

Dokumen Teknis

Education App for Monitoring Presence and Environment: SMAN 1 Parmaksian

**Tugas Proyek Akhir Semester
Mata Kuliah: Pengembangan Aplikasi Terdistribusi**

Dipersiapkan oleh:

NIM 11322006	Anastasya C. Marpaung
NIM 11322022	Keren Simanjuntak
NIM 11322036	Christian Y. Gultom
NIM 11322054	Maudy Octavia S

**Untuk:
Institut Teknologi Del
2024**



**PROYEK PENGEMBANGAN APLIKASI TERDISTRIBUSI
INSTITUT TEKNOLOGI DEL 2024**

1. Pendahuluan

Pada bab 1 berisi tentang deskripsi umum aplikasi, karakteristik pengguna aplikasi, dan fungsi pada aplikasi.

1.1 Deskripsi Umum Aplikasi

Pembangunan sistem ini ditujukan untuk sistem monitoring dan absensi . Aplikasi ini dapat digunakan oleh guru dan siswa untuk melakukan monitoring. Pada aplikasi ini, user dapat melakukan absensi sedangkan, admin dapat mengelola data siswa, pengumuman, jadwal pembelajaran (roster) , pengajar, kelas, tanya jawab. Untuk dapat mengakses sistem, setiap pengguna harus melakukan login terlebih dahulu

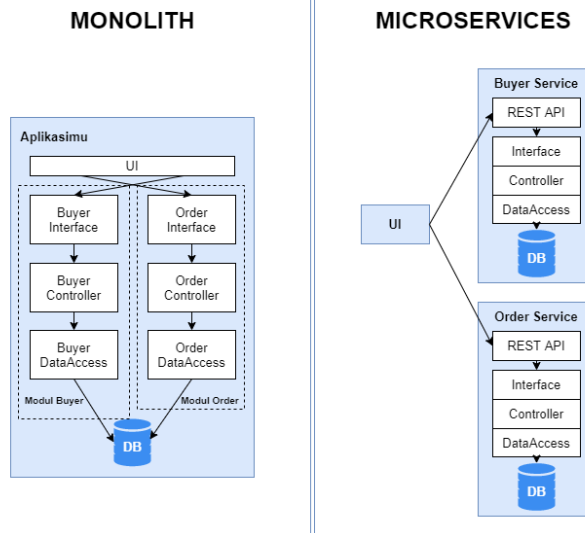
Sistem ini dibangun dengan menggunakan 3 bahasa pemrograman. Bagian *back-end* dan *frontend* dibangun dengan bahasa yang berbeda. Untuk bagian *back-end* digunakan bahasa *Go* dan untuk bagian *front-end* menggunakan bahasa *JavaScript*, *Dart*.

Teknologi yang digunakan dalam pembangunan sistem ini adalah *Flutter*, *React.js*, *Apache*(*Apache HTTP Service*), dan *Fiber* :

1. *Flutter* berfungsi sebagai *framework* yang digunakan untuk membangun aplikasi mobile, *flutter* menggunakan bahasa pemrograman *dart*.
2. *React.js* berfungsi sebagai *library* untuk membangun antarmuka pengguna. *React.js* menggunakan bahasa pemrograman *JavaScript*.
3. *Apache HTTP Server* adalah web server yang digunakan untuk melayani permintaan HTTP dan mengirimkan konten web kepada pengguna berdasarkan permintaan.
4. *Fiber* berfungsi sebagai *framework* untuk bahasa pemrograman *Go*(*golang*).

Tools yang digunakan dalam pembangunan sistem ini adalah :

1. Teks Editor/IDE : *Visual Studio Code*
2. Frontend Development : *Flutter*, *React.js*
3. Backend Development : *Fiber*
4. gVersion Control System : *GitHub*
5. Database Management System : *MySQL*
6. API Testing : *Postman*
7. Development Environment : *XAMPP*



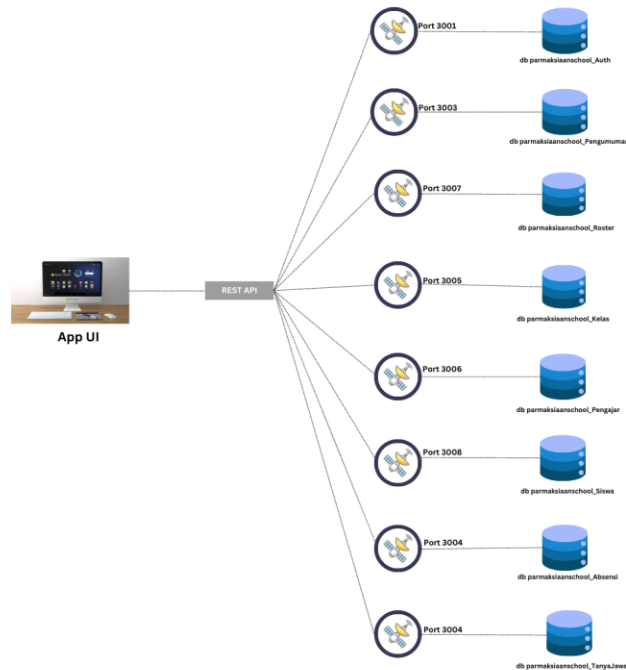
Gambar 1 Perbedaan Monolith dan Microservices

Arsitektur microservice adalah pendekatan desain perangkat lunak di mana aplikasi dibagi menjadi layanan-layanan kecil yang berjalan secara independen. Misalnya, dalam sebuah aplikasi terdapat layanan-layanan seperti User Service, Product Service, dan Order Service. Setiap layanan memiliki database sendiri dan berkomunikasi melalui protokol REST API. Hal ini memungkinkan setiap layanan berjalan secara independen, sehingga jika satu layanan mengalami gangguan, layanan lainnya tetap dapat beroperasi. Berbeda dengan arsitektur monolitik di mana seluruh fungsi aplikasi dibangun dalam satu kode sumber, dalam arsitektur microservice, setiap fungsi aplikasi dipecah menjadi layanan terpisah. Setiap layanan fokus pada satu fungsi spesifik dan berkomunikasi melalui protokol komunikasi ringan.

Komunikasi antar modul dalam monolitik biasanya dilakukan melalui pemanggilan fungsi di dalam kode, sedangkan dalam microservice, komunikasi antar layanan memerlukan protokol komunikasi eksternal yang ringan.

Deployment dalam monolitik cenderung lambat karena seluruh kode perlu didistribusikan dan di-deploy sekaligus. Di sisi lain, dalam arsitektur microservice, deployment dapat dilakukan secara terpisah untuk setiap layanan, membuatnya lebih fleksibel dan cepat.

Jadi, arsitektur microservice memungkinkan pengembangan yang lebih fleksibel, skalabilitas yang lebih baik, dan penanganan kesalahan yang lebih efisien dibandingkan dengan arsitektur monolitik.



Gambar 2 Arsitektur Microservice

Pada Gambar 1 terlampir arsitektur microservice yang digunakan pada pembangunan sistem Pendidikan. REST API digunakan sebagai penghubung antara *App UI* dengan *service* yang tersedia sehingga memungkinkan *service* dapat dikonsumsi oleh *App UI*. Hal inilah yang menyebabkan *service* dapat diakses oleh beberapa *platform* tanpa harus terbatas pada satu bahasa pemrograman saja. Sama halnya dengan pembangunan aplikasi web jual beli ini. Bahasa yang digunakan pada bagian *front-end* adalah bahasa Javascript dan pada bagian *backend* menggunakan bahasa Go. Meskipun memiliki bahasa yang berbeda, tetapi aplikasi akan tetap dapat berjalan dengan adanya REST API.

Pada bagian *back-end*, setiap *service* memiliki *database*-nya masing-masing. Pada sistem ini terdapat delapan *service* dan satu *database*. *Service* yang tersedia adalah *Auth Service* dengan *database* *parmaksianschool_Auth*, *Pengumuman Service* dengan *database* *parmaksianschool_pengumuman*, *Roster Service* dengan *database* *parmaksianschool_roster*, *Kelas Service* dengan *database* *parmaksianschool_kelas*, *Pengajar Service* dengan *parmaksianschool_pengajar*, *Siswa Service* dengan *parmaksianschool_siswa*, *Absensi Service* dengan *parmaksianschool_absensi*, dan *TanyaJawab Service* dengan

parmaksianschool_TanyaJawab. Setiap *service* dirancang memiliki *port* yang berbeda sehingga *port* tidak akan bertabrakan.

1.2 Karakteristik Pengguna Aplikasi

Pada sistem ini, terdapat dua pengguna. Pengguna pada sistem ini adalah guru (*admin*) dan siswa (*user*). Karakteristik pengguna dapat dilihat pada Tabel 1.

Tabel 1 Karakteristik Pengguna Aplikasi

Kategori Pengguna	Fungsi	Hak Akses ke Aplikasi
<i>Admin</i>	Mengelola data dalam aplikasi	<ol style="list-style-type: none"> 1. Akses ke menu <i>login</i> 2. Akses ke menu <i>logout</i> 3. Akses ke fungsi menjawab dan menghapus pertanyaan 4. Akses ke fungsi menambah, melihat, memperbaharui, dan menghapus pengumuman 5. Akses ke fungsi menambah, melihat, memperbaharui, dan menghapus kelas 6. Akses ke fungsi menambah, melihat, memperbaharui, dan menghapus pengajar 7. Akses ke fungsi menambah, melihat, memperbaharui, dan menghapus roster 8. Akses ke fungsi menambah, melihat, memperbaharui, dan menghapus siswa 9. Akses ke fungsi melihat absens

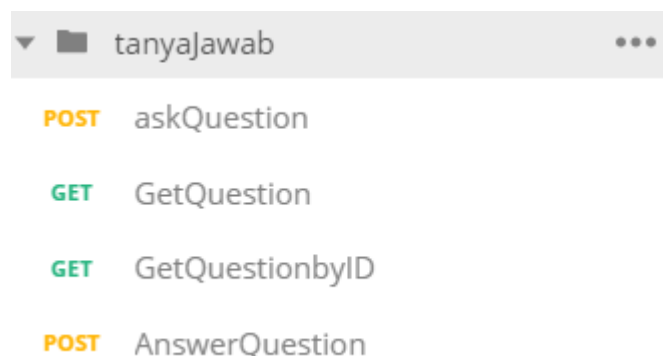
<i>User</i>	Melakukan absensi	<ol style="list-style-type: none"> 1. Akses ke menu login' 2. Akses ke menu logout 3. Akses ke menu menambah absensi 4. Akses ke fungsi mengajukan pertanyaan 5. Akses ke fungsi melihat pengumuman 6. Akses ke fungsi melihat kelas 7. Akses ke fungsi melihat pengajar 8. Akses ke fungsi melihat roster 9. Akses ke fungsi melihat data siswa
-------------	-------------------	---

1.3 Fungsi pada Aplikasi

Adapun fungsi yang terdapat pada aplikasi jual beli barang ini, yaitu:

1. Fungsi Mengelola Tanya Jawab

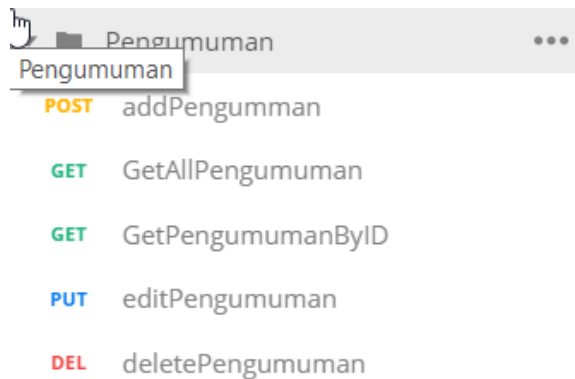
Fungsi mengelola tanya jawab digunakan oleh admin dan user. Fungsi tanya jawab terdiri dari 2 *method* yaitu *post* dan *get*. *Method post* digunakan untuk membuat pertanyaan (*askQuestion*) dan menjawab pertanyaan (*AnswerQuestion*). *Method get* digunakan untuk menampilkan semua pertanyaan (*GetQuestion*) dan menampilkan pertanyaan sesuai ID (*GetQuestionByID*).



Gambar 3 Method Mengelola Tanya Jawab

2. Fungsi Mengelola Pengumuman

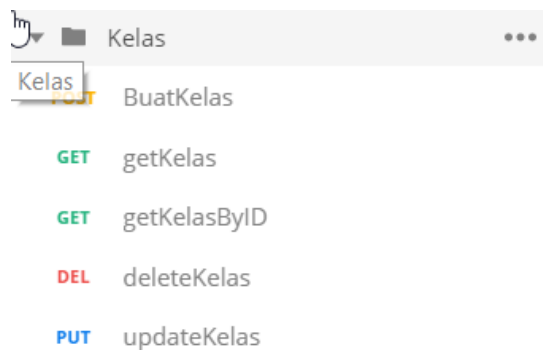
Fungsi mengelola pengumuman digunakan oleh admin dan user. Fungsi mengelola pengumuman terdiri dari 4 *method* yaitu *post*, *get*, dan *delete*. *Method post* digunakan untuk menambah pengumuman (*addPengumuman*). *Method get* digunakan untuk menampilkan semua pengumuman (*GetAllPengumuman*) dan menampilkan pengumuman berdasarkan ID (*GetPengumumanByID*). *Method put* digunakan untuk melakukan perubahan data pengumuman. *Method delete* digunakan untuk menghapus data pengumuman (*deletePengumuman*).



Gambar 4 Method Mengelola Pengumuman

3. Fungsi Mengelola Kelas

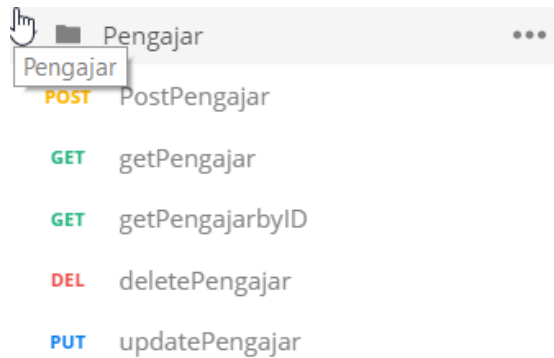
Fungsi mengelola kelas digunakan oleh admin dan user. Fungsi mengelola kelas terdiri dari 4 *method* yaitu *post*, *get*, *put* dan *delete*. *Method post* digunakan untuk menambah kelas (*BuatKelas*). *Method get* digunakan untuk menampilkan semua kelas (*getKelas*) dan menampilkan kelas berdasarkan ID (*getKelasByID*). *Method put* digunakan untuk melakukan perubahan data kelas (*updateKelas*). *Method delete* digunakan untuk menghapus data kelas (*deleteKelas*).



Gambar 5 Method Mengelola Kelas

4. Fungsi Mengelola Pengajar

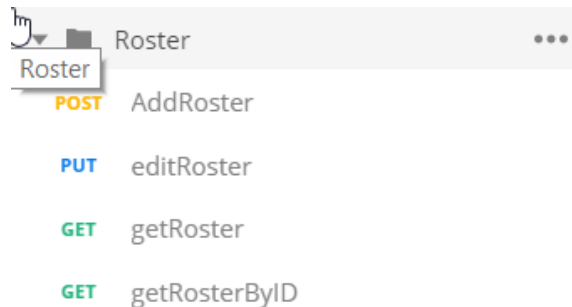
Fungsi mengelola pengajar digunakan oleh admin dan user. Fungsi mengelola pengajar terdiri dari 4 *method* yaitu *post*, *get*, *put*, dan *delete*. *Method post* digunakan untuk menambah pengajar (*PostPengajar*). *Method get* digunakan untuk menampilkan semua pengajar (*getPengajar*) dan menampilkan pengajar berdasarkan ID (*getPengajarByID*). *Method put* digunakan untuk melakukan perubahan data pengajar (*updatePengajar*). *Method delete* digunakan untuk menghapus data pengajar (*deletePengajar*).



Gambar 6 Method Mengelola Pengajar

5. Fungsi Mengelola Roster

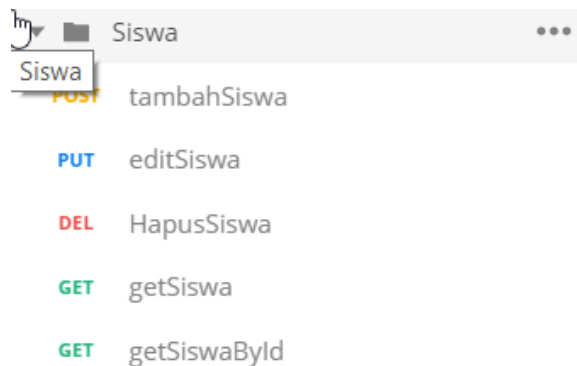
Fungsi mengelola roster digunakan oleh admin dan user. Fungsi mengelola roster terdiri dari 4 *method* yaitu *post*, *get*, dan *put*. *Method post* digunakan untuk menambah roster (*AddRoster*). *Method get* digunakan untuk menampilkan semua roster (*getRoster*) dan menampilkan roster berdasarkan ID (*getRosterByID*). *Method put* digunakan untuk melakukan perubahan data roster (*editRoster*).



Gambar 7 Method Mengelola Roster

6. Fungsi Mengelola Siswa

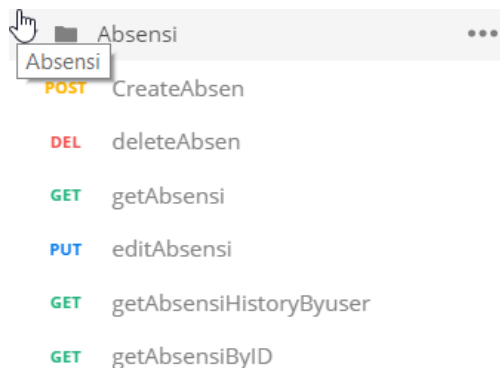
Fungsi mengelola siswa digunakan oleh admin dan user. Fungsi mengelola siswa terdiri dari 4 *method* yaitu *post*, *get*, *put*, dan *delete*. *Method post* digunakan untuk menambah siswa (*tambahSiswa*). *Method get* digunakan untuk menampilkan semua siswa (*getSiswa*) dan menampilkan siswa berdasarkan ID (*getSiswaByID*). *Method put* digunakan untuk melakukan perubahan data siswa (*editSiswa*). *Method delete* digunakan untuk menghapus data siswa (*HapusSiswa*).



Gambar 8 Method Mengelola Siswa

7. Fungsi Mengelola Absensi

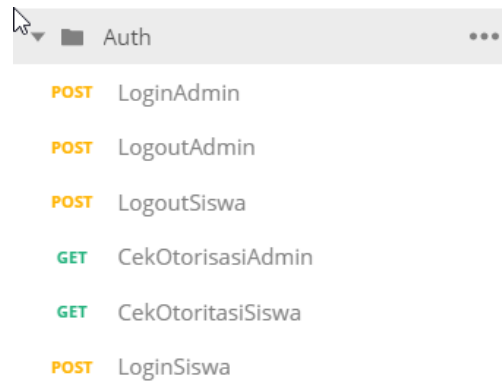
Fungsi mengelola absensi digunakan oleh admin dan user. Fungsi mengelola absensi terdiri dari 4 *method* yaitu *post*, *delete*, *get*, dan *put*. *Method post* digunakan untuk membuat absen atau menambahkan absen (*CreateAbsen*). *Method delete* digunakan untuk menghapus absensi (*deleteAbsen*). *Method get* digunakan untuk menampilkan daftar semua absen (*getAbsensi*), menampilkan daftar riwayat absensi (*getAbsensiHistoryByUser*), menampilkan daftar absensi berdasarkan ID (*getAbsensiByID*). *Method put* digunakan untuk melakukan perubahan (*editAbsensi*).



Gambar 9 Method Mengelola Absensi

8. Fungsi Autentikasi

Fungsi autentikasi dapat dilakukan oleh admin dan user. Fungsi autentikasi terdiri dari 2 *method* yaitu *post* dan *get*. *Method post* digunakan untuk melakukan login (*LoginAdmin*, *LoginSiswa*) , melakukan logout (*LogoutAdmin*, *LogoutSiswa*). *method get* digunakan untuk mengecek user sudah login atau tidak (*CekOtorisasiAdmin*, *CekOtorisasiSiswa*).

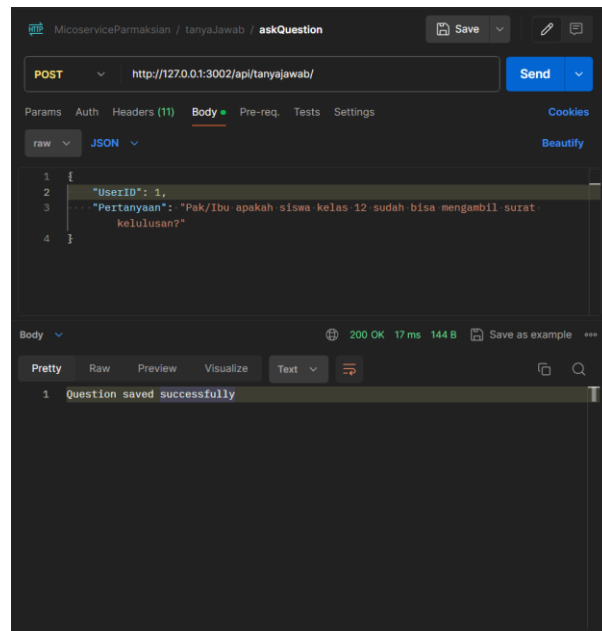


Gambar 10 Method Autentikasi

2. Rancangan Aplikasi Microservices

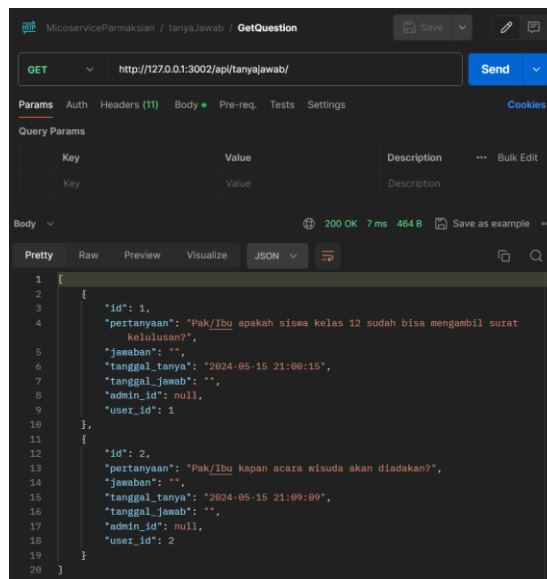
2.1.1 Tanya jawab

a. POST: AskQuestion



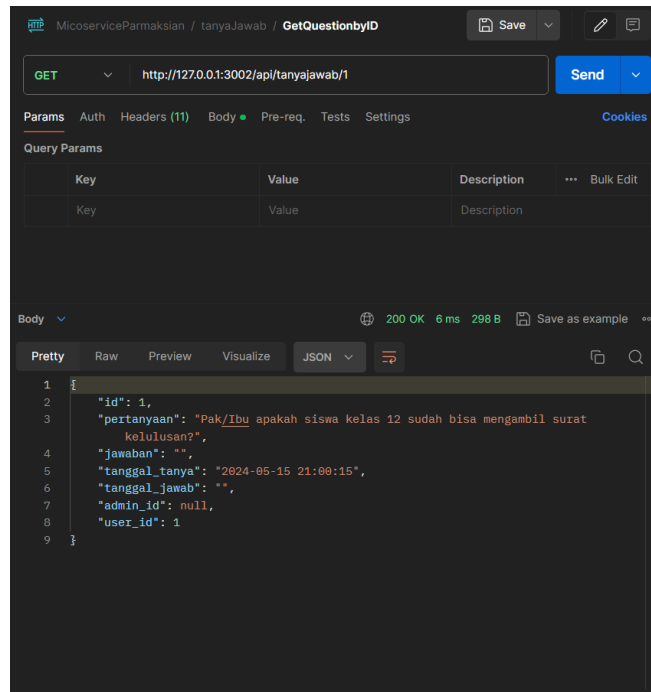
Gambar 11 Post: AskQuestion

b. GET: GetQuestion



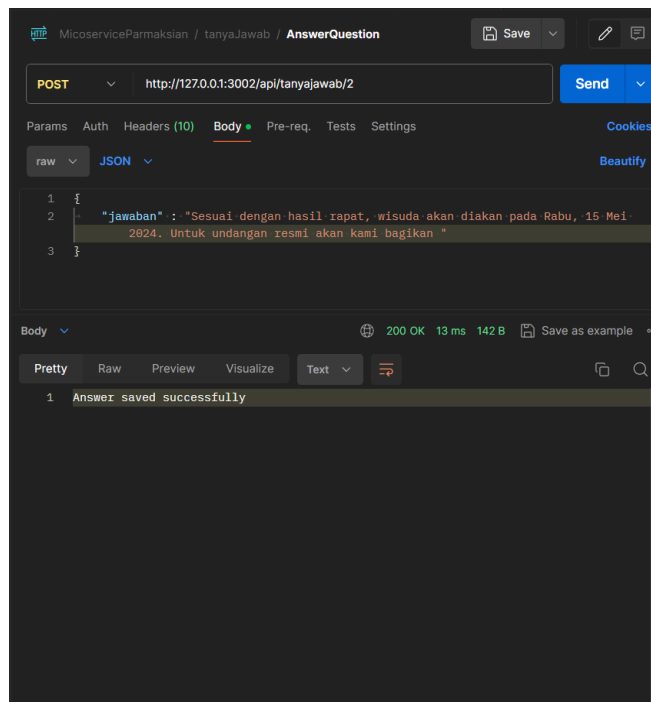
Gambar 12 GET: GetQuestion

c. GET: GetQuestionByID



Gambar 12 GET: GetQuestionByID

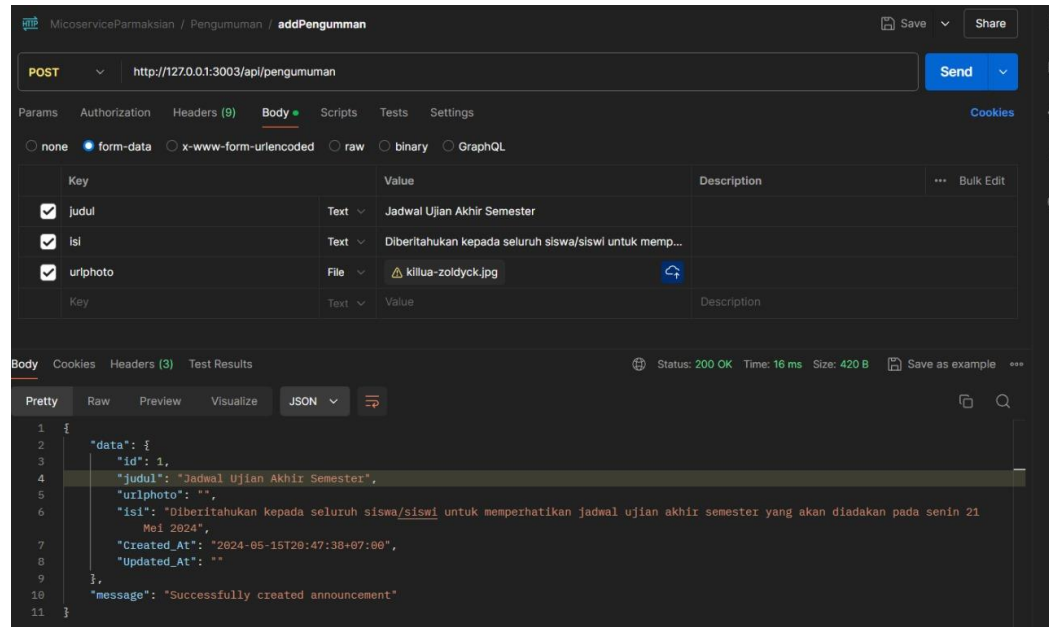
d. POST: AnswerQuestion



Gambar 13 POST: AnswerQuestion

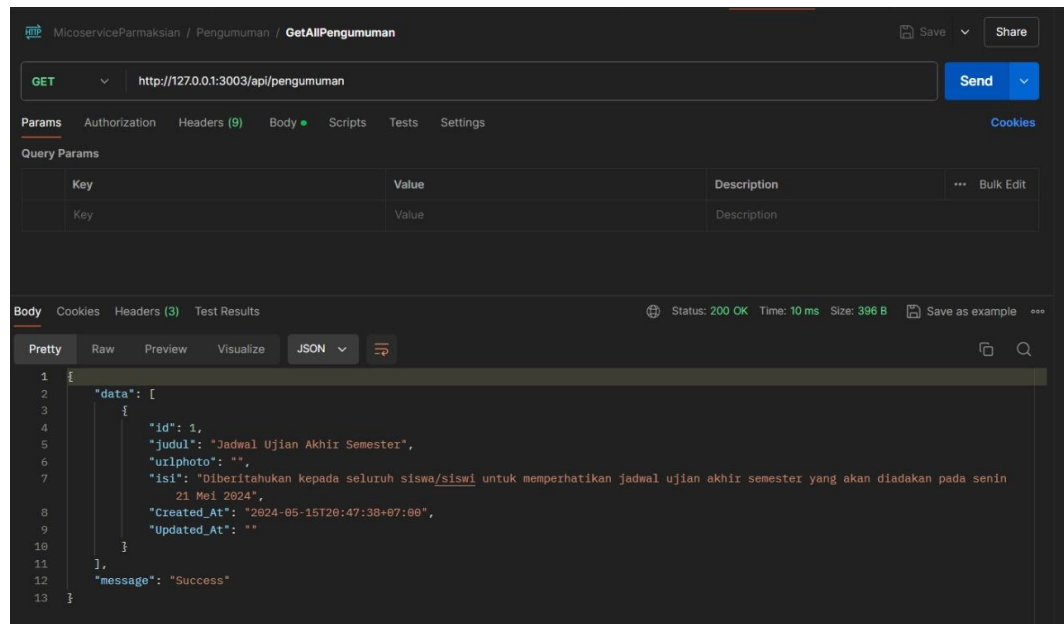
2.2.2 Pengumuman

a. POST: addPengumuman



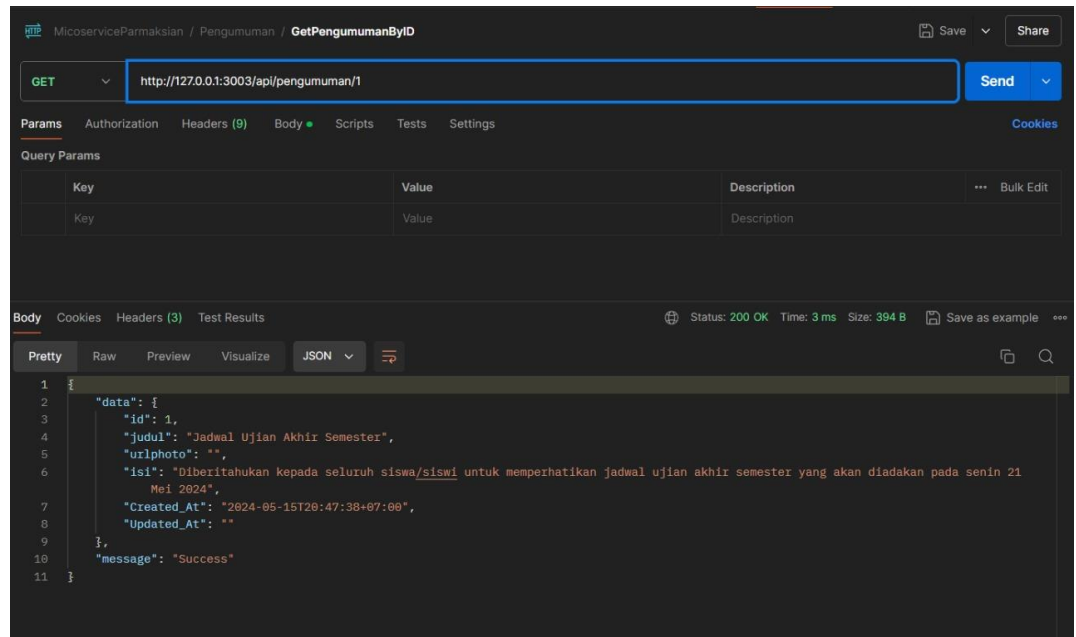
Gambar 14 POST: addPengumuman

b. GET: GetAllPengumuman



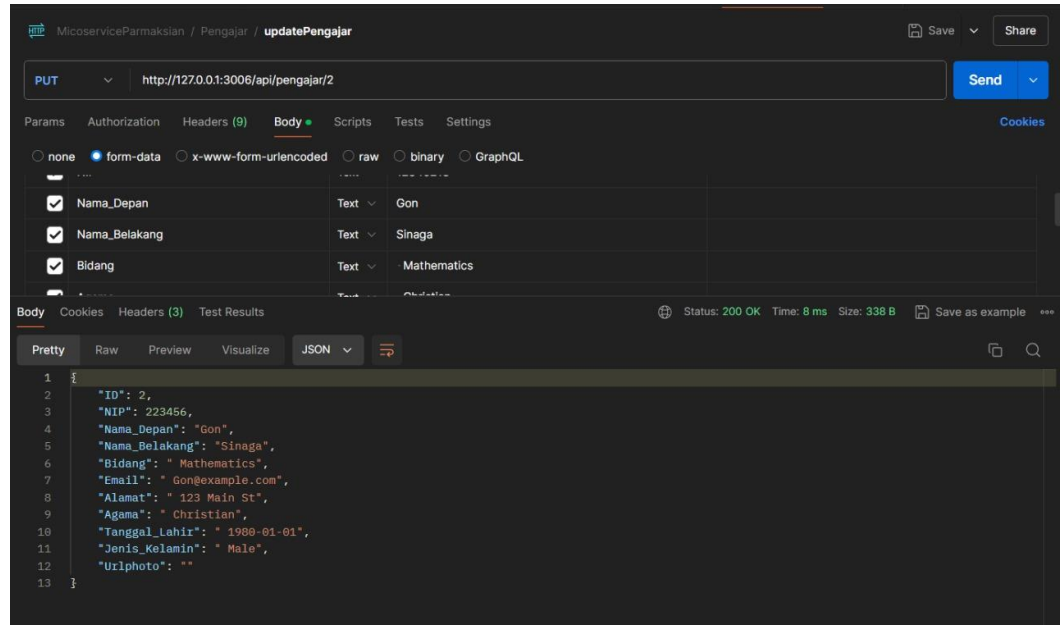
Gambar 15 GET: GetAllPengumuman

c. GET: GetPengumumanByID



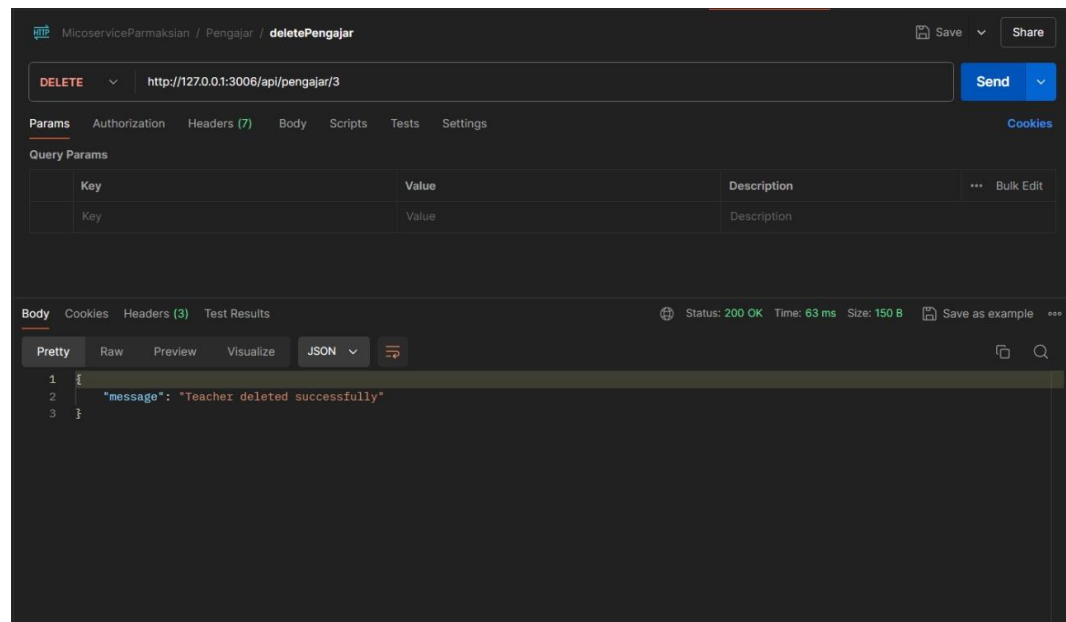
Gambar 16 GET: GetPengumumanByID

d. PUT: editPengumuman



Gambar 17 PUT: editPengumuman

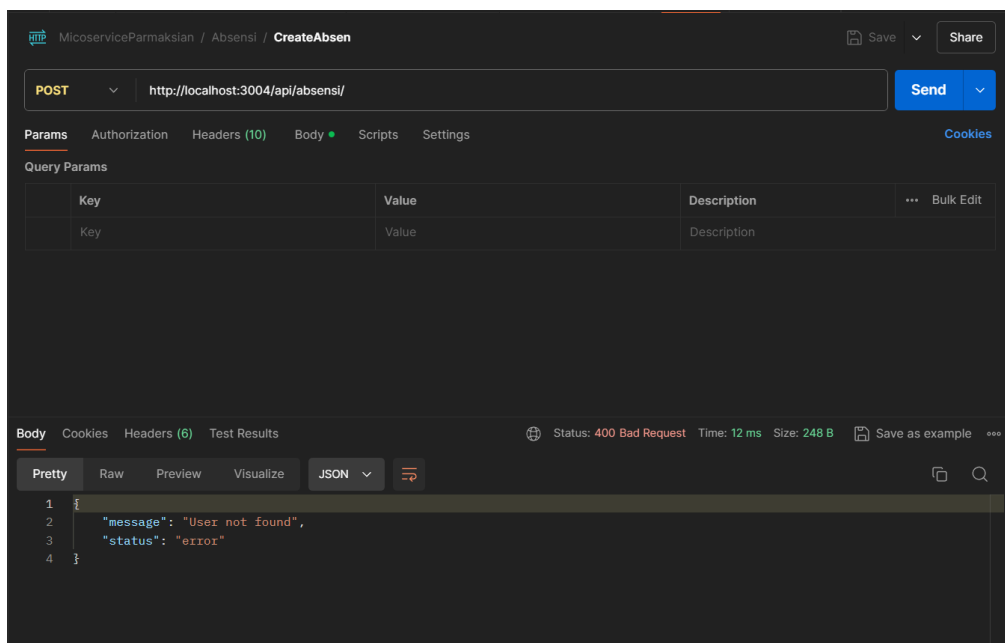
e. DELETE: deletePengumuman



Gambar 18 DELETE: deletePengumuman

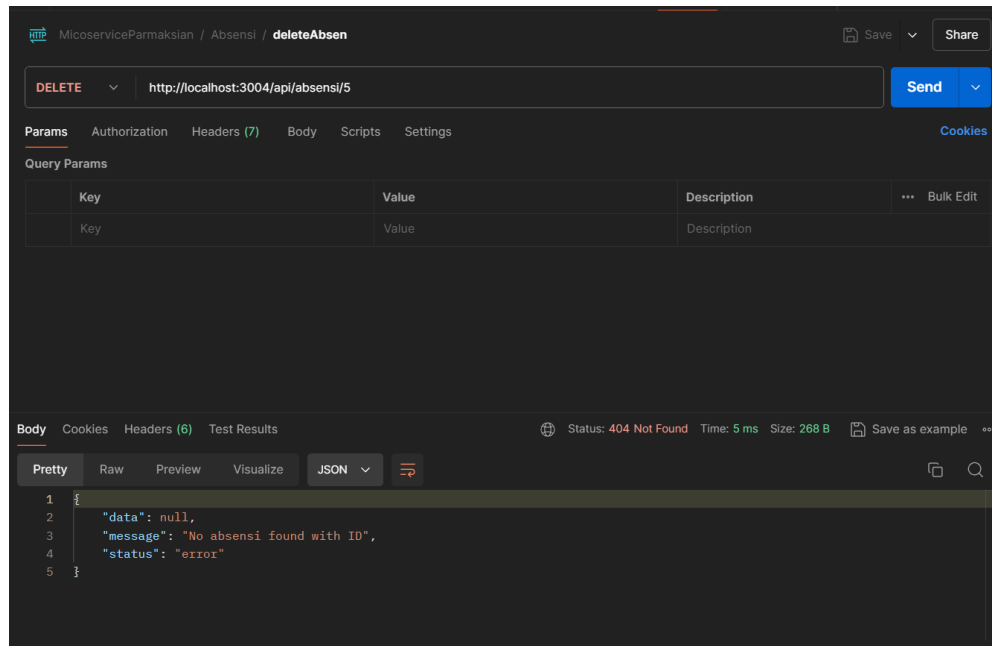
2.2.3 Absensi

a. POST: CreateAbsen



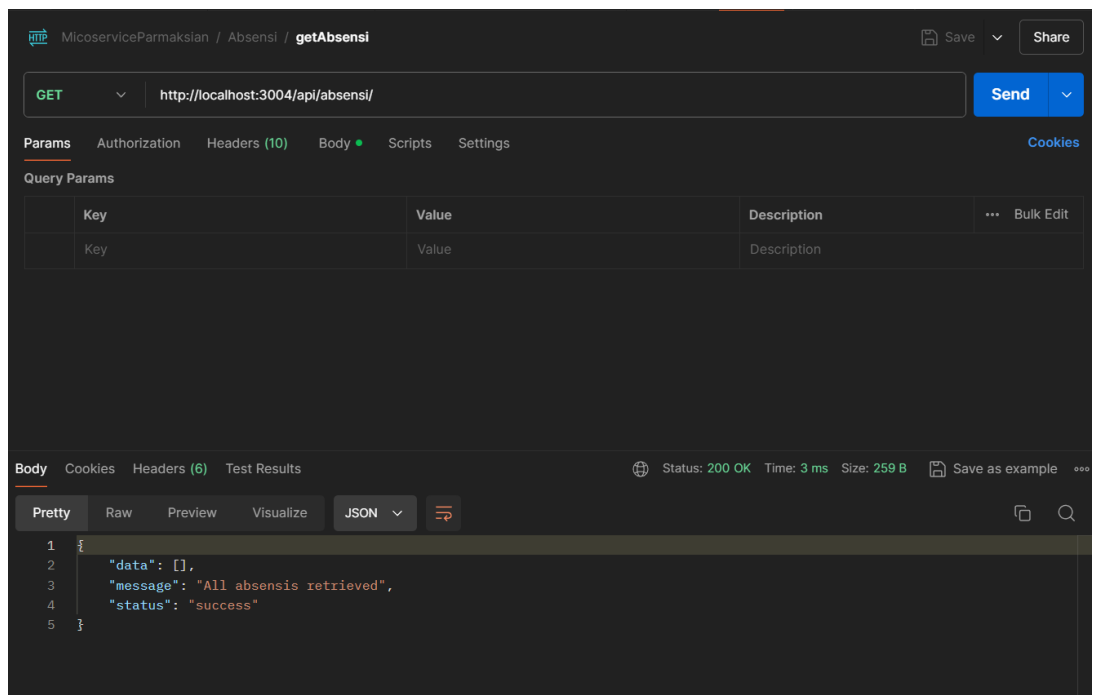
Gambar 19 POST: CreateAbsen

b. DELETE: deleteAbsen



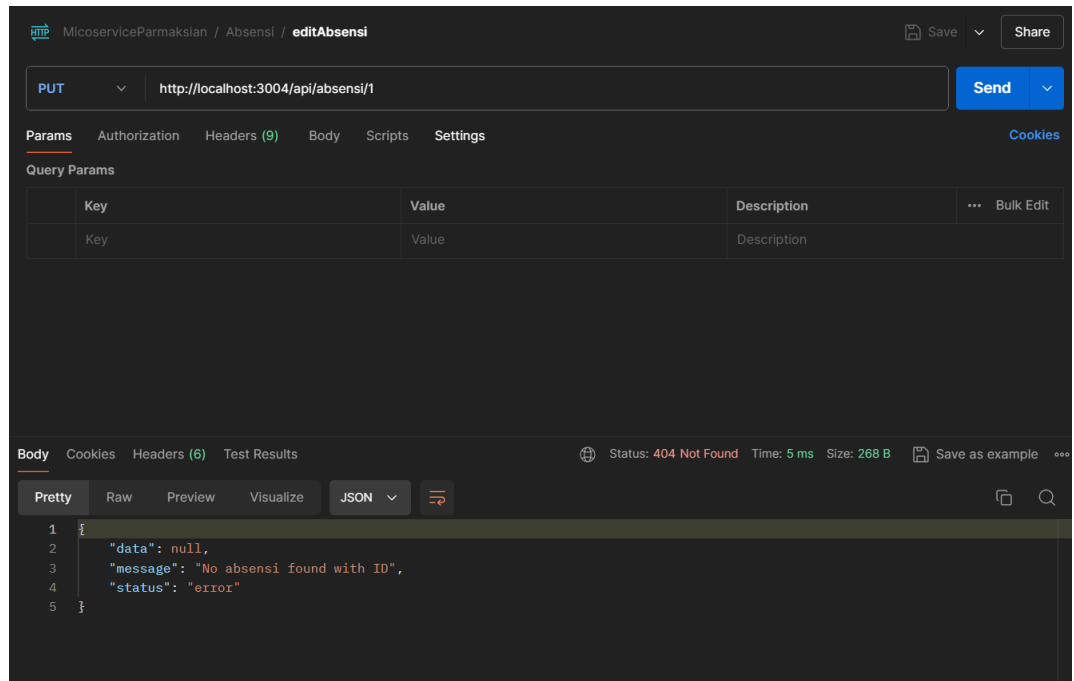
Gambar 20 DELETE: deleteAbsen

c. GET: getAbsensi



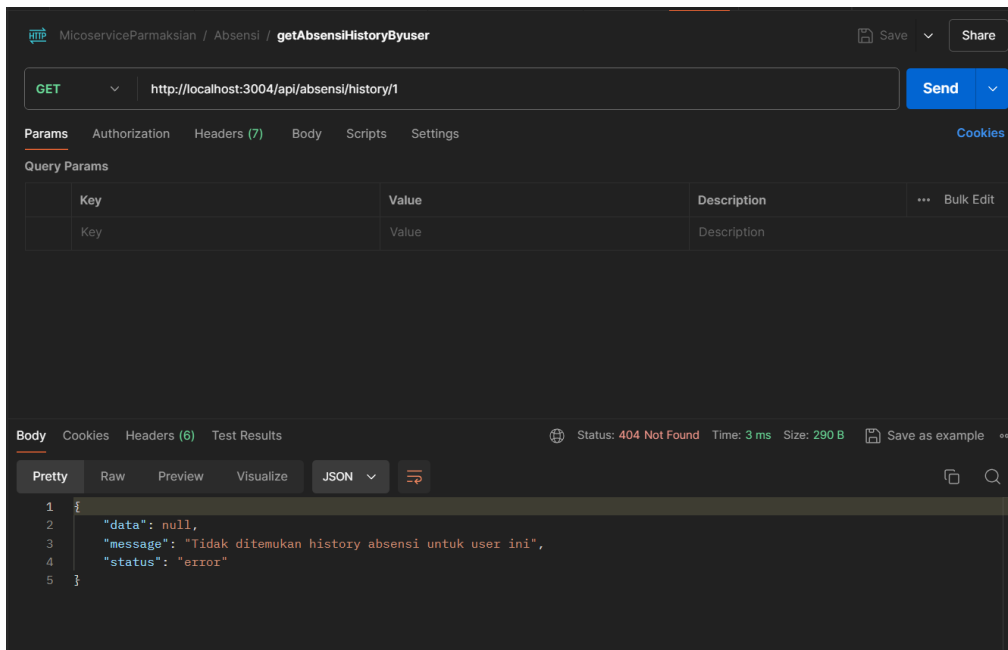
Gambar 21 GET: getAbsensi

d. PUT: editAbsensi



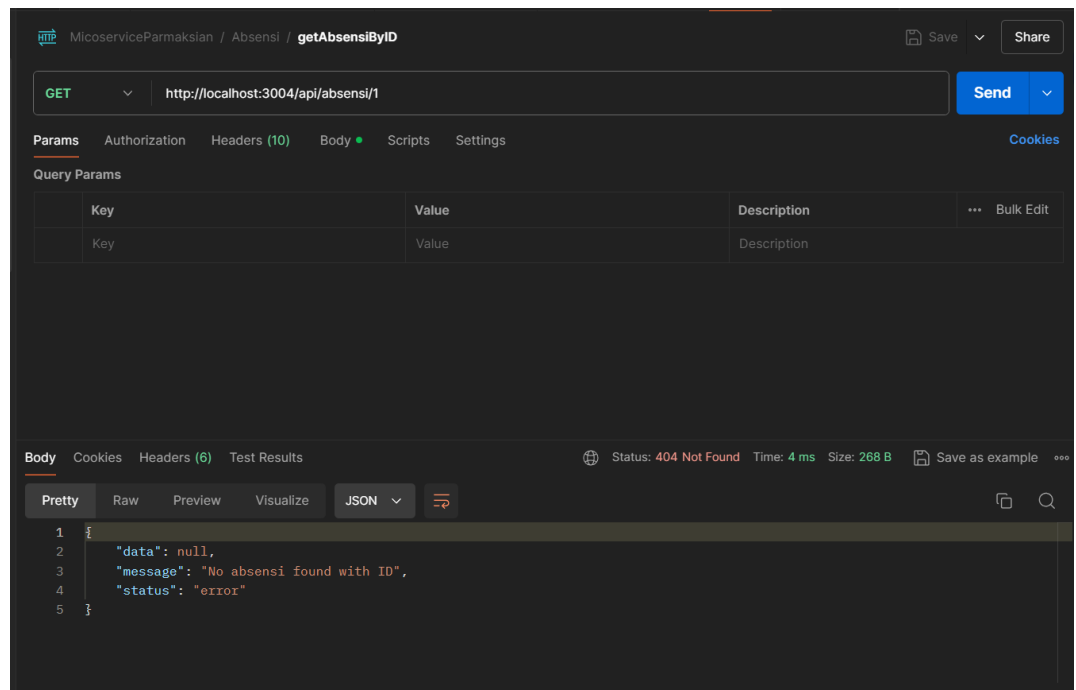
Gambar 22 PUT: editAbsensi

e. GET: getAbsensiHistoryByUser



Gambar 23 GET: getAbsensiHistoryByUser

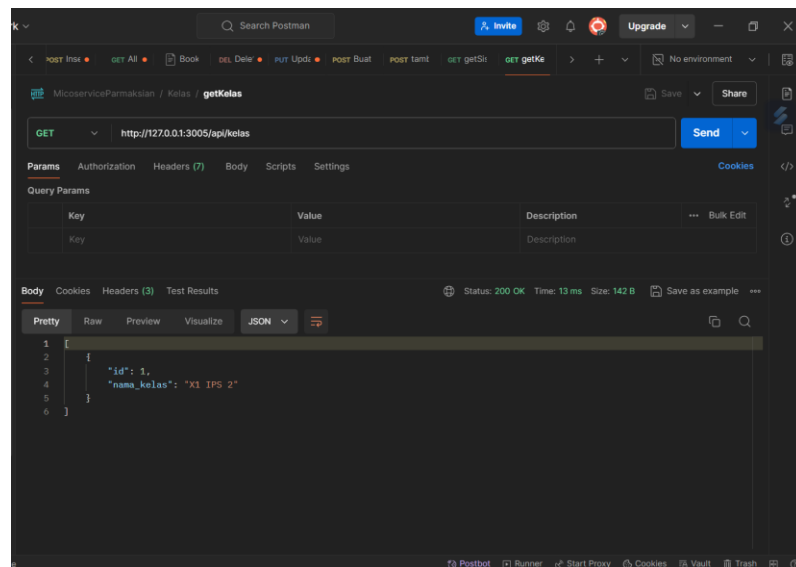
f. GET: getAbsensiByUser



Gambar 24 GET: getAbsensiByUser

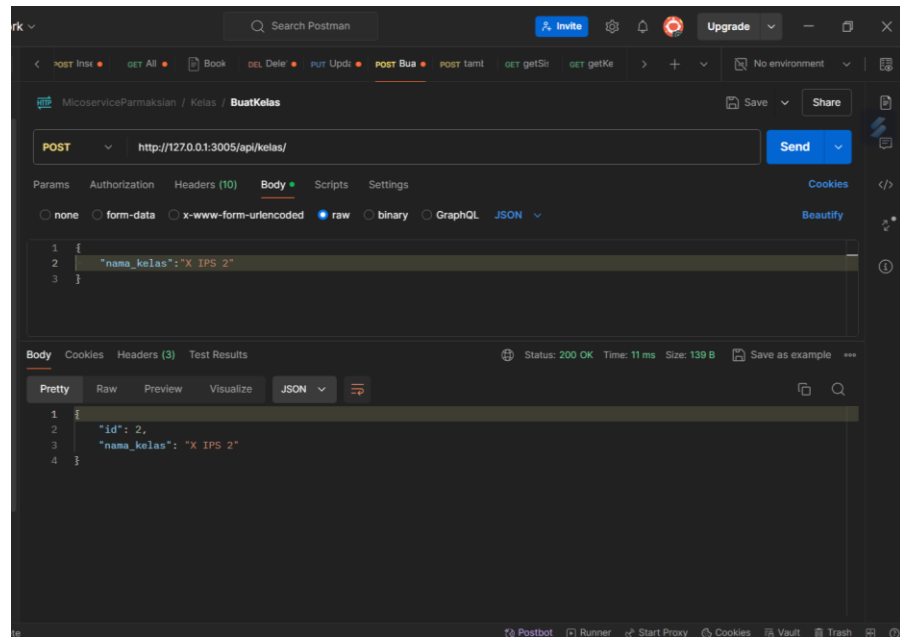
2.2.4 Kelas

a. POST: BuatKelas



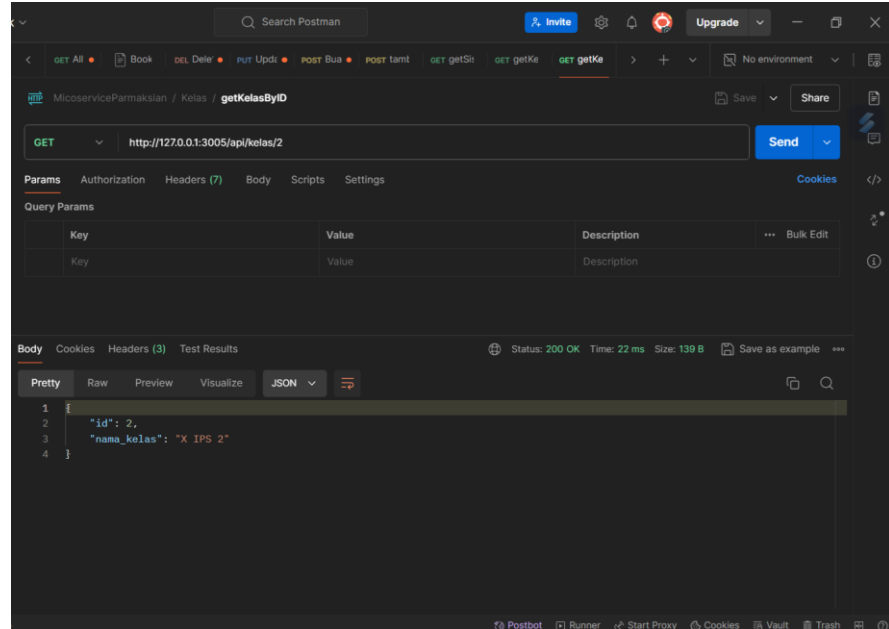
Gambar 25 POST: BuatKelas

b. GET: getKelas



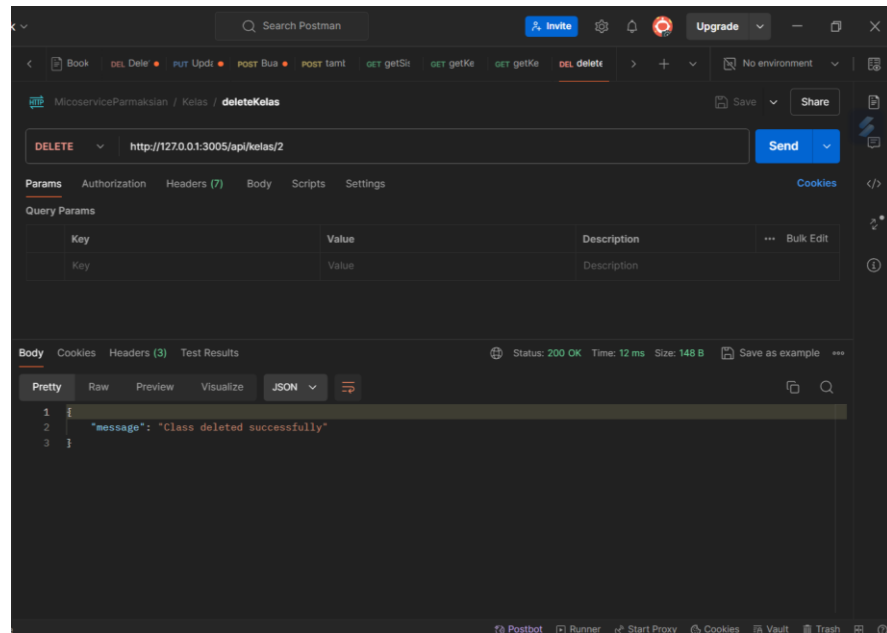
Gambar 26 GET: getKelas

c. GET: getKelasByID



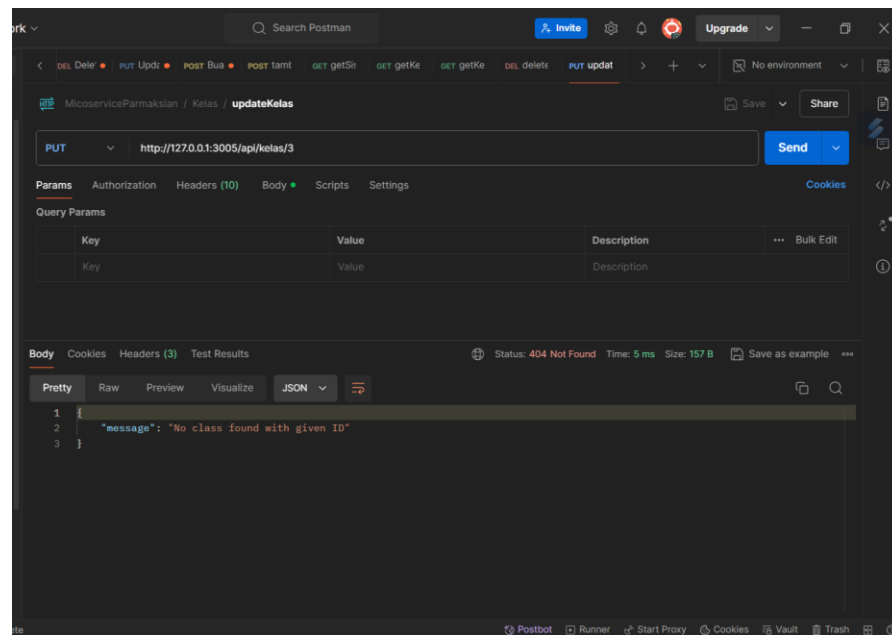
Gambar 27 GET: getKelasByID

d. DELETE: deleteKelas



Gambar 28 DELETE: deleteKelas

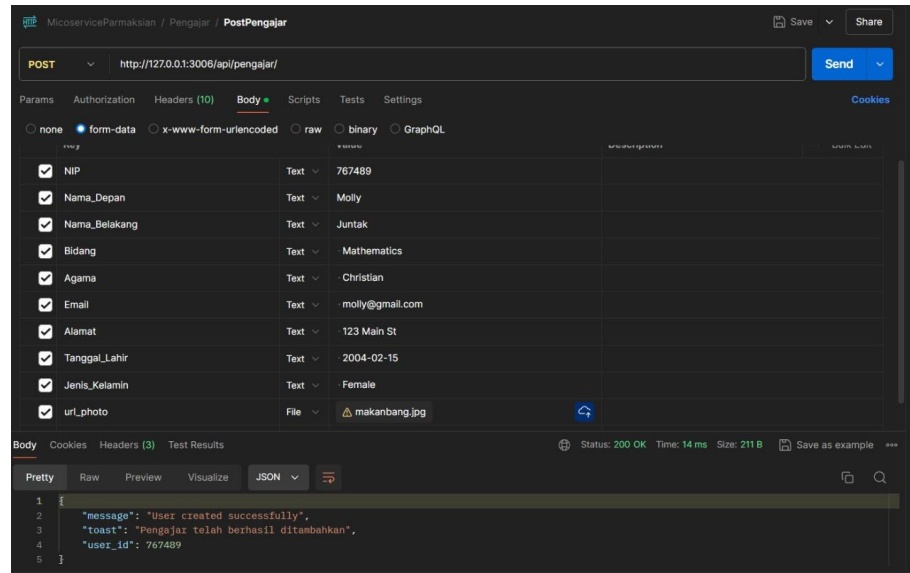
e. PUT: updateKelas



Gambar 29 PUT: updateKelas

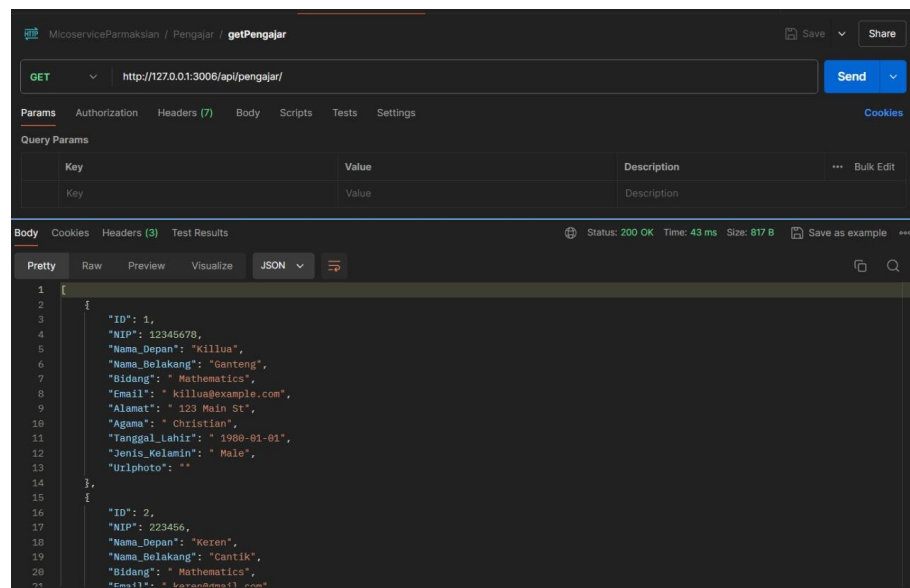
2.2.5 Pengajar

a. POST: PostPengajar



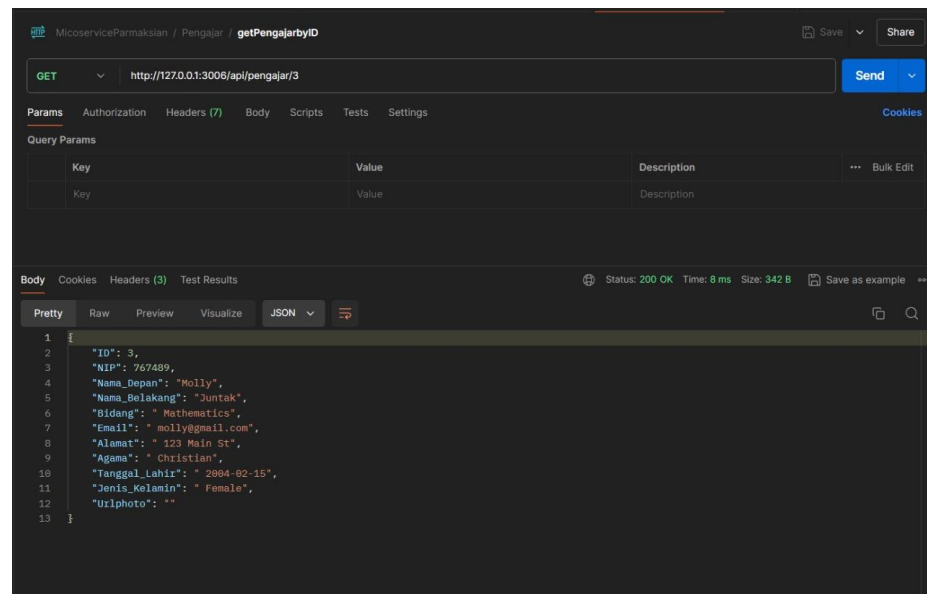
Gambar 30 POST: PostPengajar

b. GET: getPengajar



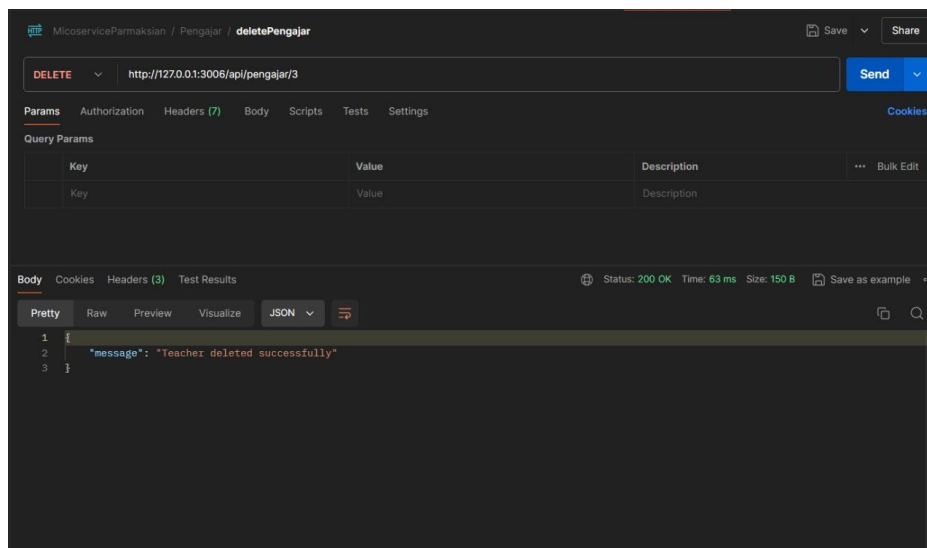
Gambar 31 GET: getPengajar

c. GET: getPengajarByID



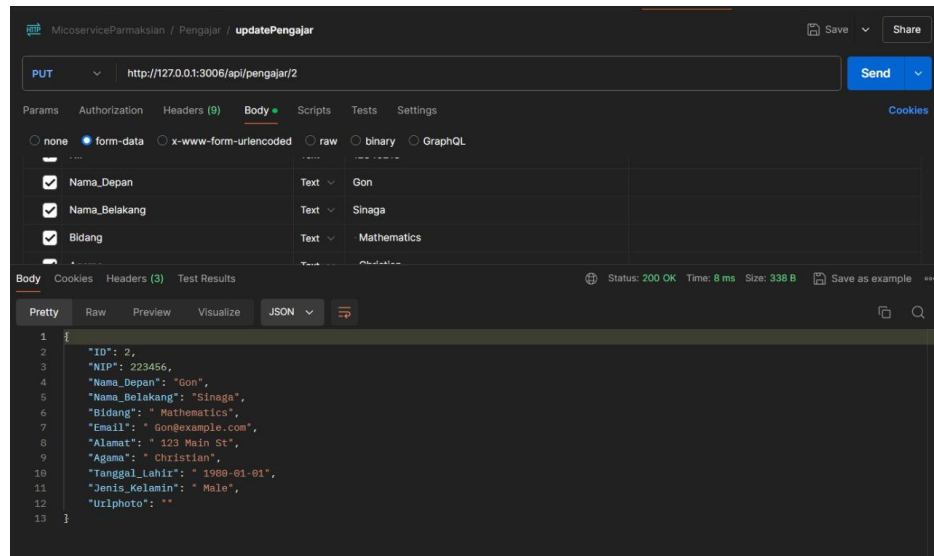
Gambar 32 GET: getPengajarByID

d. DELETE: deletePengajar



Gambar 33 DELETE: deletePengajar

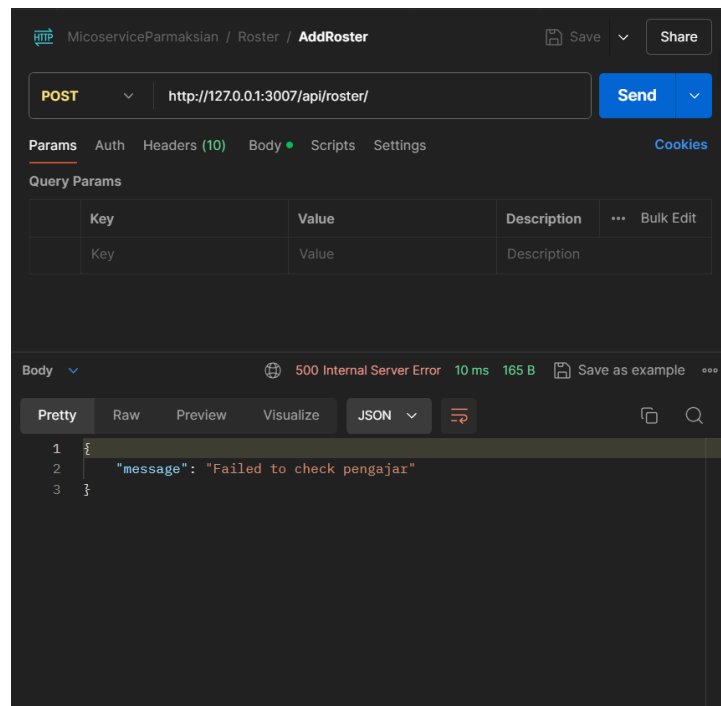
e. PUT: updatePengajar



Gambar 34 PUT: updatePengajar

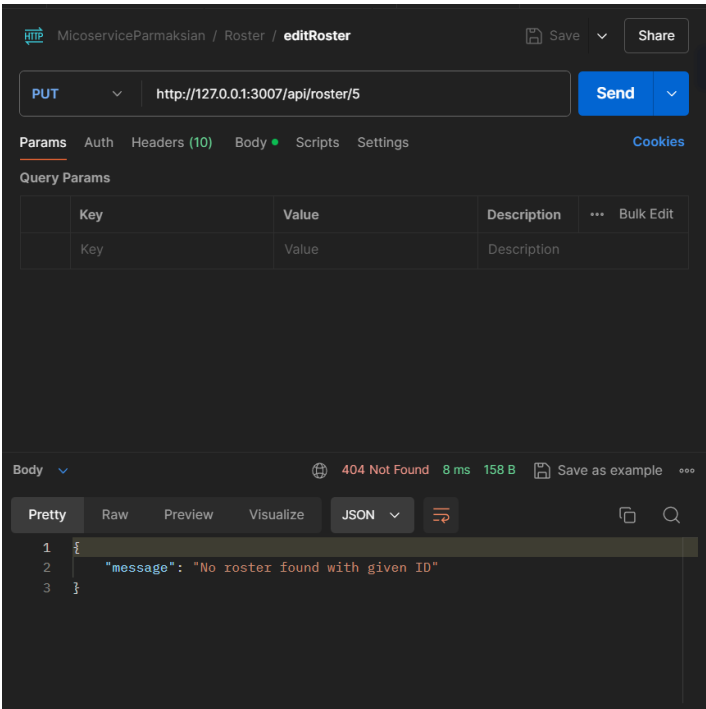
2.2.6 Roster

a. POST: AddRoster



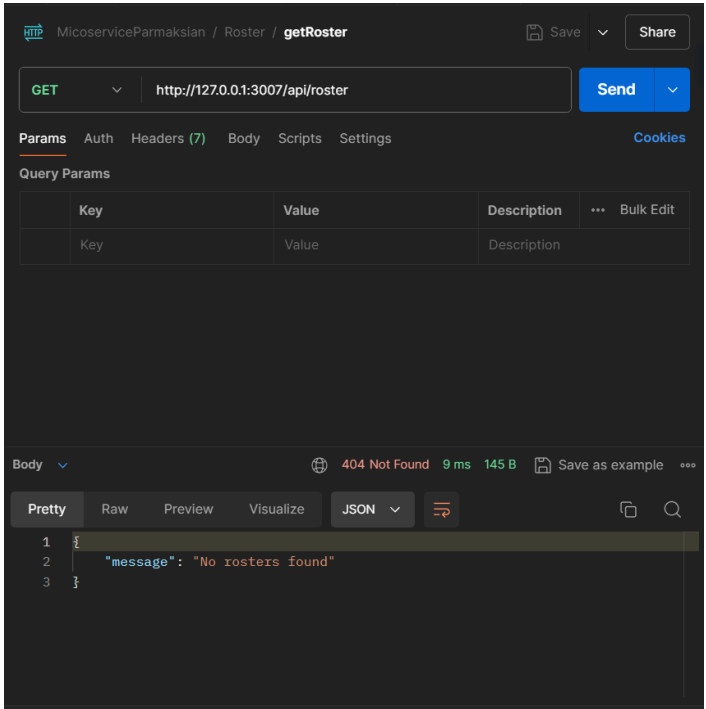
Gambar 35 POST: AddRoster

b. PUT: editRoster



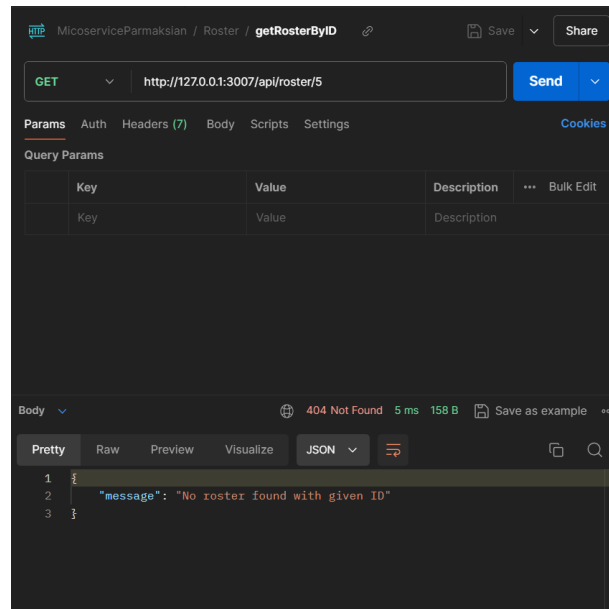
Gambar 36 PUT: editRoster

c. GET: getRoster



Gambar 37 GET: getRoster

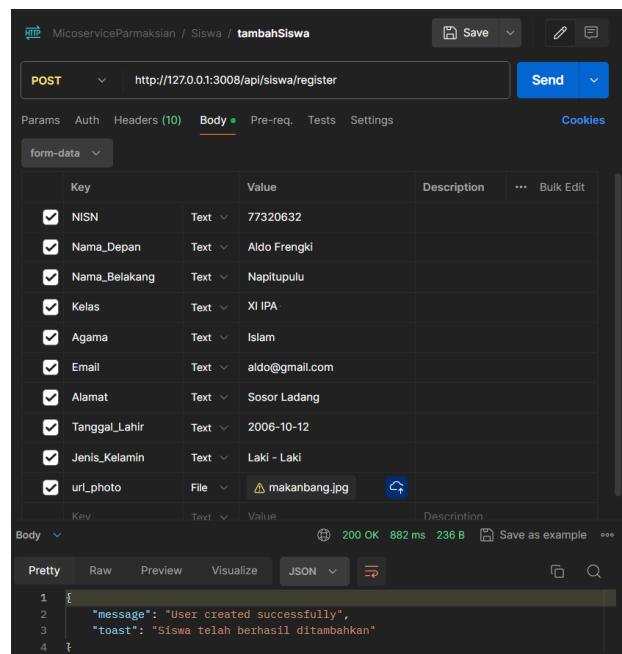
d. GET: getRosterByID



Gambar 38 GET: getRosterByID

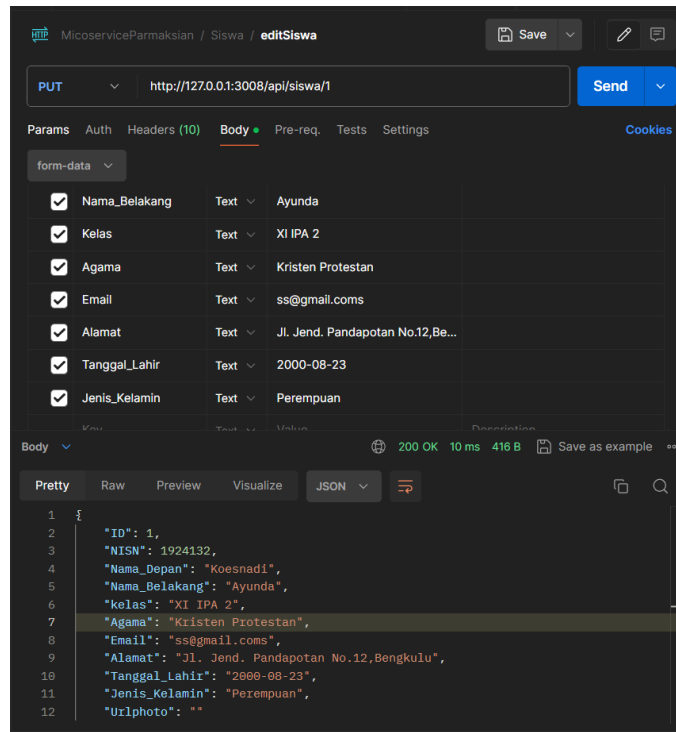
2.2.7 Siswa

a. POST: tambahSiswa



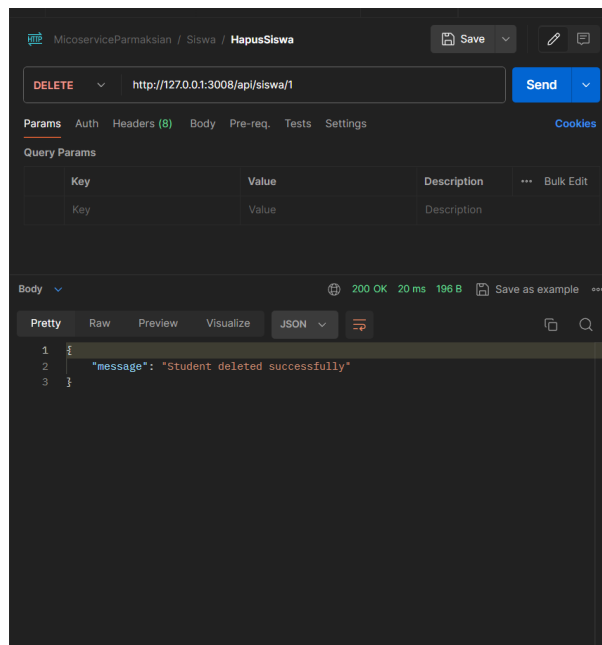
Gambar 39 POST: tambahSiswa

b. PUT: editSiswa



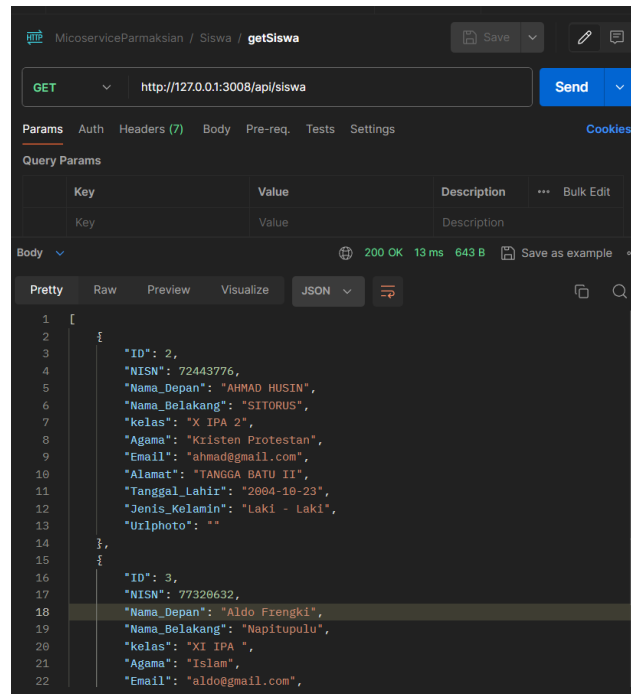
Gambar 40 PUT: editSiswa

c. DELETE: HapusSiswa



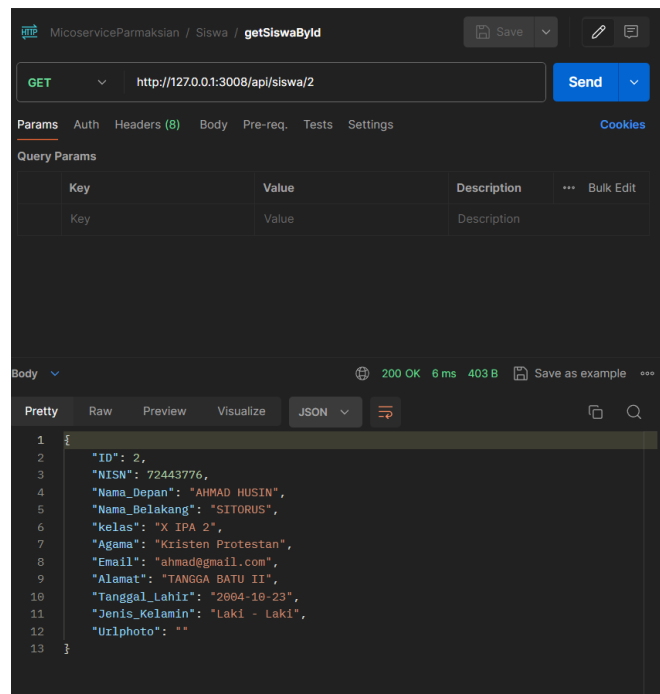
Gambar 41 DELETE: HapusSiswa

d. GET: getSiswa



Gambar 42 GET: getSiswa

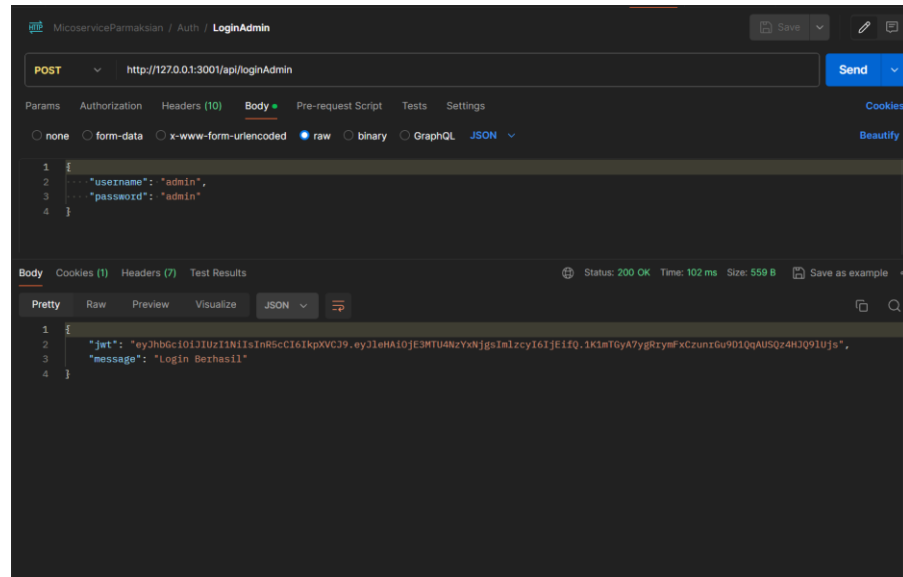
e. GET: getSiswaById



Gambar 43 GET: getSiswaById

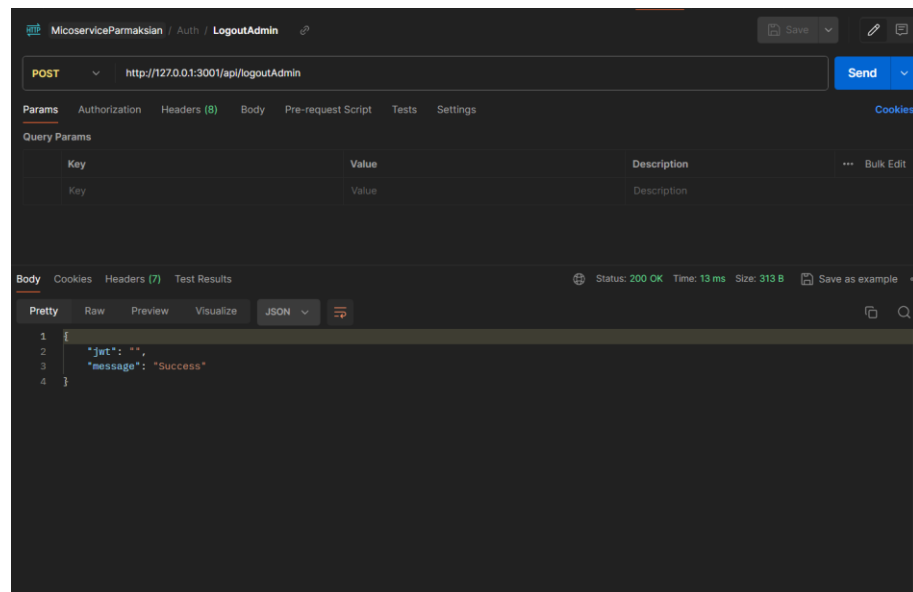
2.2.8 Authentication

a. POST: LoginAdmin



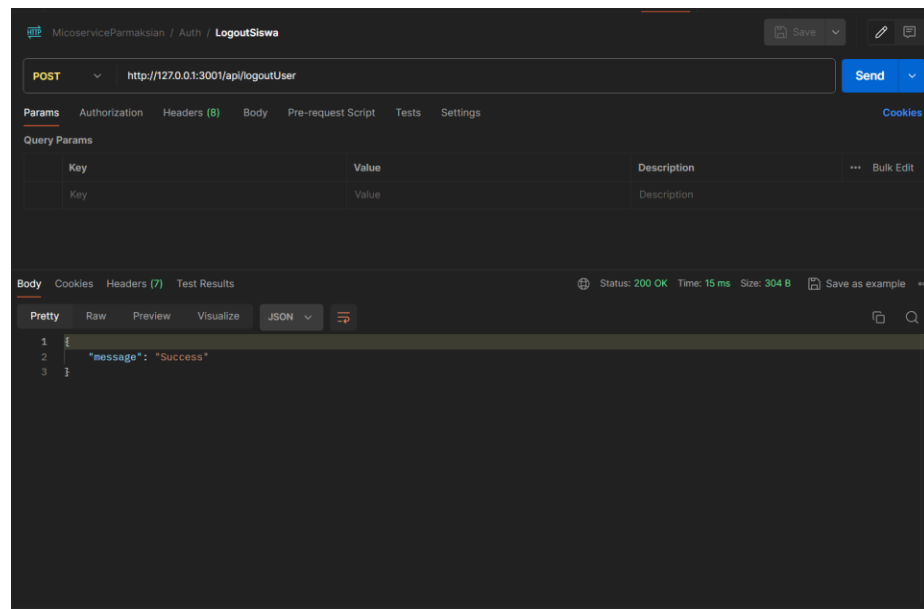
Gambar 44 POST: LoginAdmin

b. POST: LogoutAdmin



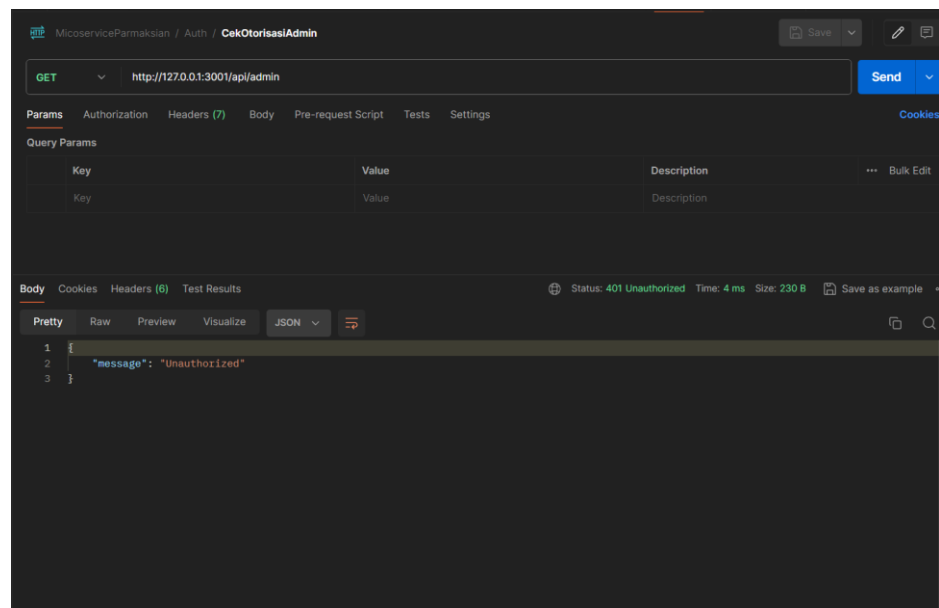
Gambar 45 POST: LogoutAdmin

c. POST: LogoutSiswa



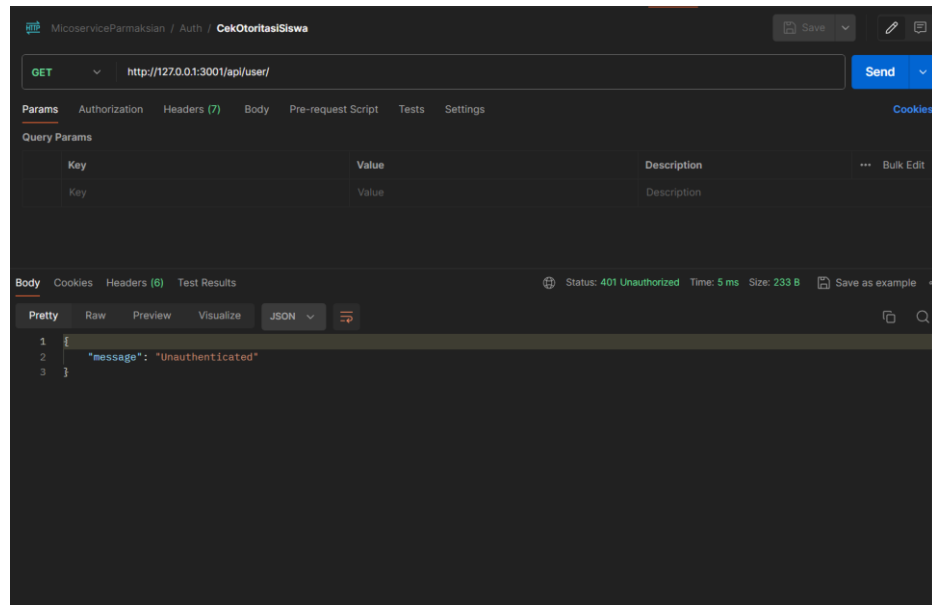
Gambar 46 POST: LogoutSiswa

d. GET: CekOtorisasiAdmin



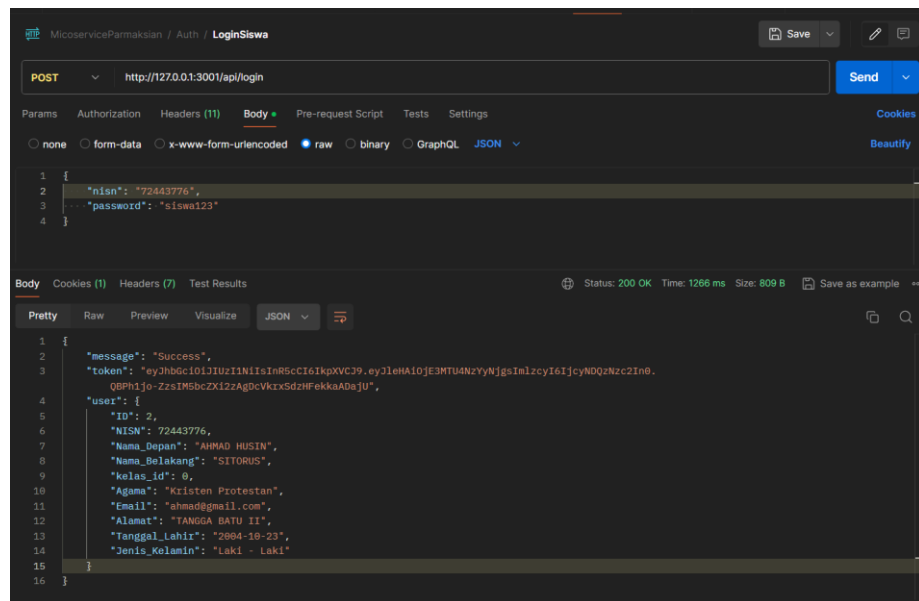
Gambar 47 GET: CekOtorisasiAdmin

e. GET: CekOtorisasiSiswa



Gambar 48 GET: CekOtorisasiSiswa

f. POST: LoginSiswa



Gambar 49 POST: Login Siswa