

LAPORAN
MONOLITH API DEVELOPMENT
PEMROGRAMAN BERBASIS FRAMEWORK



**SISTEM INFORMASI PENGELOLAAN LAPANGAN SPORT CENTER BERBASIS
WEB**

4K - Kelompok 3

Bayu Gilang Pamungkas	(244107027020)
Frankie Steinlie	(244107027008)
M. Agung Wibowo	(244107027001)
Prisca Dewi Anggita	(244107027002)

D-IV TEKNIK INFORMATIKA
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2025

BAB I

PENDAHULUAN

1.1.Deskripsi Project

Sistem Informasi Pengelolaan Lapangan Sport Center Berbasis Web merupakan sebuah platform digital yang dirancang untuk memudahkan pengelolaan, pemesanan, dan pemantauan penggunaan lapangan olahraga secara efisien. Sistem ini memungkinkan pengguna, baik pelanggan maupun pengelola, untuk mengakses informasi terkait ketersediaan lapangan, melakukan reservasi, dan pengelolaan jadwal lapangan penggunaan secara real-time melalui perangkat yang terhubung ke internet.

Dalam Sistem Informasi Pengelolaan Lapangan Sport Center Berbasis Web terdapat beberapa fitur utama yaitu :

- a. Kelola user
- b. Kelola kategori lapangan
- c. Kelola fasilitas lapangan
- d. Kelola lapangan
- e. Kelola jadwal lapangan
- f. Kelola Status lapangan
- g. Kelola Pemesanan
- h. Kelola Pembayaran

1.2.Maksud dan Tujuan Aplikasi

Adapun maksud dikembangkannya Sistem Informasi Pengelolaan Lapangan Sport Center Berbasis Web untuk menyediakan solusi digital dalam mengelola jadwal, reservasi, serta penggunaan lapangan secara lebih efektif dan efisien. Dengan adanya sistem ini, proses administrasi yang sebelumnya dilakukan secara manual dapat diotomatisasi, sehingga mengurangi potensi kesalahan, meningkatkan transparansi, dan memberikan kemudahan bagi pengelola serta pengguna dalam mengakses informasi terkait ketersediaan dan pemesanan lapangan.

Yang bertujuan untuk :

- a. Mempermudah manajemen jadwal
Mengatur dan mengelola jadwal penggunaan lapangan dengan sistem yang terstruktur, sehingga dapat menghindari tumpang tindih pemesanan.
- b. Meningkatkan efisiensi pemesanan lapangan

Memungkinkan pelanggan untuk melakukan reservasi secara mandiri melalui web tanpa harus datang langsung ke lokasi.

c. Menyediakan informasi secara real-time

Memberikan akses informasi ketersediaan lapangan secara langsung kepada pengguna dan pengelola.

d. Meningkatkan kepuasan pengguna

Memberikan pengalaman yang lebih mudah dan cepat dalam melakukan pemesanan, serta menyediakan fitur notifikasi pengingat jadwal.

e. Mempermudah pengelolaan keuangan dan laporan

Membantu pengelola dalam memantau transaksi pemesanan serta menyajikan laporan keuangan dan statistik penggunaan lapangan.

f. Mendukung transformasi digital dalam pengelolaan sport center

Meningkatkan efisiensi operasional dengan mengurangi penggunaan pencatatan manual dan beralih ke sistem digital yang lebih akurat dan mudah diakses.

Dengan adanya sistem ini, diharapkan pengelolaan sport center dapat berjalan lebih optimal, meningkatkan kualitas layanan kepada pengguna, serta mempercepat proses administrasi yang berkaitan dengan reservasi dan manajemen fasilitas olahraga.

1.3.Kemampuan Aplikasi

Sistem Informasi Pengelolaan Lapangan Sport Center Berbasis Web memiliki berbagai kemampuan yang dirancang untuk meningkatkan efisiensi dalam pengelolaan fasilitas olahraga, memudahkan reservasi lapangan, serta memberikan kemudahan bagi pengguna dan pengelola. Berikut adalah kemampuan utama aplikasi ini :

a. Kelola user

- a) Sistem registrasi dan login bagi pengguna dengan autentikasi aman
- b) Manajemen hak akses pengguna berdasarkan peran (admin, dan pelanggan)
- c) Pengelolaan data pengguna, termasuk informasi profil, riwayat pemesanan, dan transaksi.
- d) Fitur pencarian dan pengelompokan pengguna berdasarkan status keanggotaan.

b. Kelola Kategori Lapangan

- a) Mengatur berbagai jenis lapangan berdasarkan kategori seperti futsal, basket, dan badminton.
- b) Memudahkan pencarian lapangan berdasarkan kategori tertentu
- c) Menampilkan informasi kategori lapangan yang tersedia untuk pengguna.

- c. Kelola Fasilitas Lapangan
 - a) Pengelolaan fasilitas tambahan disetiap lapangan, seperti gazebo, kamar mandi, atau fitting room.
 - b) Informasi fasilitas ditampilkan dalam deskripsi lapangan saat pemesanan.
 - c) Memungkinkan pengelola untuk menambah atau memperbaiki fasilitas yang tersedia.
- d. Kelola Lapangan
 - a) Menampilkan daftar lapangan yang tersedia dengan detail informasi seperti ukuran, lokasi, harga sewa, dan fasilitas.
 - b) Pengelolaan lapangan oleh admin, termasuk penambahan lapangan baru dan pembaruan informasi.
 - c) Sistem pencarian dan filter untuk mempermudah pengguna dalam menemukan lapangan yang sesuai.
- e. Kelola Jadwal Lapangan
 - a) Menyediakan jadwal penggunaan lapangan berdasarkan hari dan sesi waktu tertentu.
 - b) Memungkinkan pengelola untuk mengatur dan memperbaiki jadwal secara dinamis.
 - c) Sistem real-time yang mencegah pemesanan ganda (double booking).
 - d) Menampilkan jadwal yang telah dipesan dan yang masih tersedia kepada pengguna.
- f. Kelola Status Lapangan
 - a) Menampilkan status lapangan (tersedia, disewa, dalam perbaikan).
 - b) Memungkinkan admin untuk memperbaiki status lapangan secara otomatis atau manual.
 - c) Memberikan informasi real-time kepada pengguna mengenai ketersediaan lapangan.
- g. Kelola Pemesanan
 - a) Sistem pemesanan online yang memungkinkan pengguna memilih lapangan, jadwal, dan fasilitas tambahan.
 - b) Pengguna dapat melihat riwayat pemesanan dan status reservasi mereka.
 - c) Konfirmasi pemesanan otomatis setelah pembayaran berhasil dilakukan.
 - d) Fitur pembatalan atau perubahan jadwal pemesanan dengan ketentuan tertentu.
- h. Kelola Pembayaran

- a) Mendukung berbagai metode pembayaran seperti transfer dan midtrans.
- b) Sistem verifikasi pembayaran otomatis dan manual oleh admin.
- c) Terdapat bukti transfer menggunakan midtrans atau transfer manual.
- d) Tertera status (menunggu verifikasi, belum dibayar, ditolak, diverifikasi).

1.4.Deskripsi User

Dalam aplikasi Sistem Informasi Pengelolaan Lapangan Sport Center Berbasis Web, terdapat dua jenis pengguna utama, yaitu Admin dan Pelanggan. Masing-masing memiliki peran dan hak akses yang berbeda untuk memastikan sistem berjalan dengan optimal.

a) Admin

Peran :

Admin merupakan pengguna dengan hak akses penuh yang bertanggung jawab atas pengelolaan sistem, mulai dari mengatur jadwal hingga memproses pemesanan dan pembayaran.

Hak Akses dan Kemampuan:

- a) Mengelola data pengguna (menambah, mengedit, atau menghapus akun pelanggan).
- b) Mengatur kategori lapangan dan fasilitas yang tersedia.
- c) Menjadwalkan dan memperbarui status ketersediaan lapangan.
- d) Memproses dan mengelola pemesanan yang masuk.
- e) Memverifikasi dan memproses pembayaran pelanggan.
- f) Melihat dan mengelola laporan transaksi serta statistik pemakaian lapangan.
- g) Mengirim notifikasi atau pengumuman terkait perubahan jadwal atau promo.

b) Pelanggan

Peran :

Pelanggan merupakan pengguna yang menggunakan sistem untuk melihat informasi lapangan, melakukan reservasi, dan melakukan pembayaran.

Hak Akses dan Kemampuan:

- a) Mendaftar dan mengelola akun pribadi.
- b) Melihat daftar lapangan beserta kategori, fasilitas, dan jadwalnya.
- c) Melakukan pemesanan lapangan secara online.
- d) Melakukan pembayaran melalui berbagai metode yang tersedia.
- e) Melihat riwayat pemesanan dan status pembayaran.

- f) Menerima notifikasi konfirmasi dan pengingat jadwal pemesanan.
- g) Membatalkan atau mengubah pemesanan sesuai kebijakan yang berlaku.

BAB II

IDENTIFIKASI PROSES BISNIS/REQUIREMENT

1.1.Manajemen user

- a. API untuk menambahkan user
- b. API untuk mengedit user
- c. API untuk menampilkan semua user
- d. API untuk menampilkan user berdasarkan id
- e. API untuk menampilkan user berdasarkan nim
- f. API untuk menghapus user

1.2.Manajemen kategori

- a. API untuk menambahkan kategori
- b. API untuk mengedit kategori
- c. API untuk menampilkan semua kategori
- d. API untuk menampilkan kategori berdasarkan id
- e. API untuk menghapus kategori

1.3.Manajemen fasilitas

- a. API untuk menambahkan fasilitas
- b. API untuk mengedit fasilitas
- c. API untuk menampilkan semua fasilitas
- d. API untuk menampilkan fasilitas berdasarkan id
- e. API untuk menghapus fasilitas

1.4.Manajemen sesi

- a. API untuk menambahkan sesi
- b. API untuk mengedit sesi
- c. API untuk menampilkan semua sesi
- d. API untuk menampilkan sesi berdasarkan id
- e. API untuk menghapus sesi

1.5.Manajemen lapangan

- a. API untuk menambahkan lapangan
- b. API untuk mengedit lapangan
- c. API untuk menampilkan semua lapangan
- d. API untuk menampilkan lapangan berdasarkan id
- e. API untuk menghapus lapangan

1.6.Manajemen status lapangan

- a. API untuk menambahkan status lapangan
- b. API untuk mengedit status lapangan
- c. API untuk menampilkan semua status lapangan
- d. API untuk menampilkan status lapangan berdasarkan id
- e. API untuk menghapus status lapangan

1.7.Manajemen pemesanan

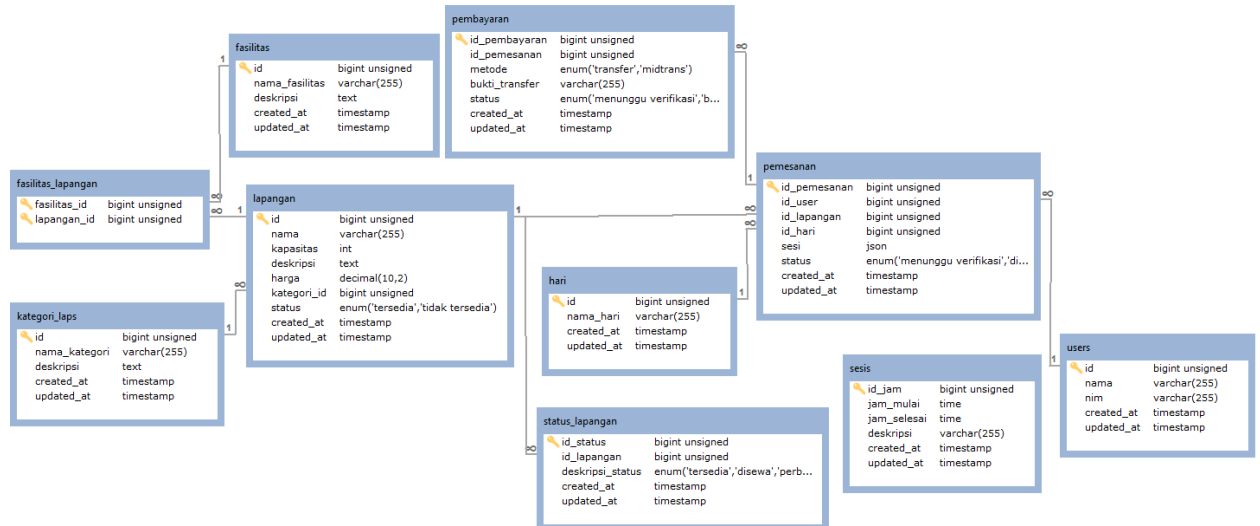
- a. API untuk menambahkan pemesanan
- b. API untuk mengedit pemesanan
- c. API untuk menampilkan semua pemesanan
- d. API untuk menampilkan pemesanan berdasarkan id
- e. API untuk menghapus pemesanan

1.8.Manajemen pembayaran

- a. API untuk menambahkan pembayaran
- b. API untuk mengedit pembayaran
- c. API untuk menampilkan semua pembayaran
- d. API untuk menampilkan pembayaran berdasarkan id
- e. API untuk menghapus pembayaran

BAB III

1.1.Database Design



1.2. Penjelasan Database

- a. Table users
 - a) Menyimpan informasi tentang pengguna yang melakukan pemesanan.
 - b) Kolom utama: id, nama, nim, created_at, update_at.
- b. Table fasilitas
 - a) Menyimpan fasilitas yang tersedia dilapangan.
 - b) Kolom utama: id, nama_fasilitas, deskripsi.
- c. Table fasilitas_lapangan
 - a) Menghubungkan fasilitas dengan lapangan.
 - b) Kolom utama: fasilitas_id, lapangan_id.
- d. Table sesi
 - a) Menyimpan informasi sesi waktu yang bisa dipilih saat melakukan pemesanan.
 - b) Kolom utama: id_jam, jam_mulai, jam_selesai, deskripsi.
- e. Table lapangan
 - a) Menyimpan informasi tentang lapangan yang tersedia untuk disewa.
 - b) Kolom utama: id, nama, kapasitas, deskripsi, harga, kategori_id, status.
- f. Table kategori_laps
 - a) Menyimpan kategori untuk lapangan.
 - b) Kolom utama: id, nama_kategori, deskripsi.

- g. Table hari
 - a) Menyimpan daftar hari yang bisa digunakan dalam pemesanan.
 - b) Kolom utama: id, nama_hari.
- h. Table status_lapangan
 - a) Menyimpan status dari lapangan, seperti tersedia, disewa, atau perbaikan.
 - b) Kolom utama: id_status, id_lapangan, deskripsi_status.
- i. Table pemesanan
 - a) Menyimpan data pemesanan lapangan oleh pengguna.
 - b) Kolom utama: id_pemesanan, id_user, id_lapangan, id_hari, sesi, status.
- j. Table pembayaran
 - a) Menyimpan informasi pembayaran dari pemesanan yang dilakukan.
 - b) Kolom utama: id_pembayaran, id_pemesanan, metode, bukti_transfer, status.

Adapun penjelasan relasi antar tabelnya sebagai berikut:

- 1) **Users** terkait dengan pemesanan melalui **id_user**.
- 2) **Lapangan** memiliki hubungan dengan **kategori_laps**, **fasilitas_lapangan**, **status_lapangan**, dan **pemesanan**.
- 3) **Fasilitas** dihubungkan dengan **lapangan** melalui **fasilitas_lapangan**.
- 4) **Pemesanan** terhubung ke **hari** dan **sesi**.
- 5) **Pemesanan** memiliki relasi dengan **pembayaran**.

BAB IV

END POINT API

1.1.End Point API Users

POST	/api/users	Menambahkan user baru
PUT	/api/users/{id}	Mengupdate user berdasarkan id
GET	/api/users	Menampilkan semua user
GET	/api/users/{id}	Menampilkan user berdasarkan id
GET	/api/users/nim/{nim}	Menampilkan user berdasarkan nim
DELETE	/api/users/{id}	Menghapus user berdasarkan id

1.2.End Point API Kategori

POST	/api/kategori-laps	Menambahkan kategori lapangan baru
PUT	/api/kategori-laps/{id}	Mengupdate kategori lapangan berdasarkan id
GET	/api/kategori-laps	Menampilkan semua kategori lapangan
GET	/api/kategori-laps/{id}	Menampilkan kategori lapangan berdasarkan id
DELETE	/api/kategori-laps/{id}	Menghapus kategori lapangan berdasarkan id

1.3.End Point API Fasilitas

POST	/api/fasilitas-lapangan	Menambahkan relasi fasilitas-lapangan baru
PUT	/api/fasilitas-lapangan/{id}	Mengupdate relasi fasilitas-lapangan berdasarkan id
GET	/api/fasilitas-lapangan	Menampilkan semua relasi fasilitas-lapangan
GET	/api/fasilitas-lapangan/fasilitas/{fasilitas_id}	Menampilkan relasi berdasarkan fasilitas id
GET	/api/fasilitas-lapangan/lapangan/{lapangan_id}	Menampilkan relasi berdasarkan lapangan id
DELETE	/api/fasilitas-lapangan/{id}	Menghapus relasi fasilitas-lapangan berdasarkan id

1.4.End Point API Sesi

POST	/api/sesi	Menambahkan sesi baru
PUT	/api/sesi/{id}	Mengupdate sesi berdasarkan id
GET	/api/sesi	Menampilkan semua sesi
GET	/api/sesi/{id}	Menampilkan sesi berdasarkan id
GET	/api/sesi/jam/{jam_id}	Menampilkan sesi berdasarkan jam id
DELETE	/api/sesi/{id}	Menghapus sesi berdasarkan id

1.5.End Point API Lapangan

POST	/api/lapangan	Menambahkan lapangan baru
PUT	/api/lapangan/{id}	Mengupdate lapangan berdasarkan id
GET	/api/lapangan	Menampilkan semua lapangan
GET	/api/lapangan/{id}	Menampilkan lapangan berdasarkan id
GET	/api/lapangan/status/{status}	Menampilkan lapangan berdasarkan status
DELETE	/api/lapangan/{id}	Menghapus lapangan berdasarkan id

1.6.End Point API Status Lapangan

POST	/api/status-lapangan	Menambahkan status lapangan baru
PUT	/api/status-lapangan/{id}	Mengupdate status lapangan berdasarkan id
GET	/api/status-lapangan	Menampilkan semua status lapangan
GET	/api/status-lapangan/{id}	Menampilkan status lapangan berdasarkan id
GET	/api/status-lapangan/lapangan/{lapangan_id}	Menampilkan status berdasarkan lapangan id
DELETE	/api/status-lapangan/{id}	Menghapus status lapangan berdasarkan id

1.7.End Point API Pemesanan

POST	/api/pemesanan	Menambahkan pemesanan baru
PUT	/api/pemesanan/{id}	Mengupdate pemesanan berdasarkan id
GET	/api/pemesanan	Menampilkan semua pemesanan

GET	/api/pemesanan/{id}	Menampilkan pemesanan berdasarkan id
GET	/api/pemesanan/user/{user_id}	Menampilkan pemesanan berdasarkan user id
GET	/api/pemesanan/lapangan/{lapangan_id}	Menampilkan pemesanan berdasarkan lapangan id
GET	/api/pemesanan/status/{status}	Menampilkan pemesanan berdasarkan status
DELETE	/api/pemesanan/{id}	Menghapus pemesanan berdasarkan id

1.8.End Point API Pembayaran

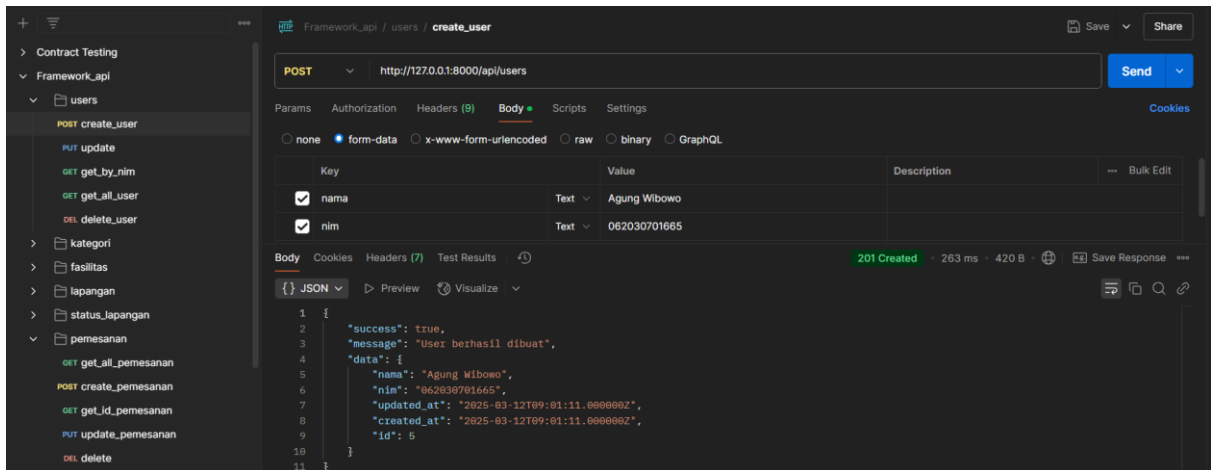
POST	/api/pembayaran	Menambahkan pembayaran baru
PUT	/api/pembayaran/{id}	Mengupdate pembayaran berdasarkan id
GET	/api/pembayaran	Menampilkan semua pembayaran
GET	/api/pembayaran/{id}	Menampilkan pembayaran berdasarkan id
GET	/api/pembayaran/pemesanan/{pemesanan_id}	Menampilkan pembayaran berdasarkan pemesanan id
GET	/api/pembayaran/metode/{metode}	Menampilkan pembayaran berdasarkan metode
GET	/api/pembayaran/status/{status}	Menampilkan pembayaran berdasarkan status
DELETE	/api/pembayaran/{id}	Menghapus pembayaran berdasarkan id

BAB V

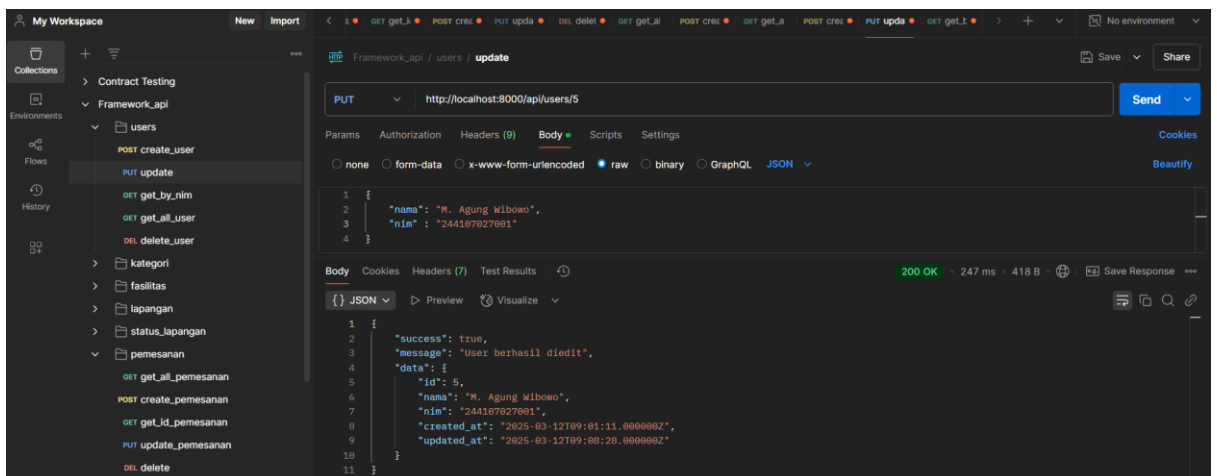
DOKUMENTASI API DAN CONTOH REQUEST

1.1. Dokumentasi API Users

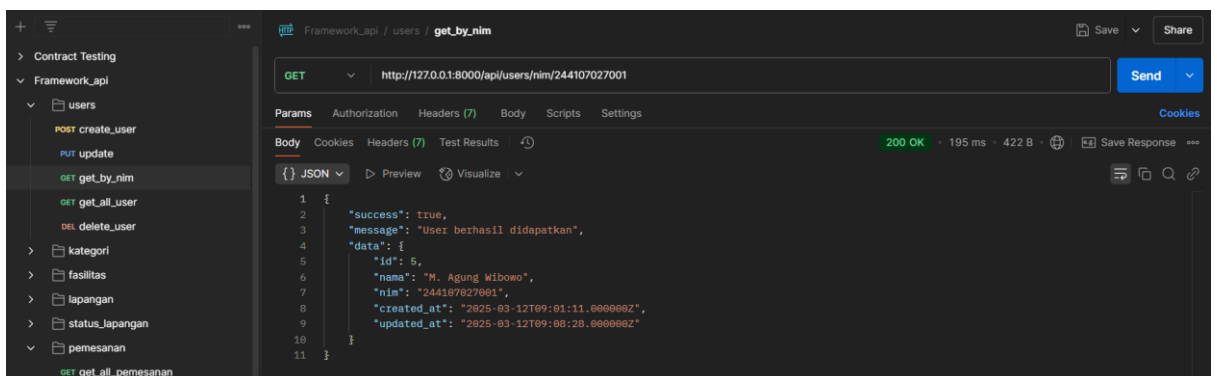
1. Membuat data User (POST/api/users)



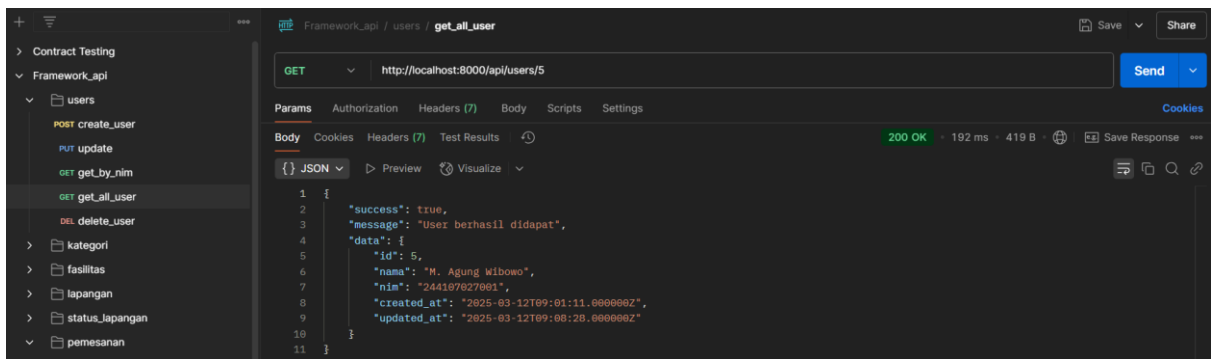
2. Update Users (PUT/api/users)



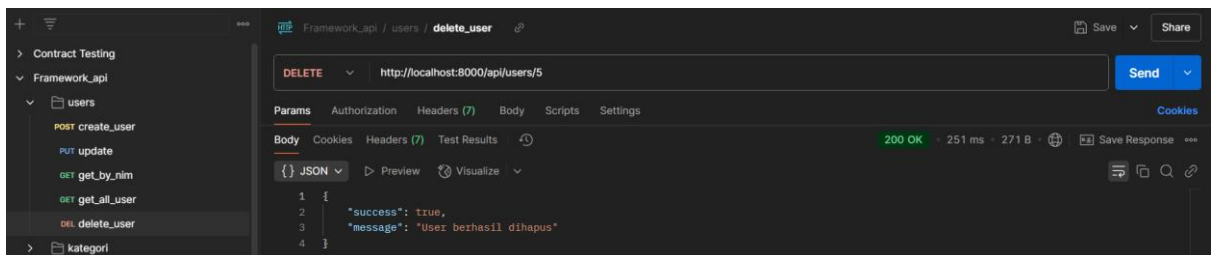
3. Mengambil data berdasarkan NIM (GET/api/users/{nim})



4. Mengambil data berdasarkan id users (GET/api/users/{id})

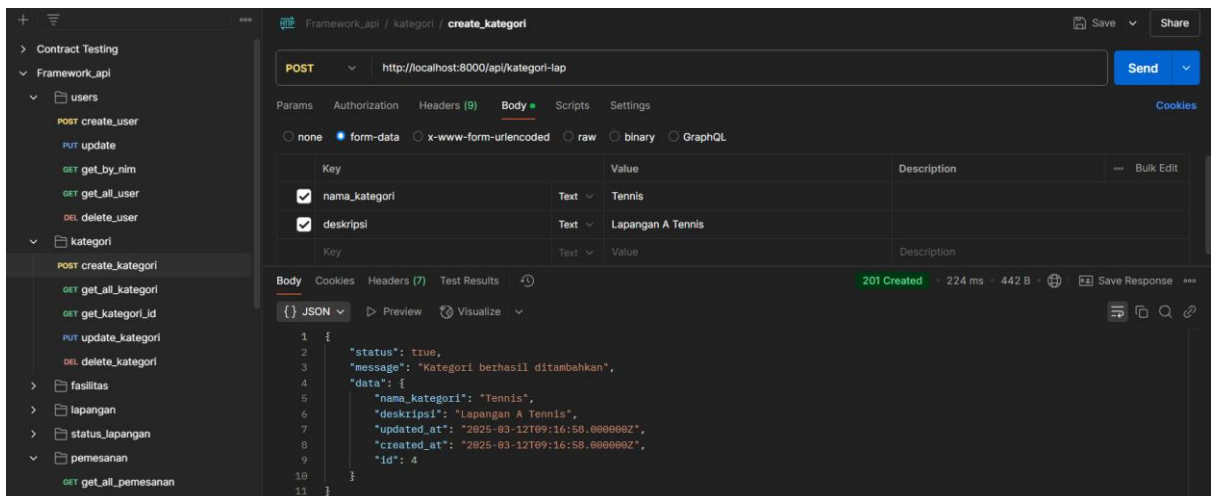


5. Menghapus data berdasarkan id users (DELETE/api/users)

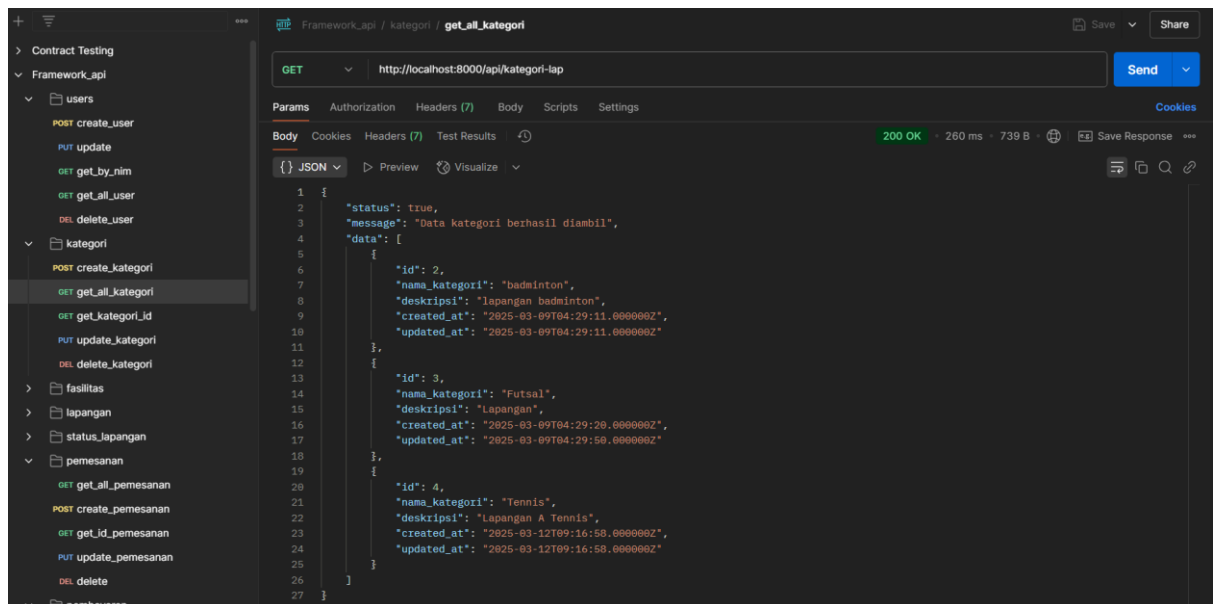


1.2. Dokumentasi API Kategori Lapangan

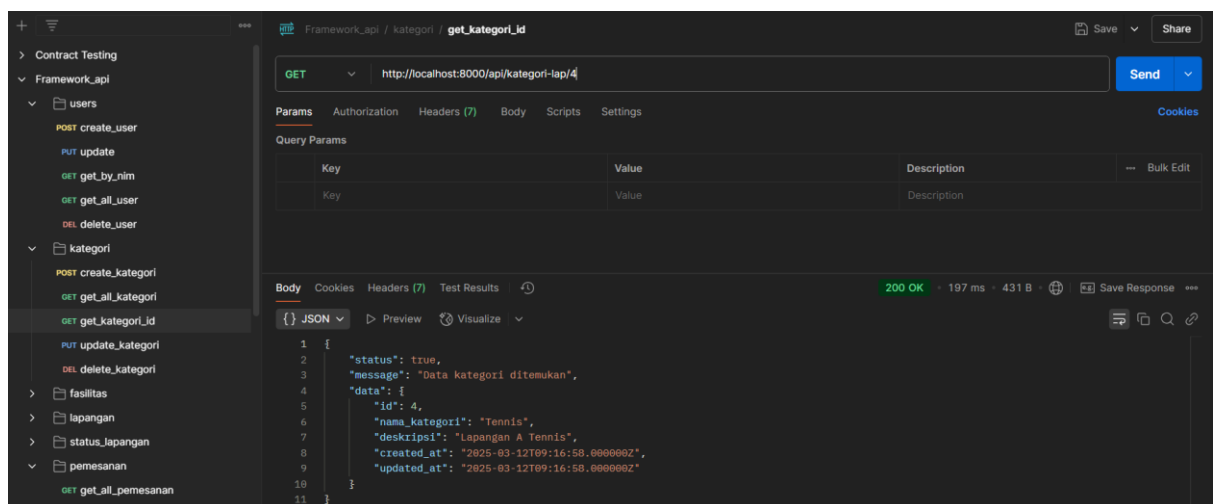
1. Membuat data Kategori (POST/api/kategori-lap)



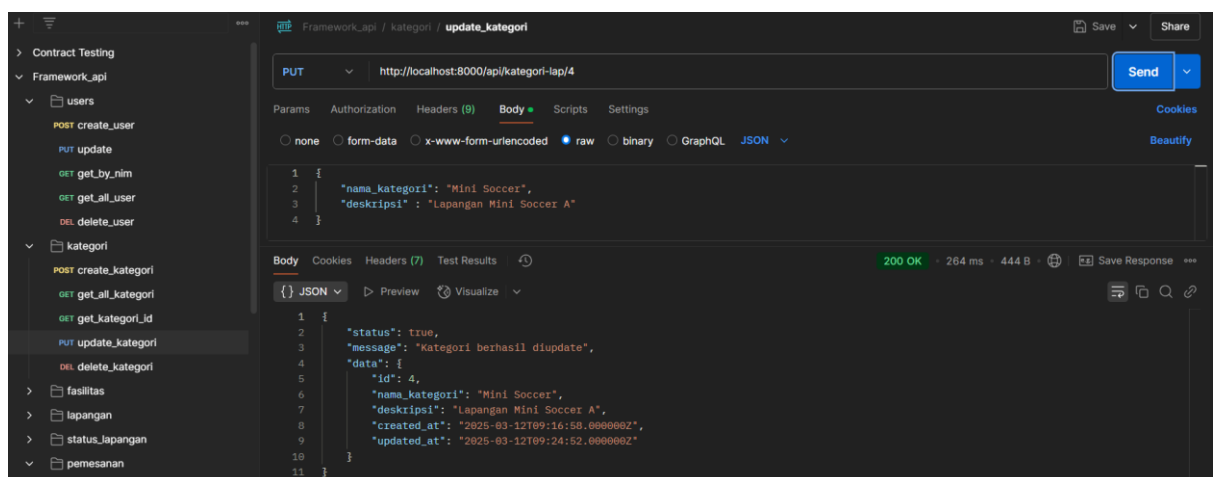
2. Mengambil semua data Kategori Lapangan (GET/api/kategori-lap)



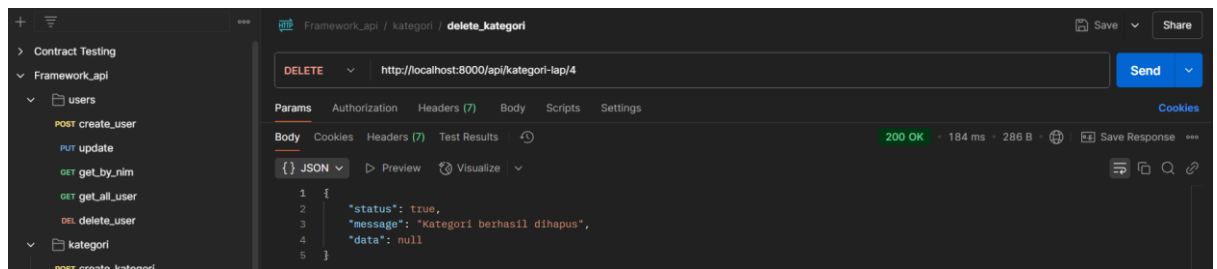
3. Mengambil data berdasarkan id kategori (GET/api/kategori-lap/{id})



4. Update kategori lapangan berdasarkan id (PUT/api/kategori-lap/{id})

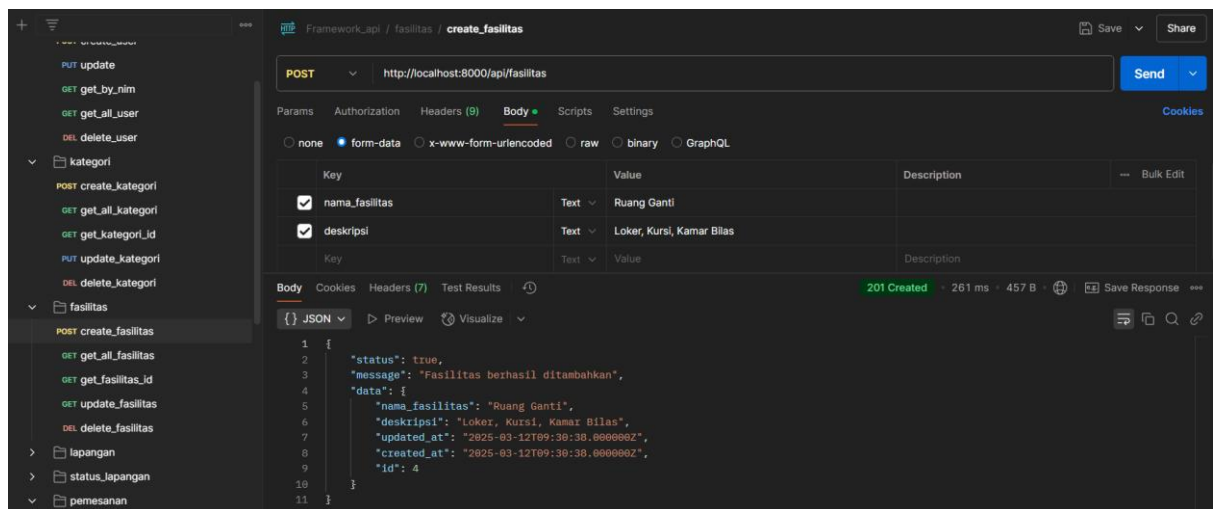


5. Hapus kategori lapangan berdasarkan id (DELETE/api/kategori-lap/{id})

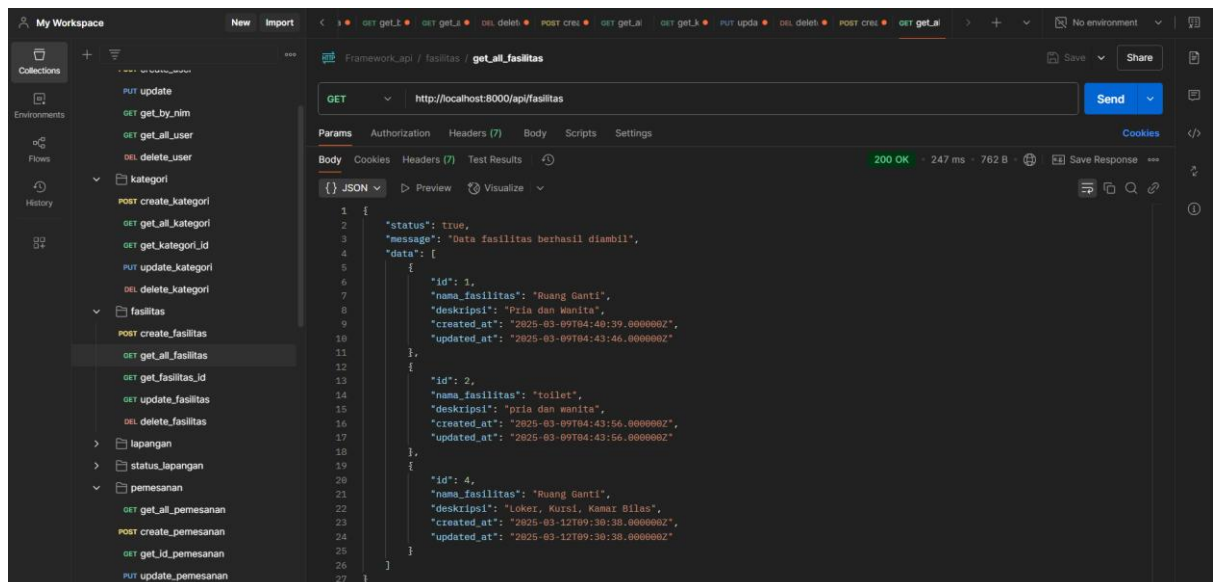


1.3 Dokumentasi API Fasilitas

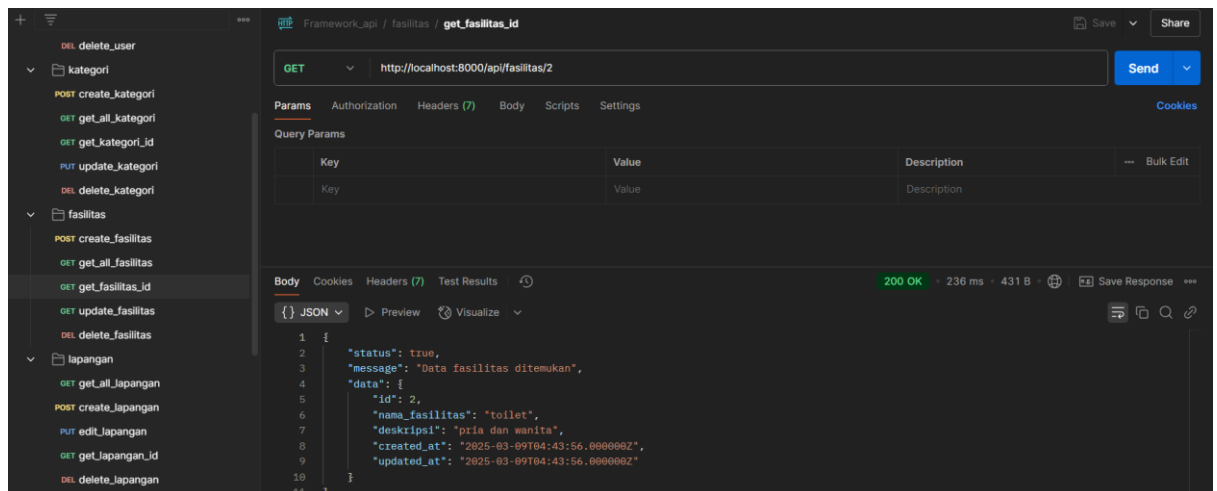
1. Membuat data Fasilitas (POST/api/fasilitas)



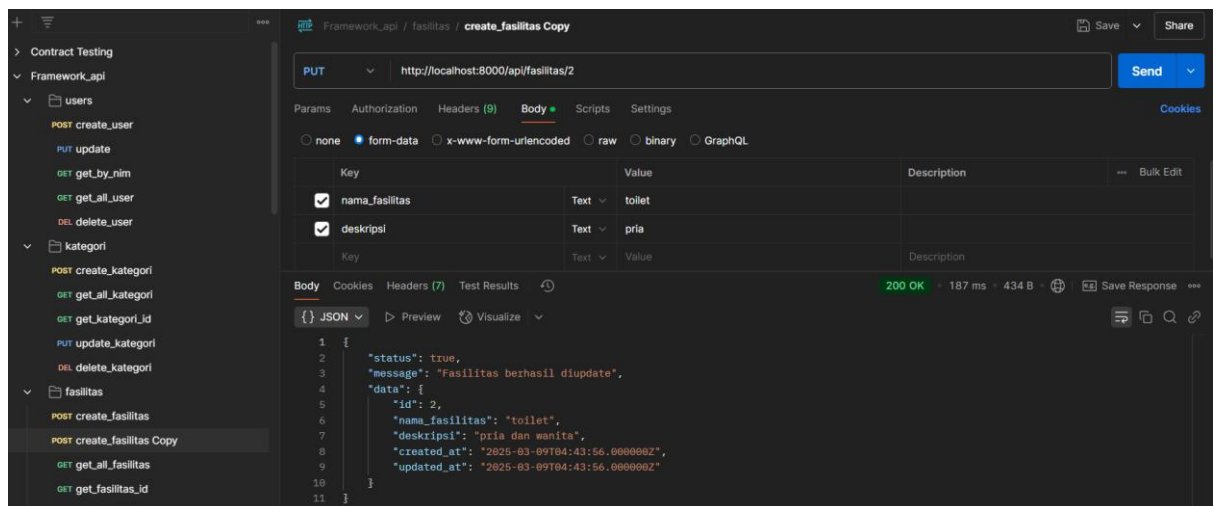
2. Mengambil semua data Fasilitas (GET/api/fasilitas)



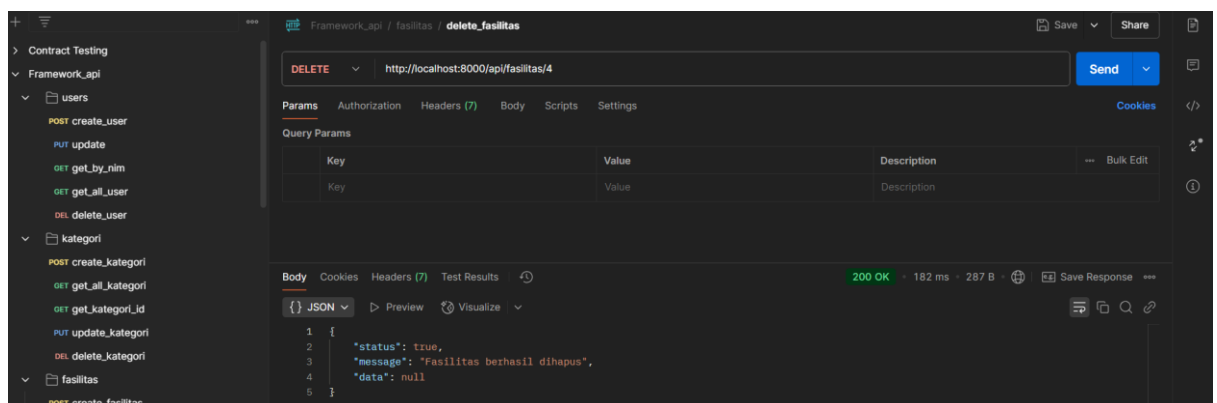
3. Mengambil data berdasarkan id Fasilitas (GET/api/fasilitas/{id})



4. Update data Fasilitas berdasarkan id (PUT/api/fasilitas/{id})



5. Hapus data Fasilitas berdasarkan id (DELETE/api/fasilitas/id)



1.4 Dokumentasi API Sesi

1. Membuat data Sesi → POST /api/sesi

The screenshot shows the Postman interface for a POST request to `http://127.0.0.1:8000/api/sesi`. The request body is in JSON format with the following data:

Key	Value	Description
jam_mulai	09:00	
jam_selesai	10:00	
deskripsi	Sesi 3	

The response is a 201 Created status with the following JSON body:

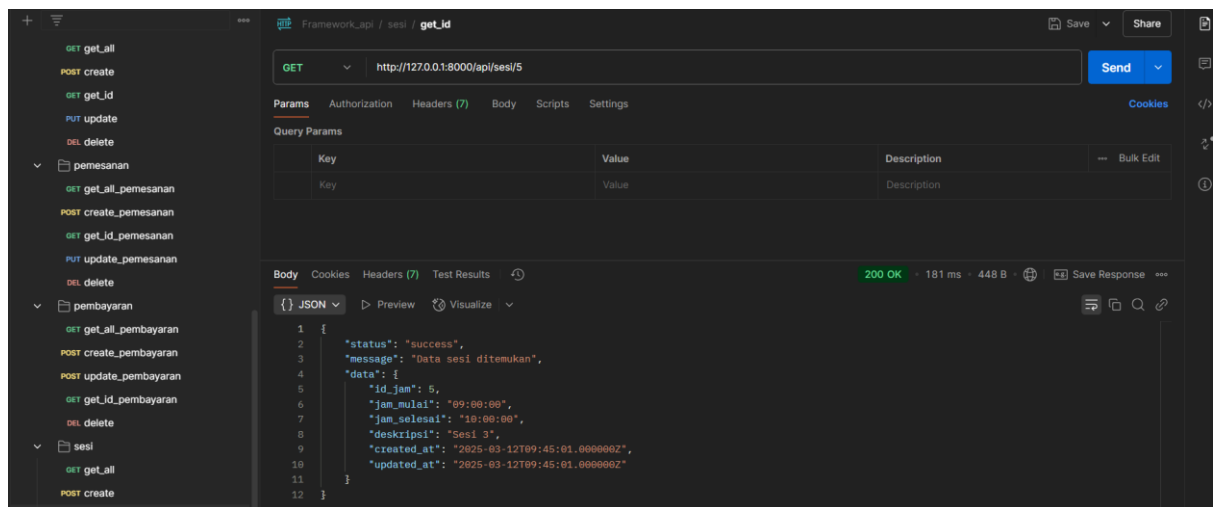
```
{
  "status": "success",
  "message": "Data sesi berhasil ditambahkan",
  "data": {
    "jam_mulai": "09:00",
    "jam_selesai": "10:00",
    "deskripsi": "Sesi 3",
    "updated_at": "2025-03-12T09:45:01.000000Z",
    "created_at": "2025-03-12T09:45:01.000000Z",
    "id_jam": 5
  }
}
```

2. Mengambil semua data Sesi → GET /api/sesi

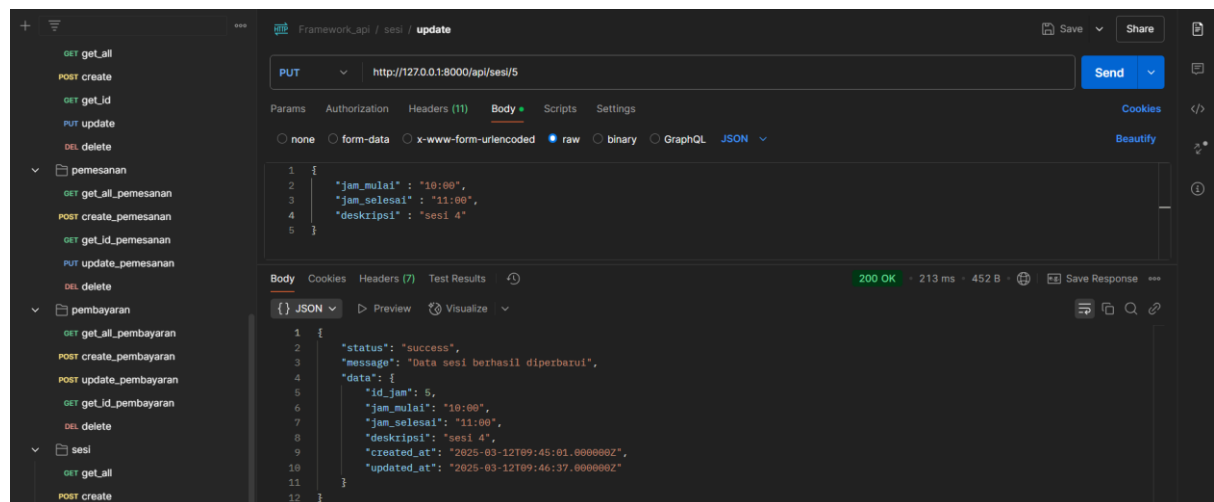
The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:8000/api/sesi`. The response is a 200 OK status with the following JSON body:

```
{
  "status": "success",
  "message": "Data sesi berhasil diambil",
  "data": [
    {
      "id_jam": 1,
      "jam_mulai": "07:00:00",
      "jam_selesai": "08:00:00",
      "deskripsi": "sesi 1",
      "created_at": "2025-03-11T09:19:59.000000Z",
      "updated_at": "2025-03-11T09:27:26.000000Z"
    },
    {
      "id_jam": 4,
      "jam_mulai": "08:00:00",
      "jam_selesai": "09:00:00",
      "deskripsi": "Sesi 2",
      "created_at": null,
      "updated_at": null
    },
    {
      "id_jam": 5,
      "jam_mulai": "09:00:00",
      "jam_selesai": "10:00:00",
      "deskripsi": "Sesi 3",
      "created_at": "2025-03-12T09:45:01.000000Z",
      "updated_at": "2025-03-12T09:45:01.000000Z"
    }
  ]
}
```

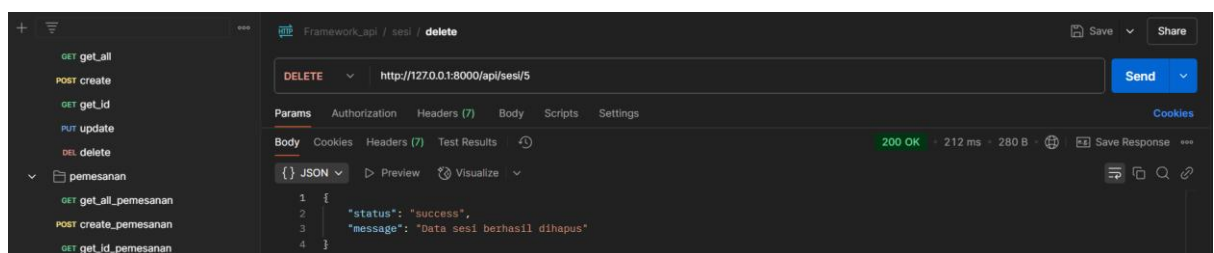
3. Mengambil data berdasarkan ID Sesi → GET /api/sesi/{id}



4. Update data Sesi berdasarkan ID → PUT /api/sesi/{id}

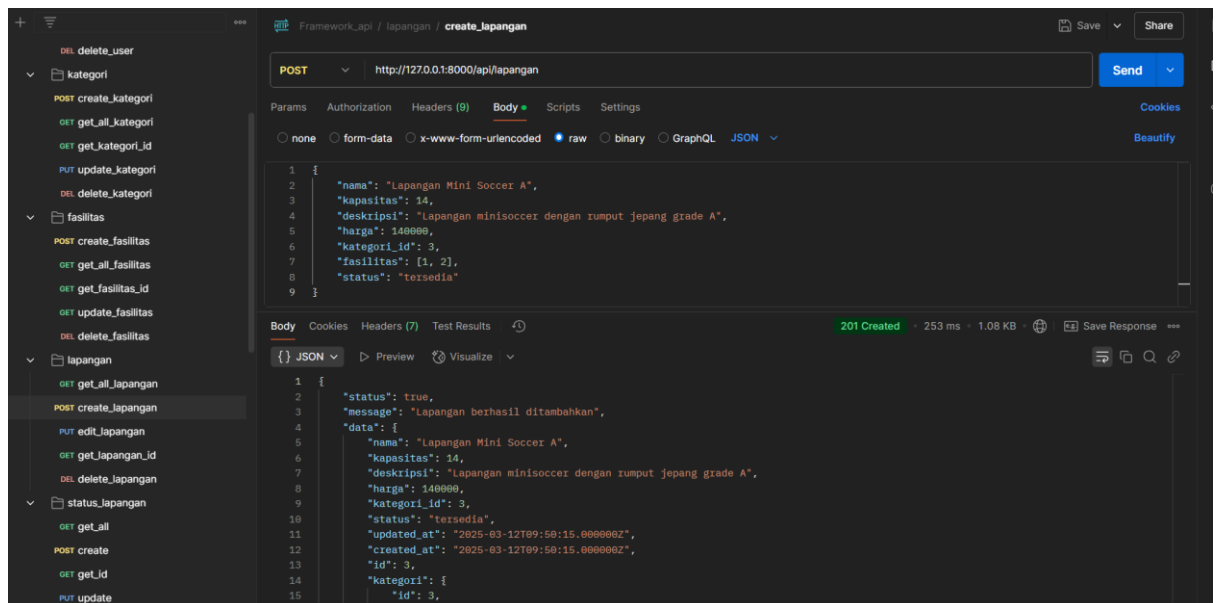


5. Hapus data Sesi berdasarkan ID → DELETE /api/sesi/{id}

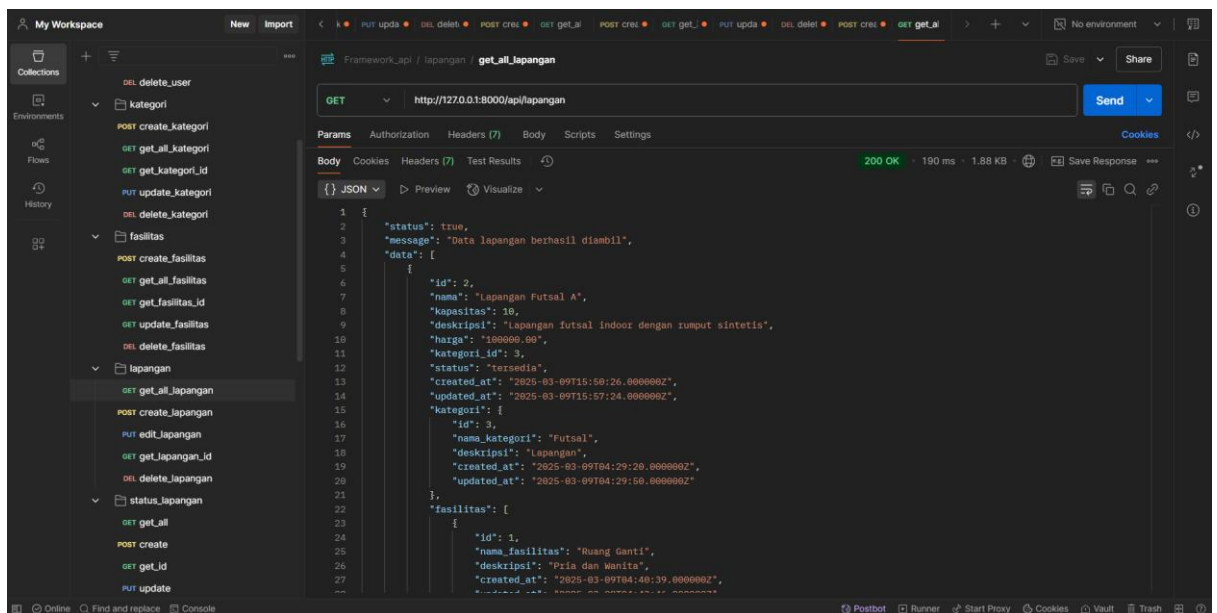


1.5 Dokumentasi API Lapangan

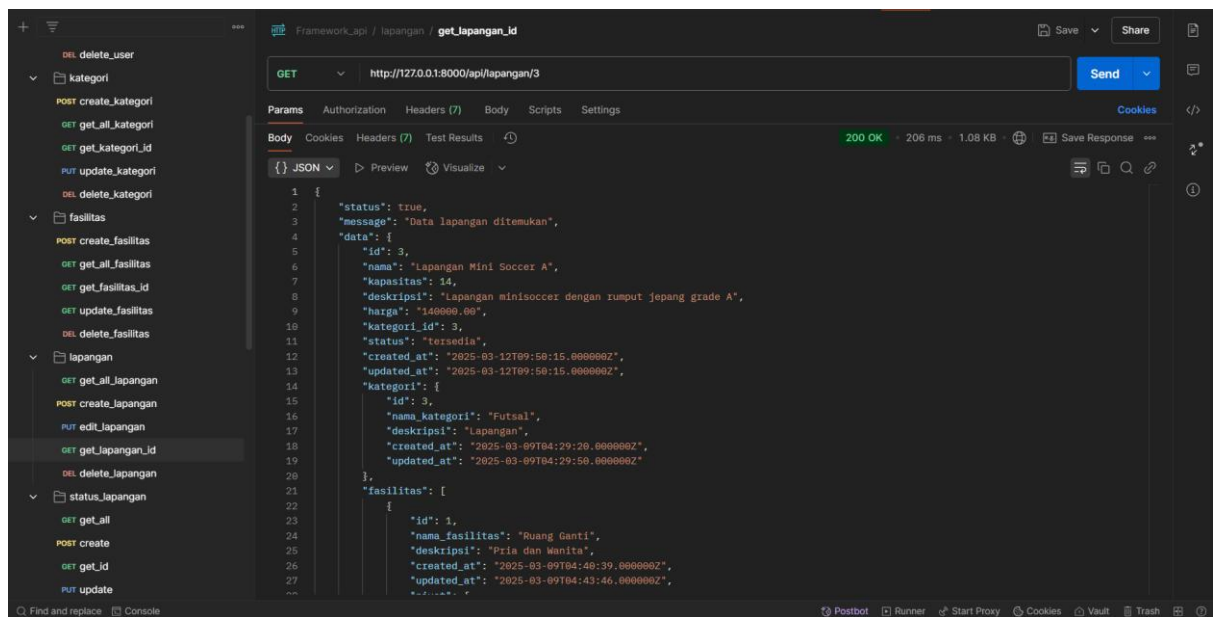
1. Membuat data Lapangan → POST /api/lapangan



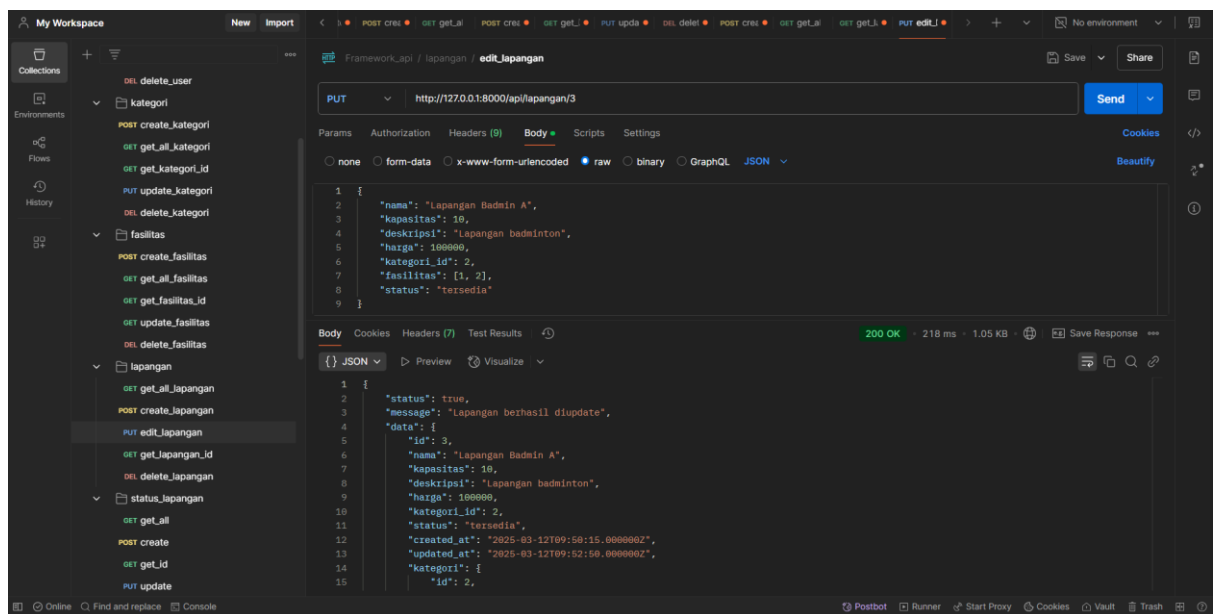
2. Mengambil semua data Lapangan → GET /api/lapangan



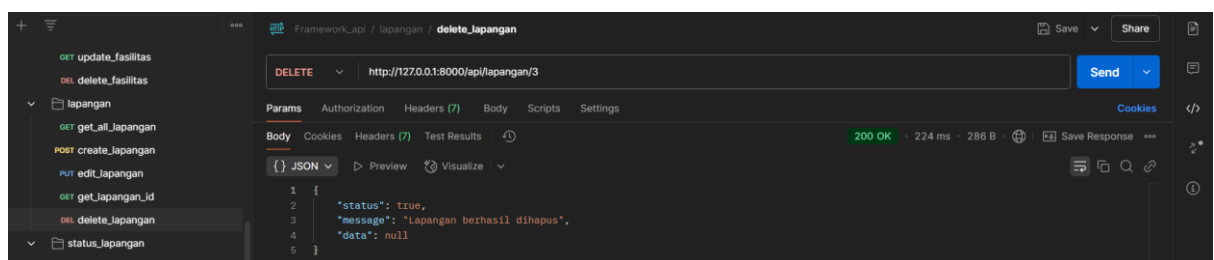
3. Mengambil data berdasarkan ID Lapangan → GET /api/lapangan/{id}



4. Update data Lapangan berdasarkan ID → PUT /api/lapangan/{id}

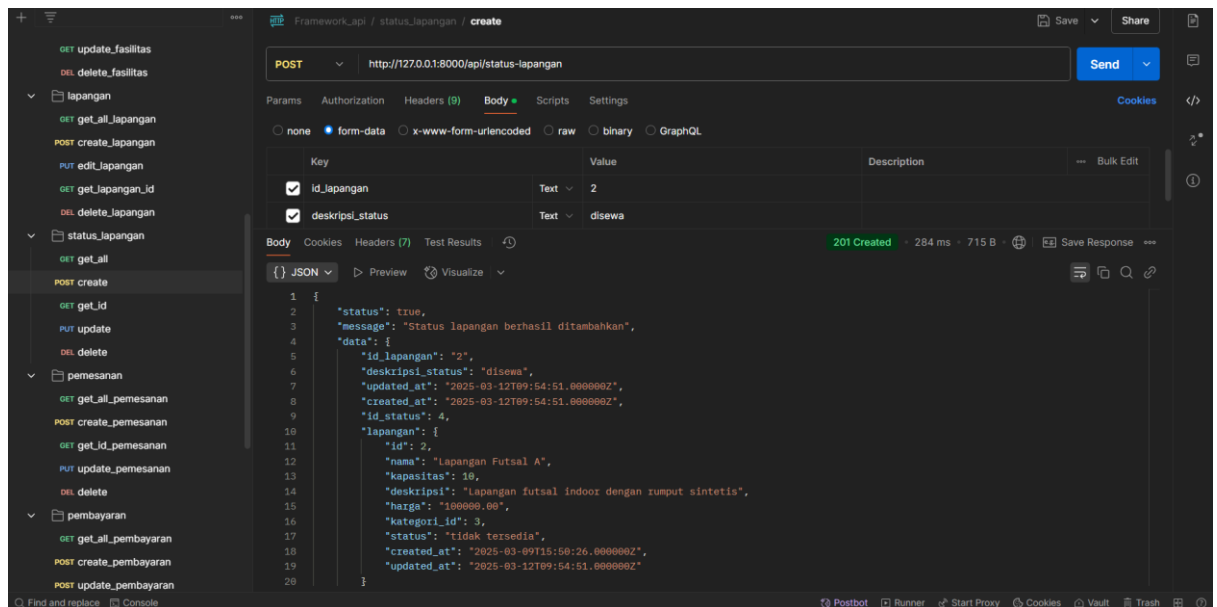


5. Hapus data Lapangan berdasarkan ID → DELETE /api/lapangan/{id}

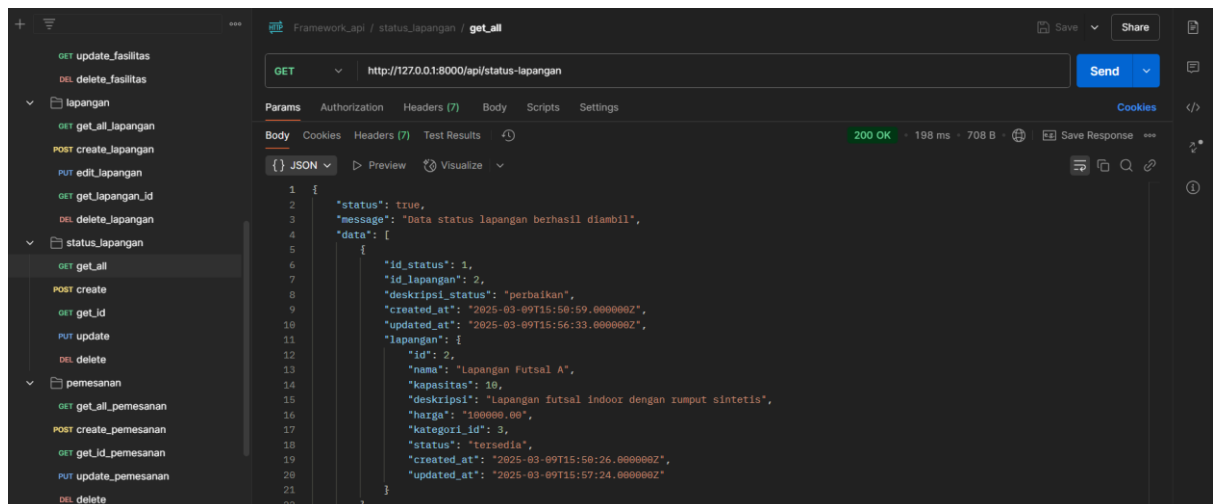


1.6 Dokumentasi API Status Lapangan

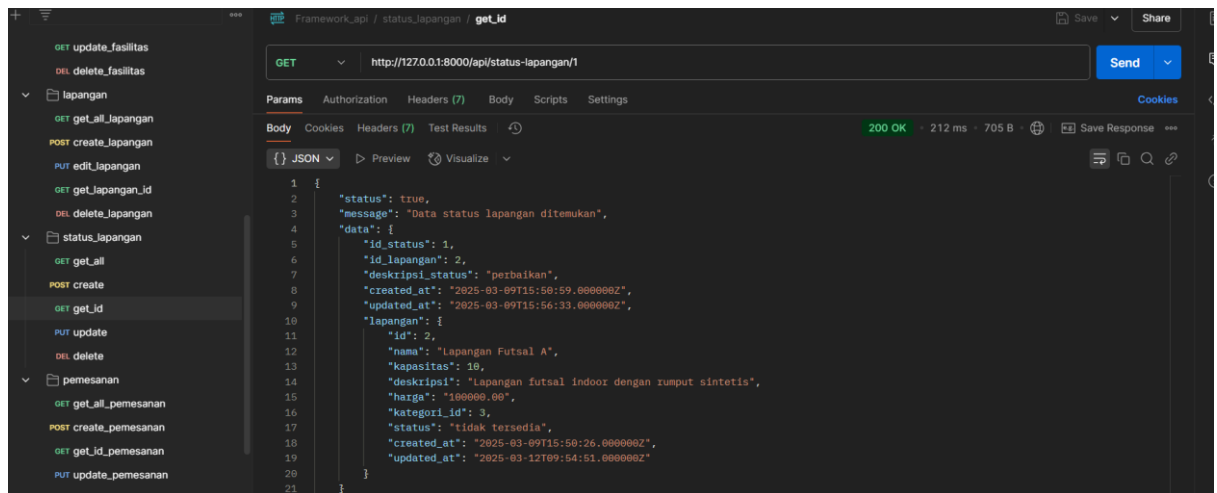
1. Membuat data Status Lapangan → POST /api/status-lapangan



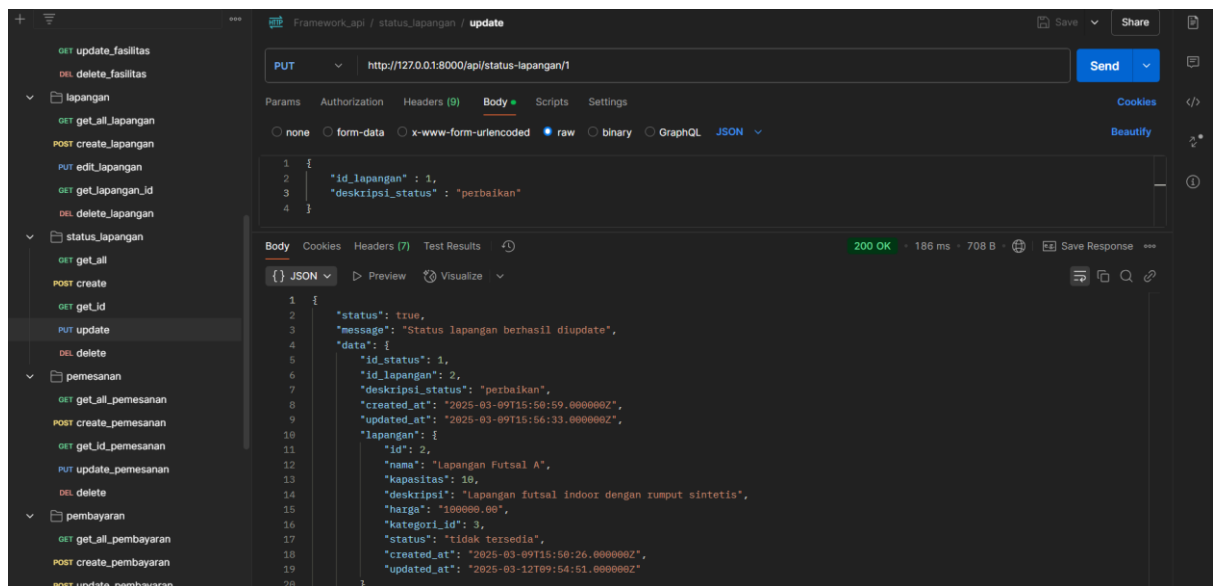
2. Mengambil semua data Status Lapangan → GET /api/status-lapangan



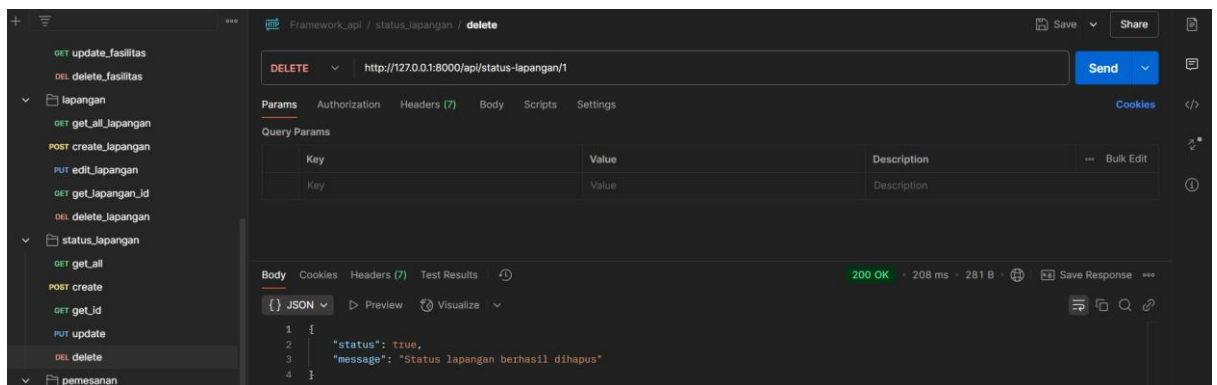
3. Mengambil data berdasarkan ID Status Lapangan → GET /api/status-lapangan/{id}



4. Update data Status Lapangan berdasarkan ID → PUT /api/status-lapangan/{id}



5. Hapus data Status Lapangan berdasarkan ID → DELETE /api/status-lapangan/{id}



1.7 Dokumentasi API Pemesanan

1. Membuat data Pemesanan → POST /api/pemesanan

The screenshot shows the Postman interface for a POST request to `http://127.0.0.1:8000/api/pemesanan`. The request body is a JSON object with the following data:

Key	Value	Description
<input checked="" type="checkbox"/> id_user	4	
<input checked="" type="checkbox"/> id_lapangan	2	
<input checked="" type="checkbox"/> id_hari	1	
<input checked="" type="checkbox"/> sesi	[1, 2]	
<input checked="" type="checkbox"/> status	menunggu verifikasi	

The response is a 201 Created status with the following JSON body:

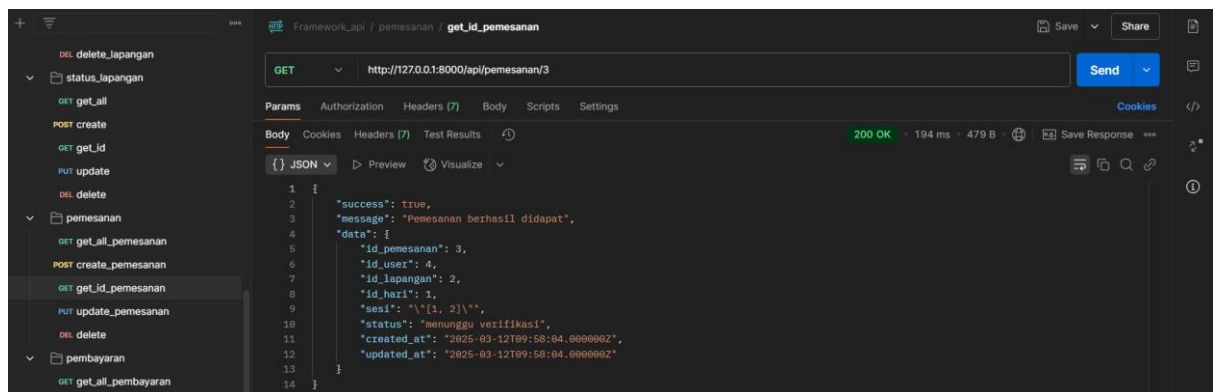
```
1 {
2   "success": true,
3   "message": "Pemesanan berhasil dibuat",
4   "data": {
5     "id_user": "4",
6     "id_lapangan": "2",
7     "id_hari": "1",
8     "sesi": "[1, 2]",
9     "status": "menunggu verifikasi",
10    "updated_at": "2025-03-12T09:58:04.000000Z",
11    "created_at": "2025-03-12T09:58:04.000000Z",
12    "id_pemesanan": 3
13  }
14 }
```

2. Mengambil semua data Pemesanan → GET /api/pemesanan

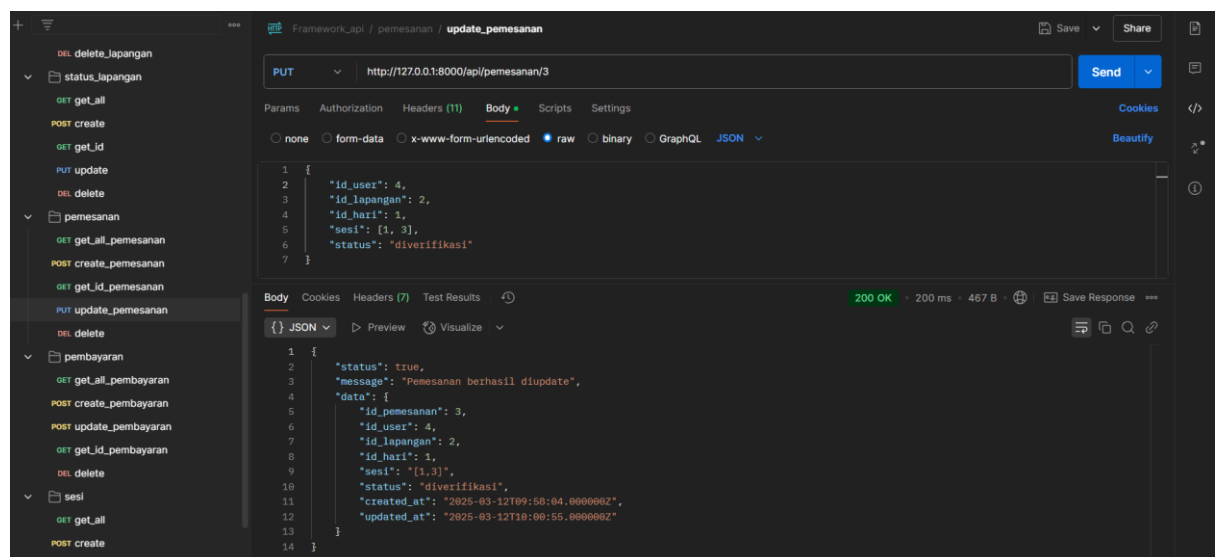
The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:8000/api/pemesanan`. The response is a 200 OK status with the following JSON body:

```
1 {
2   "success": true,
3   "message": "Data Pemesanan berhasil diambil",
4   "data": [
5     {
6       "id_pemesanan": 2,
7       "id_user": 4,
8       "id_lapangan": 2,
9       "id_hari": 1,
10      "sesi": "[3, 4]",
11      "status": "diverifikasi",
12      "created_at": "2025-03-11T02:44:43.000000Z",
13      "updated_at": "2025-03-11T02:57:35.000000Z"
14    }
15  ]
16 }
```

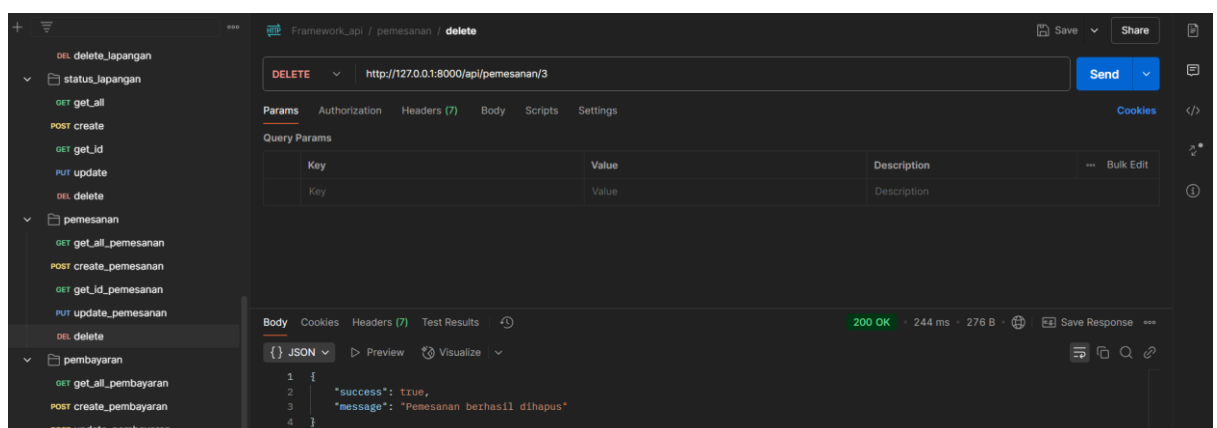
3. Mengambil data berdasarkan ID Pemesanan → GET /api/pemesanan/{id}



4. Update data Pemesanan berdasarkan ID → PUT /api/pemesanan/{id}

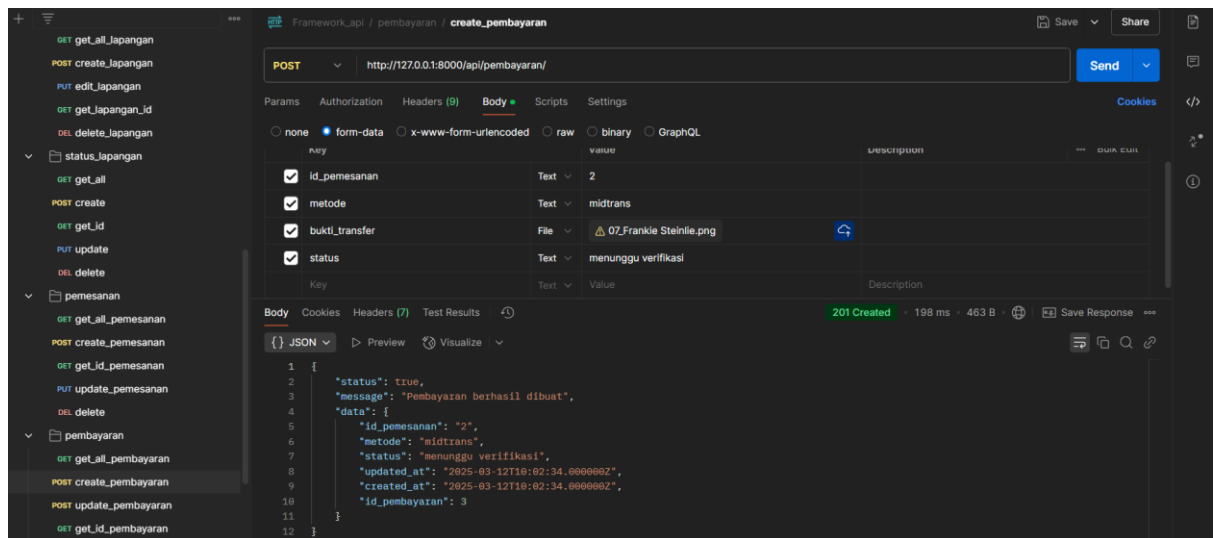


5. Hapus data Pemesanan berdasarkan ID → DELETE /api/pemesanan/{id}

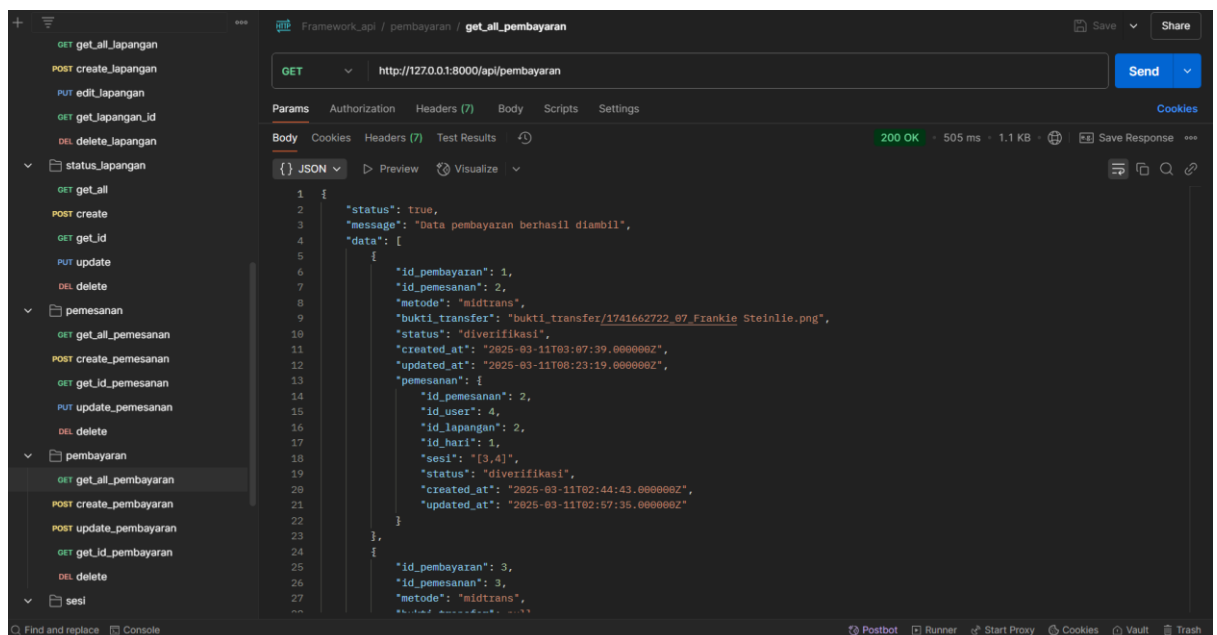


1.8 Dokumentasi API Pembayaran

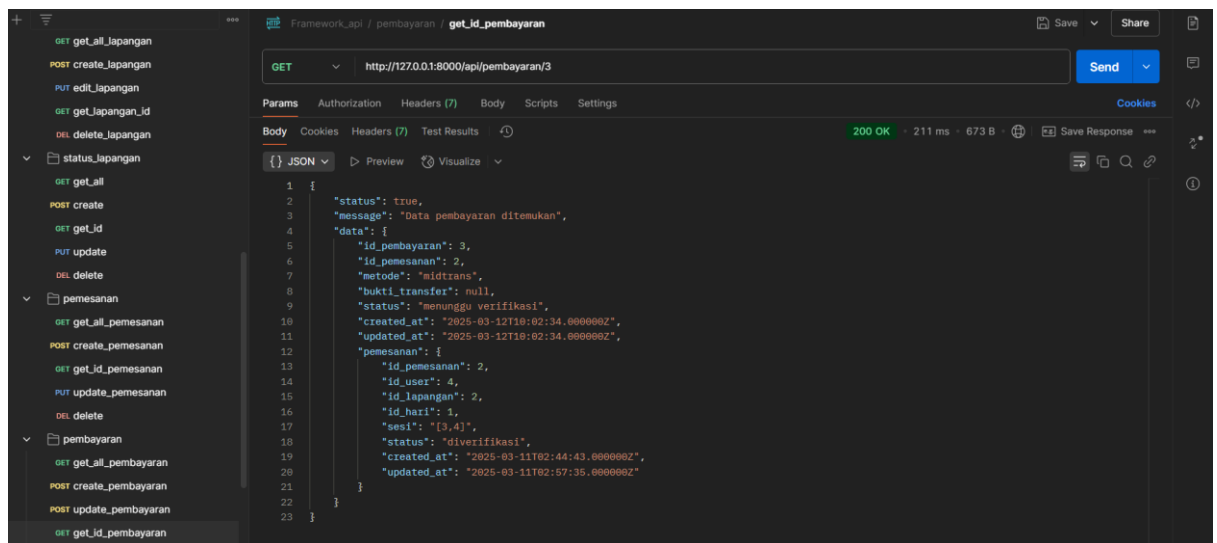
1. Membuat data Pembayaran → POST /api/pembayaran



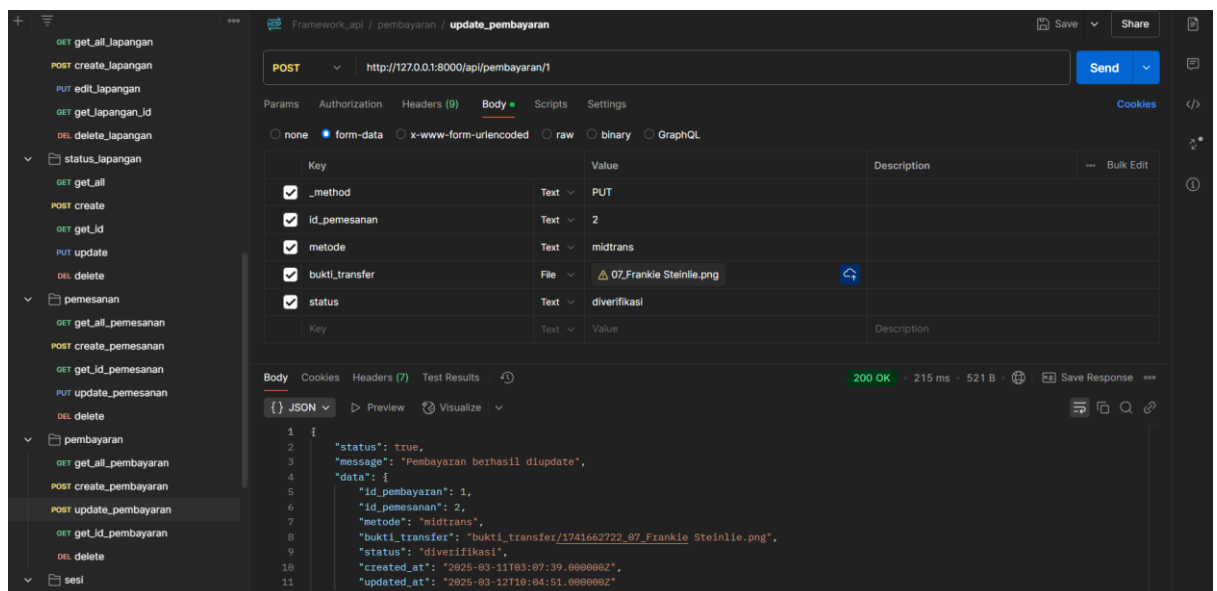
2. Mengambil semua data Pembayaran → GET /api/pembayaran



3. Mengambil data berdasarkan ID Pembayaran → GET /api/pembayaran/{id}



4. Update data Pembayaran berdasarkan ID → PUT /api/pembayaran/{id}



5. Hapus data Pembayaran berdasarkan ID → DELETE /api/pembayaran/{id}

