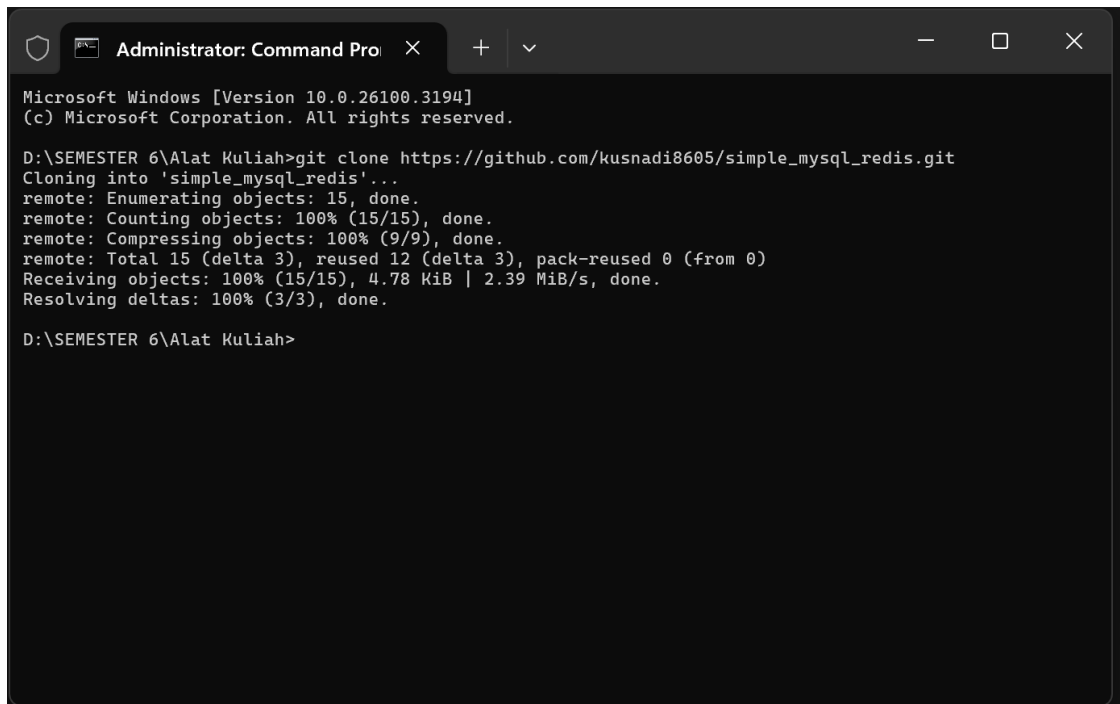
1. Visual Studio Code++
2. Command Prompt
3. Golang
4. MySQL (Laragon/XAMPP)
5. Redis
6. Postman

B. Langkah Kerja

1. Buat database dengan nama simple_mysql_redis. Dan tabel users



2. Lakukan klon repository dari github.



```
Administrator: Command Pro
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

D:\SEMESTER 6\Alat Kuliah>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 | 2.39 MiB/s, done.
Resolving deltas: 100% (3/3), done.

D:\SEMESTER 6\Alat Kuliah>
```

3. Konfigurasi database dan redis agar bisa saling terhubung pada file main.go. Pada dbConfig hapus atau kosongkan password. Pada konfigurasi Redis gunakan IP Onlinenya yaitu 8.215.73.55 menggantikan localhost.

```
// Main function
func main() {
    // Database configuration
    dbConfig := DBConfig{
        User:      "root",
        Password:  "",
        Host:      "127.0.0.1",
        Port:      3306,
        DBName:    "simple_mysql_redis",
    }

    // Redis configuration
    redisOption := redis.Options{
        Addr:      "8.215.73.55:6379",
        Password:  "",
        DB:        0,
    }
```

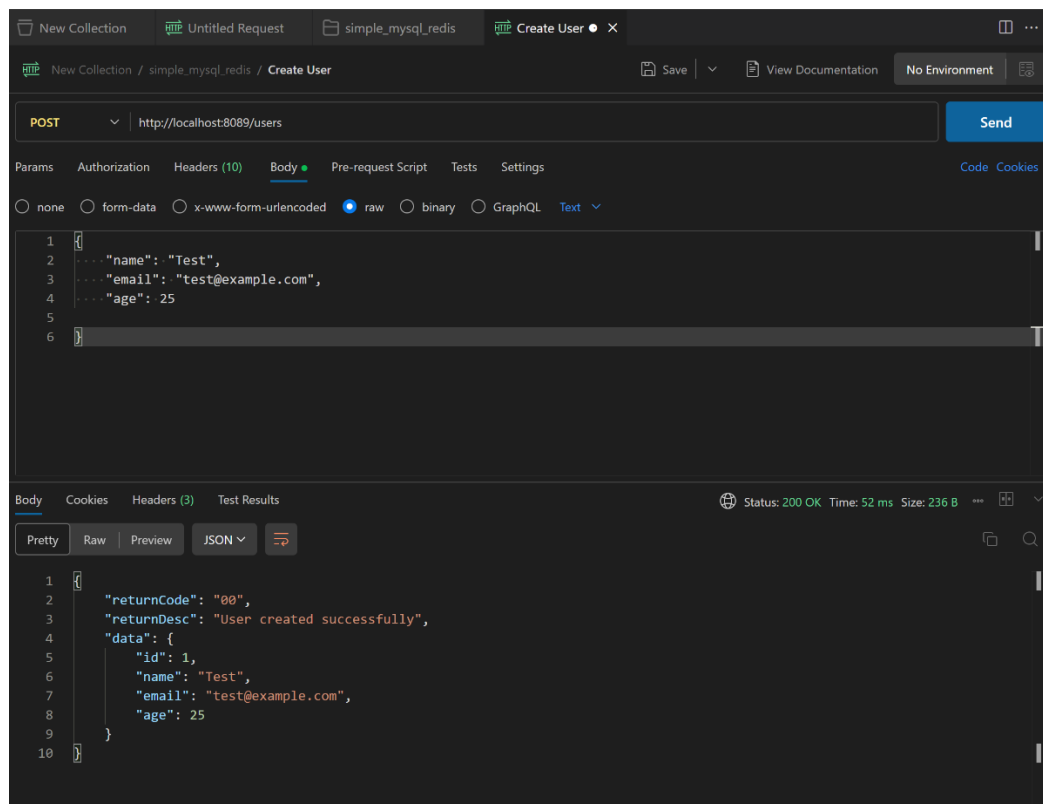
4. Jalankan server.

```
PS C:\Users\ACER> cd "D:\SEMESTER 6\Alat Kuliah\simple_mysql_redis"
PS D:\SEMESTER 6\Alat Kuliah\simple_mysql_redis> go run .
Database connection successful!
Redis connection successful!
Server is running on port 8080

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

-----o/-----
      o\
⇒ http server started on [::]:8089
```

5. Selanjutnya menjalankan CRUD API dengan Postman.
6. 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**, pada bagian Body pilih raw dan ketikan json yang ingin diisikan baik nama, email, age lalu lakukan send untuk lakukan POST API. Jika berhasil maka muncul response JSON dari server.



Data berhasil masuk ke tabel.

Server: localhost:3306 » Database: simple_mysql_redis » Tabel: users

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses

✓ Menampilkan baris 0 - 0 (total 1, Pencarian dilakukan dalam 0,0010 detik.)

`SELECT * FROM `users``

☐ Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

☐ Tampilkan semua | Jumlah baris: 25 ▾ | Saring baris: Cari di tabel ini

Extra options

	id	name	email	age
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1	Test	test@example.com	25

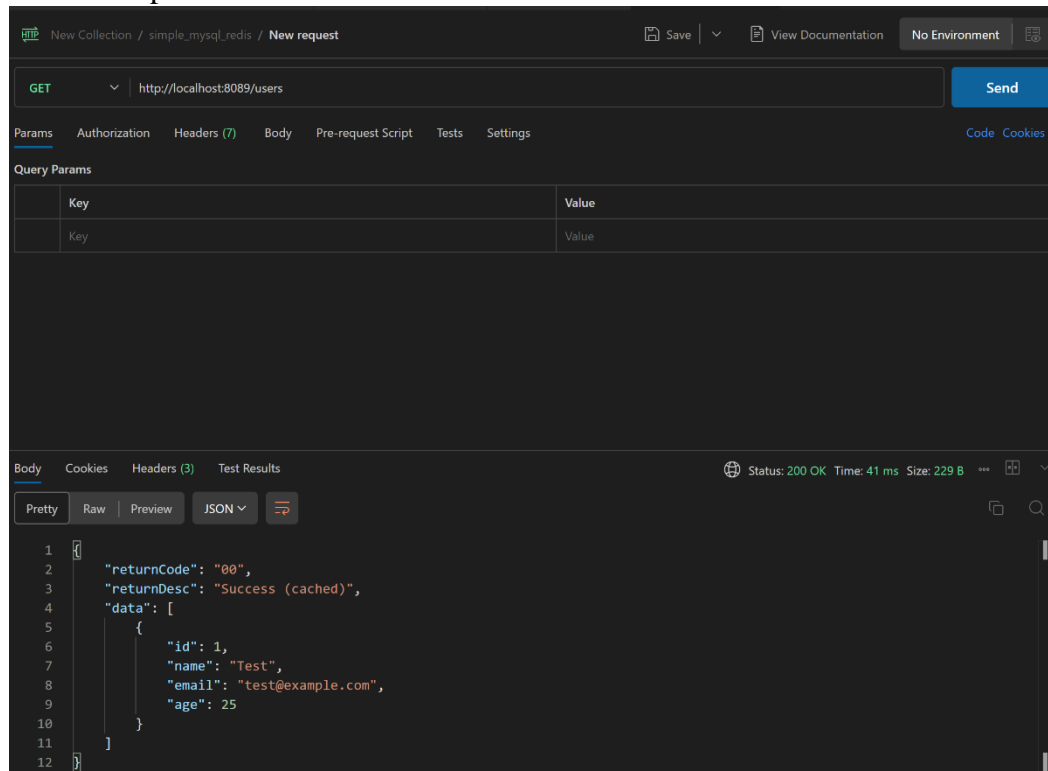
↑ ☐ Pilih Semua Dengan pilihan: ☐ Ubah ☐ Salin ☐ Hapus ☐ Ekspor

☐ Tampilkan semua | Jumlah baris: 25 ▾ | Saring baris: Cari di tabel ini

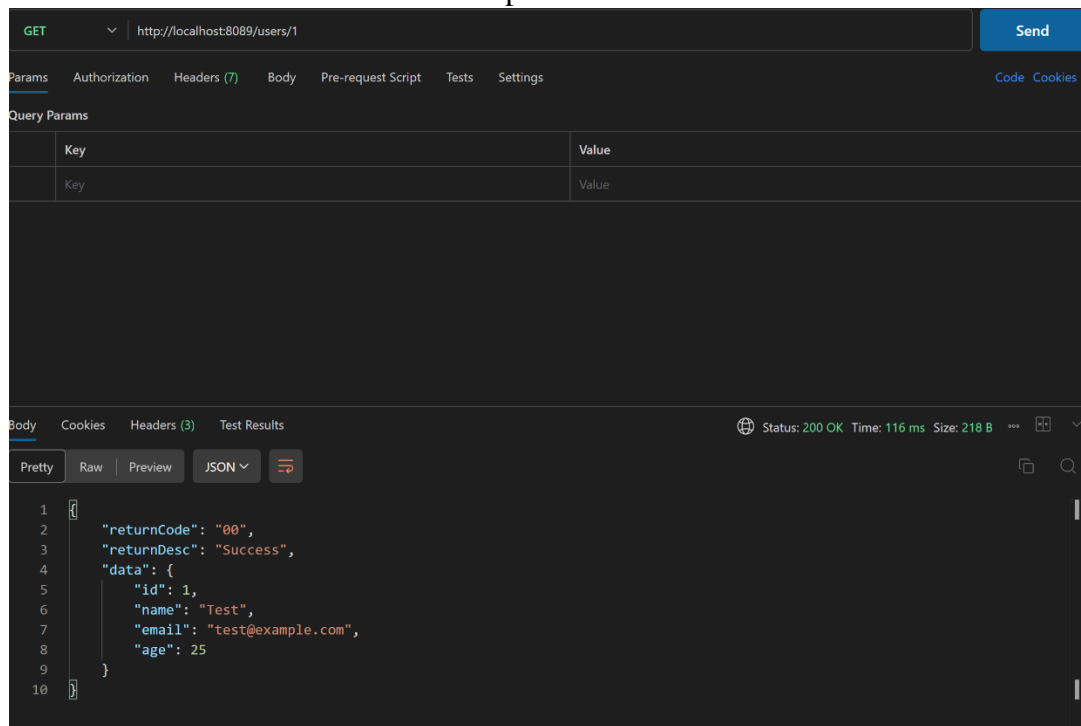
Operasi hasil kueri

☐ Cetak ☐ Salin ke clipboard ☐ Ekspor ☐ Tampilkan bagan ☐ Buat tampilan

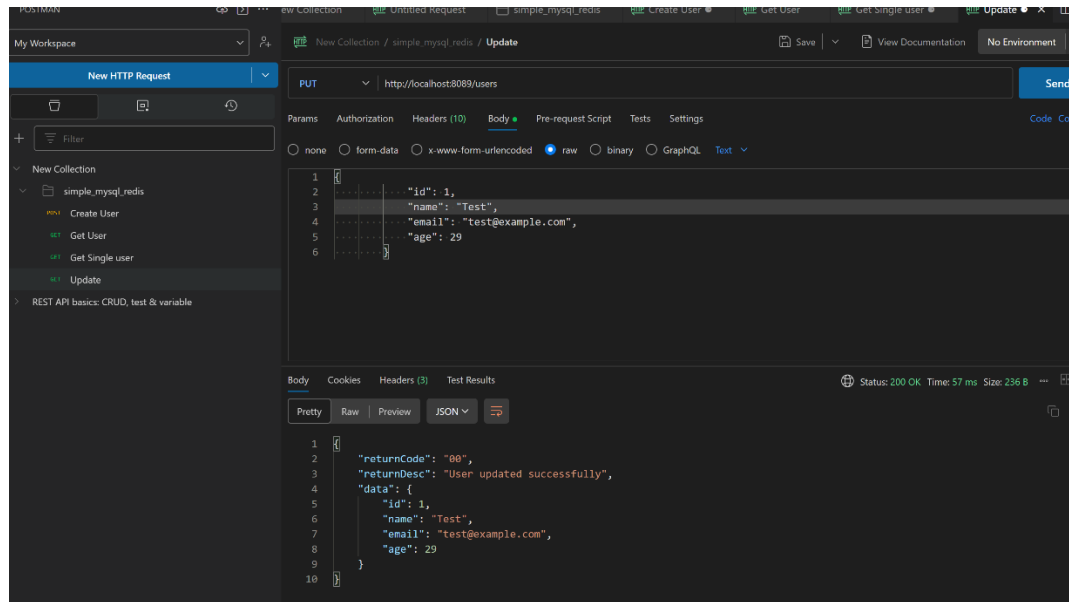
7. Kedua **Get All Users**, lakukan **GET** pada Postman dengan url <http://localhost:8089/users> untuk **GET** semua data yang ada. Jika berhasil maka muncul response.



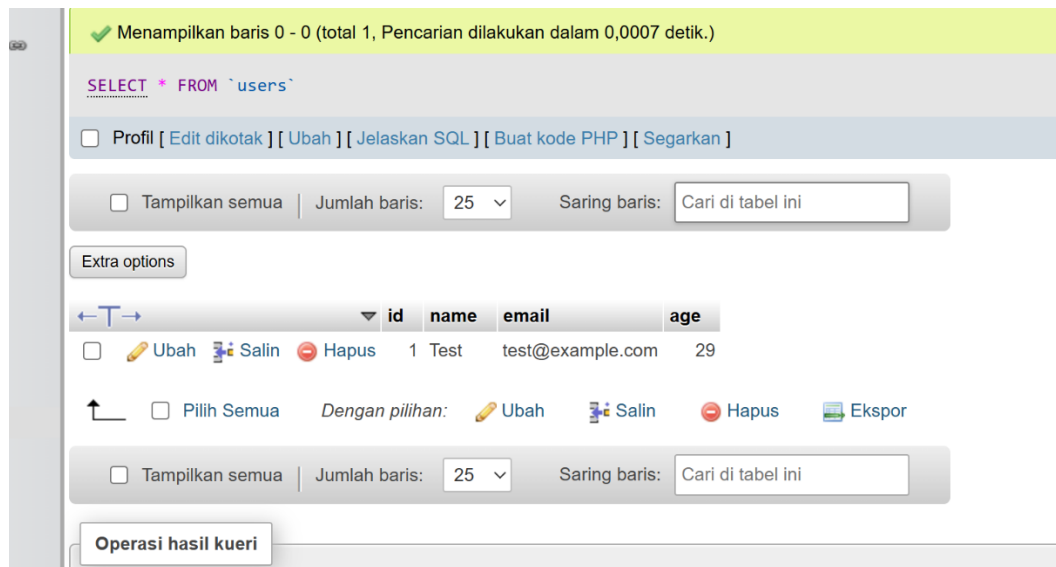
8. Ketiga **Get Single User**, lakukan **GET** pada Postman dengan url <http://localhost:8089/users/1> lalu Send untuk **GET** salah satu data berdasarkan id. Jika berhasil maka akan muncul response.



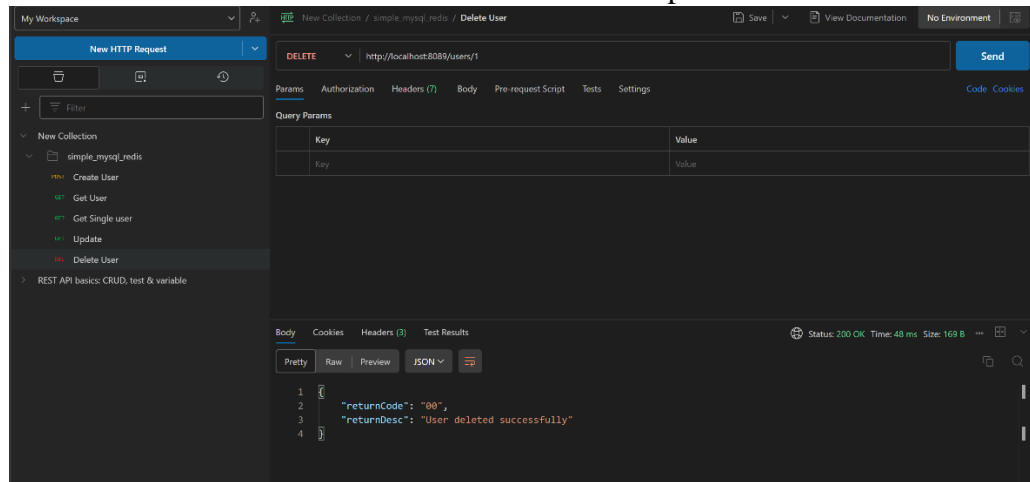
9. Keempat **Update User**, lakukan PUT pada Postman dengan url <http://localhost:8089/users>. Pada **Header** isi **Key** dengan **Content-Type** dan **Values** dengan **application/json**. Selanjutnya pada bagian **Body** pilih **raw** dan ketikkan json yang ingin diisikan baik id, nama, email, age yang mau diubah atau diperbaharui. Jika sudah selesai penginputan lakukan send untuk lakukan Update data (Put). Jika berhasil maka muncul response JSON dari server.



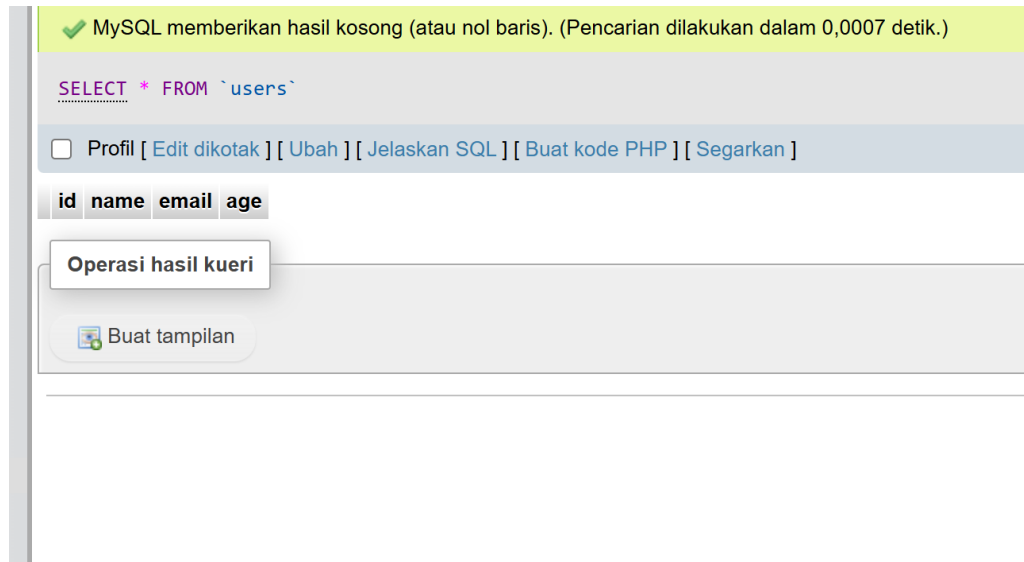
Data berhasil diubah.



10. 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 tabel.



11. Semua proses API CRUD berhasil dilakukan.