LAPORAN PRAKTIKUM REST API (CRUD)
MATA KULIAH PERANCANGAN WEBSITE 2

Disusun oleh:

Nama	**: Audrey Tarita Aurellia**
NIM	**: 2402008**
Kelas	**: TI – 3A**
Mata Kuliah	**: Perancangan Website 2**

PROGRAM STUDI D3 TEKNIK INFORMATIKA

POLITEKINK PURBAYA

2025

BAB I
PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi mendorong aplikasi berbasis web untuk semakin mengandalkan komunikasi data yang cepat, terstruktur, dan mudah dikembangkan. Saat ini, sebuah sistem tidak lagi berdiri sendiri, melainkan saling terhubung antara sisi client dan server, bahkan dengan sistem lain yang berbeda platform. Kondisi ini menuntut adanya mekanisme pertukaran data yang efisien, konsisten, dan dapat diakses secara luas.

Berbagai penelitian dalam bidang rekayasa perangkat lunak menunjukkan bahwa pendekatan arsitektur berbasis layanan (service-oriented architecture) menjadi solusi yang efektif untuk kebutuhan tersebut. Salah satu implementasi yang paling banyak digunakan adalah REST API (Representational State Transfer Application Programming Interface). REST API memungkinkan client dan server berkomunikasi melalui protokol HTTP dengan struktur data yang ringan, seperti JSON, sehingga mudah dipahami dan diolah oleh berbagai bahasa pemrograman.

Dalam praktiknya, REST API memanfaatkan metode HTTP seperti GET, POST, PUT, dan DELETE untuk merepresentasikan operasi pengambilan, penambahan, pembaruan, dan penghapusan data. Pendekatan ini dinilai lebih fleksibel dan scalable dibandingkan metode konvensional, karena memisahkan logika backend dari tampilan frontend. Beberapa studi juga menyebutkan bahwa penggunaan REST API dapat meningkatkan keteraturan sistem, mempermudah pemeliharaan kode, serta mendukung pengembangan aplikasi berbasis microservices.

Berdasarkan hal tersebut, praktikum REST API menjadi penting untuk memberikan pemahaman secara langsung mengenai cara kerja pertukaran data antara client dan server. Melalui praktikum ini, mahasiswa tidak hanya mempelajari konsep teoritis, tetapi juga menerapkannya secara nyata dengan menghubungkan REST API ke database dan melakukan pengujian menggunakan tools seperti Postman. Dengan demikian, mahasiswa diharapkan mampu memahami alur kerja REST API secara menyeluruh serta siap menerapkannya pada pengembangan sistem informasi berbasis web.

1.2. Tujuan Praktikum

Praktikum ini bertujuan untuk memberikan pemahaman praktis mengenai penerapan REST API dalam pengembangan aplikasi berbasis web. Melalui kegiatan ini, mahasiswa diharapkan mampu memahami bagaimana proses pertukaran data antara client dan server dilakukan menggunakan protokol HTTP. Selain itu, praktikum ini bertujuan untuk melatih mahasiswa dalam menerapkan metode HTTP seperti GET, POST, PUT, dan DELETE sesuai dengan fungsi masing-masing dalam pengelolaan data.

Tujuan lainnya adalah agar mahasiswa dapat mengintegrasikan REST API dengan database MySQL serta melakukan pengujian endpoint API menggunakan tools Postman. Dengan demikian, mahasiswa tidak hanya memahami konsep REST API secara teoritis, tetapi juga mampu mengimplementasikannya secara langsung dalam sebuah sistem yang terhubung dengan basis data.

1.3. Manfaat Praktikum

Adapun manfaat yang diperoleh dari pelaksanaan praktikum REST API ini antara lain sebagai berikut:

A. Mahasiswa memperoleh pemahaman yang lebih mendalam mengenai konsep komunikasi data client–server berbasis REST API.
B. Mahasiswa mampu mengimplementasikan operasi CRUD (Create, Read, Update, Delete) menggunakan metode HTTP secara tepat.
C. Mahasiswa terbiasa melakukan pengujian dan validasi REST API menggunakan Postman sebelum API digunakan pada sistem yang lebih besar.
D. Praktikum ini dapat menjadi dasar bagi mahasiswa dalam mengembangkan aplikasi web yang terstruktur, scalable, dan mudah dikembangkan di masa mendatang.

BAB II
LANDASAN TEORI

2.1. REST API (Representational State Transfer Application Programming Interface)

REST API merupakan sebuah pendekatan arsitektur yang digunakan untuk membangun layanan berbasis web yang memungkinkan terjadinya pertukaran data antara client dan server. Konsep REST pertama kali diperkenalkan untuk menciptakan sistem yang sederhana, ringan, dan mudah dikembangkan tanpa bergantung pada teknologi tertentu. Dalam implementasinya, REST API memanfaatkan protokol HTTP sebagai media komunikasi utama.

REST API bekerja dengan merepresentasikan sumber daya (resource) dalam bentuk URL dan menggunakan format data yang umum seperti JSON. Setiap resource dapat diakses dan dikelola melalui metode HTTP yang sesuai, sehingga client dapat berinteraksi dengan server tanpa perlu mengetahui detail implementasi di sisi backend. Pendekatan ini dinilai efektif karena mendukung interoperabilitas antar sistem serta memudahkan integrasi dengan berbagai platform.

2.2. Metode HTTP pada REST API

Metode HTTP merupakan komponen penting dalam REST API karena menentukan jenis operasi yang dilakukan terhadap data. Beberapa metode HTTP yang umum digunakan dalam REST API antara lain sebagai berikut:

A. **GET**

Metode GET digunakan untuk mengambil atau membaca data dari server. Operasi ini tidak mengubah data yang ada di database dan hanya bersifat menampilkan informasi.

B. **POST**

Metode POST digunakan untuk mengirim atau menambahkan data baru ke server. Data dikirim melalui body request dalam format tertentu, seperti JSON, kemudian diproses dan disimpan ke database.

C. **PUT**

Metode PUT digunakan untuk memperbarui data yang sudah ada. Proses

pembaruan dilakukan berdasarkan identitas data tertentu, seperti ID, sehingga data yang diubah tetap terkontrol.

D. **DELETE**

Metode DELETE digunakan untuk menghapus data dari server. Data yang dihapus biasanya ditentukan berdasarkan ID agar tidak terjadi kesalahan penghapusan data.

Penggunaan metode HTTP yang tepat membantu menjaga konsistensi dan kejelasan fungsi REST API dalam pengelolaan data.

2.3. Format Data JSON

JSON (JavaScript Object Notation) merupakan format pertukaran data yang banyak digunakan dalam REST API karena bersifat ringan, mudah dibaca, dan didukung oleh berbagai bahasa pemrograman. Struktur JSON yang sederhana memungkinkan client dan server untuk saling bertukar data secara efisien tanpa membutuhkan proses parsing yang kompleks.

Dalam REST API, JSON sering digunakan untuk mengirim request maupun menerima response. Hal ini menjadikan JSON sebagai standar format data yang efektif dalam pengembangan aplikasi berbasis web modern.

2.4. Postman sebagai Alat Pengujian REST API

Postman adalah sebuah aplikasi yang digunakan untuk menguji dan mengelola REST API. Dengan Postman, pengembang dapat mengirim request menggunakan berbagai metode HTTP serta melihat response yang diberikan oleh server secara langsung. Tool ini sangat membantu dalam proses pengembangan dan debugging API karena memungkinkan pengujian endpoint dilakukan tanpa perlu membuat antarmuka aplikasi terlebih dahulu.

Penggunaan Postman juga mempermudah pengelompokan request ke dalam collection, sehingga pengujian API dapat dilakukan secara terstruktur dan berulang. Oleh karena itu, Postman sering digunakan sebagai alat bantu utama dalam pembelajaran dan pengujian REST API.

BAB III
ALAT DAN BAHAN

3.1. Perangkat Lunak

Perangkat lunak yang digunakan dalam praktikum REST API ini berfungsi untuk mendukung proses pengembangan, pengujian, serta pengelolaan data. Adapun perangkat lunak yang digunakan adalah sebagai berikut:

A. XAMPP: XAMPP digunakan sebagai server lokal yang menyediakan layanan Apache dan MySQL. Apache berperan sebagai web server untuk menjalankan file PHP, sedangkan MySQL digunakan sebagai sistem manajemen basis data untuk menyimpan data pengguna.
B. PHP: PHP digunakan sebagai bahasa pemrograman server-side untuk membangun REST API. PHP berfungsi untuk menangani request dari client, memproses data, serta berkomunikasi dengan database.
C. MySQL: MySQL digunakan sebagai basis data untuk menyimpan data pengguna. Database ini terintegrasi langsung dengan REST API sehingga memungkinkan proses pengambilan, penambahan, pembaruan, dan penghapusan data.
D. Postman: Postman digunakan sebagai alat untuk menguji REST API. Melalui Postman, pengujian metode HTTP seperti GET, POST, PUT, dan DELETE dapat dilakukan dengan mudah serta response dari server dapat diamati secara langsung.
E. Web Browser: Web browser digunakan untuk mengakses server lokal dan memastikan API dapat dijalankan dengan baik melalui alamat localhost.

3.2. Perangkat Keras

Perangkat keras yang digunakan dalam praktikum ini adalah sebuah laptop atau komputer yang mendukung proses pengembangan aplikasi web. Spesifikasi perangkat keras disesuaikan dengan kebutuhan pengembangan sederhana, sehingga tidak memerlukan spesifikasi tinggi.

BAB IV
PERANCANGAN SISTEM

4.1. Gambaran Umum Sistem

Sistem yang dirancang pada praktikum ini berupa layanan **REST API** yang digunakan untuk mengelola data pengguna. REST API berperan sebagai penghubung antara client dan database, sehingga client dapat melakukan proses pengambilan, penambahan, pembaruan, dan penghapusan data tanpa harus berinteraksi langsung dengan database.

Pada sistem ini, client mengirimkan request ke server melalui endpoint REST API menggunakan metode HTTP tertentu. Server kemudian memproses request tersebut menggunakan bahasa pemrograman PHP dan berinteraksi dengan database MySQL yang dikelola melalui phpMyAdmin. Hasil pemrosesan selanjutnya dikirim kembali ke client dalam bentuk response berformat JSON.

4.2. Perancangan Database

Database yang digunakan dalam sistem ini menyimpan data pengguna yang dibutuhkan oleh REST API. Struktur database dirancang agar mendukung proses CRUD (Create, Read, Update, Delete) secara efektif. Tabel utama yang digunakan adalah tabel pengguna dengan beberapa atribut penting, antara lain:

- id_pengguna
- nama
- username
- password
- email
- role
- status
- tanggal_dibuat

Setiap data pengguna memiliki `id_pengguna` sebagai primary key yang berfungsi sebagai identitas unik. Kolom `username` dibuat unik untuk mencegah

terjadinya duplikasi data pengguna, sedangkan kolom `password` disimpan dalam bentuk terenkripsi demi menjaga keamanan data.

4.3. Perancangan REST API Endpoint

REST API pada sistem ini dirancang menggunakan beberapa endpoint yang masing-masing memiliki fungsi berbeda sesuai dengan metode HTTP yang digunakan. Perancangan endpoint bertujuan agar setiap operasi data dapat dilakukan secara terpisah dan terstruktur. Adapun endpoint REST API yang dirancang adalah sebagai berikut:

A. GET

   Digunakan untuk mengambil seluruh data pengguna dari database dan menampilkannya dalam format JSON.

B. POST

   Digunakan untuk menambahkan data pengguna baru ke dalam database dengan data yang dikirim melalui body request.

C. PUT

   Digunakan untuk memperbarui data pengguna yang sudah ada berdasarkan ID pengguna.

D. DELETE

   Digunakan untuk menghapus data pengguna dari database berdasarkan ID pengguna.

Dengan pemisahan fungsi pada masing-masing endpoint, sistem menjadi lebih mudah dipahami dan dikembangkan.

4.4. Perancangan Alur Sistem

Alur kerja sistem dimulai ketika client mengirimkan request ke server melalui Postman. Request tersebut diterima oleh REST API dan diproses sesuai metode HTTP yang digunakan. Server kemudian melakukan validasi data sebelum berinteraksi

dengan database. Setelah proses selesai, server mengirimkan response kembali ke client dalam format JSON.

Perancangan alur ini bertujuan untuk memastikan setiap request diproses secara sistematis serta meminimalkan kesalahan dalam pengelolaan data. Selain itu, alur sistem ini memudahkan proses debugging dan pengujian REST API.

BAB V
IMPLEMENTASI DAN PENGUJIAN

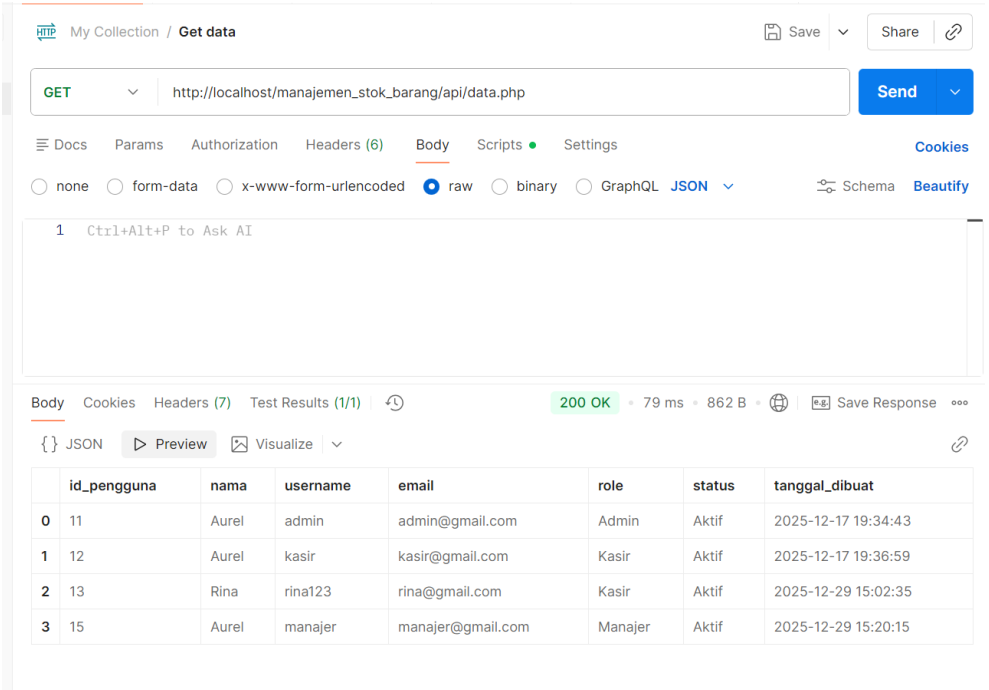
5.1. Implementasi REST API

Implementasi REST API pada praktikum ini dilakukan menggunakan bahasa pemrograman PHP yang dijalankan pada server lokal XAMPP. REST API berfungsi sebagai penghubung antara client dan database untuk mengelola data pengguna. Seluruh endpoint REST API dirancang untuk mendukung operasi CRUD (Create, Read, Update, Delete) dan mengembalikan response dalam format JSON.

Struktur file REST API ditempatkan pada folder `api` di dalam direktori project, sehingga setiap endpoint dapat diakses melalui URL `localhost`. Implementasi ini memudahkan proses pengujian serta pengembangan API secara terpisah dari antarmuka aplikasi.

5.2. Pengujian REST API Metode GET

Pengujian metode GET dilakukan untuk memastikan REST API mampu mengambil data pengguna dari database dengan benar. Metode GET digunakan untuk menampilkan seluruh data pengguna yang tersimpan di dalam tabel `pengguna`.

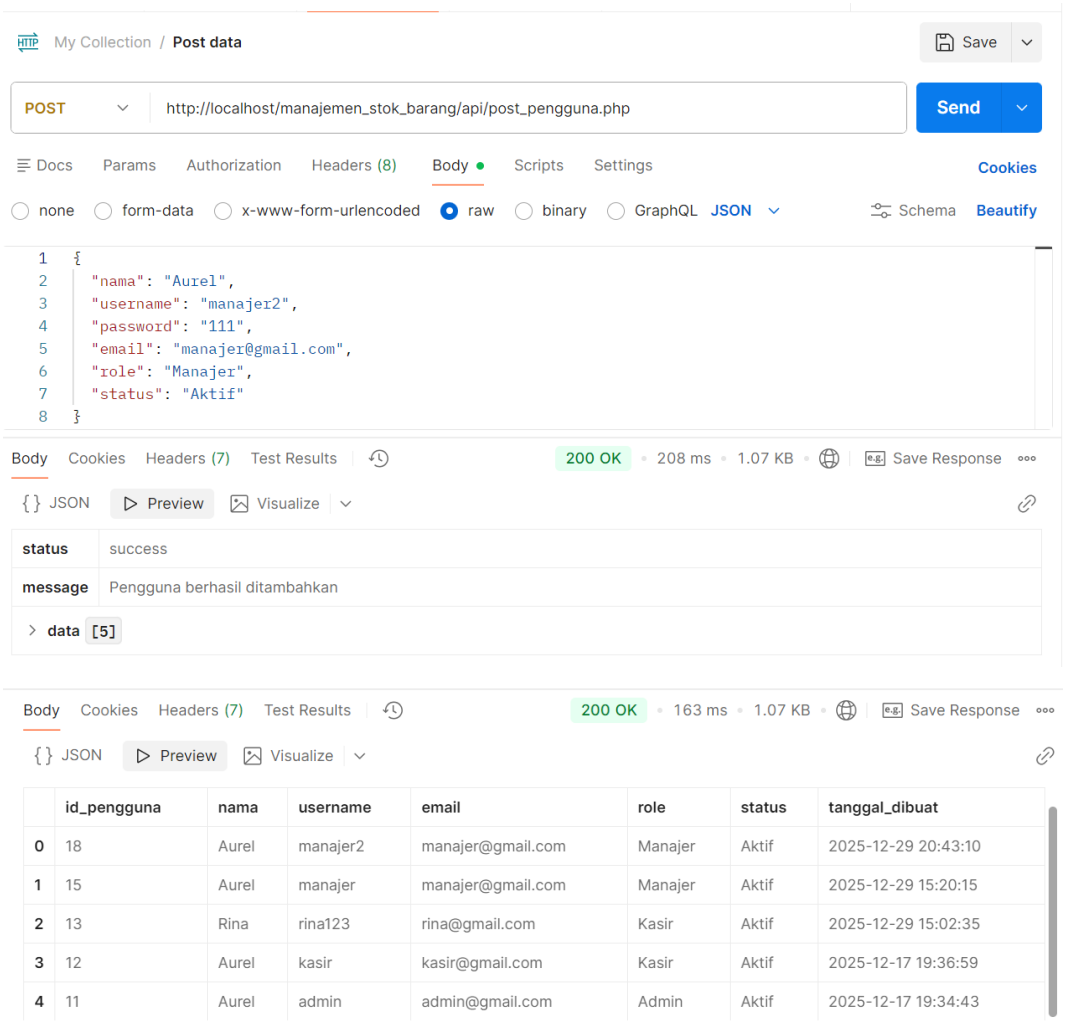


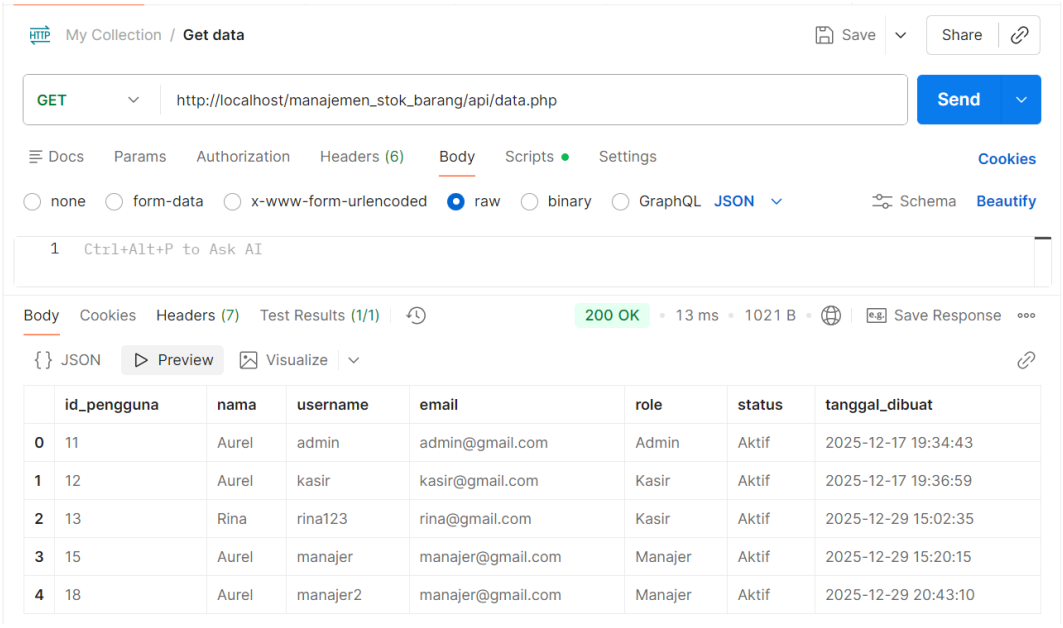
	id_pengguna	nama	username	email	role	status	tanggal_dibuat
0	11	Aurel	admin	admin@gmail.com	Admin	Aktif	2025-12-17 19:34:43
1	12	Aurel	kasir	kasir@gmail.com	Kasir	Aktif	2025-12-17 19:36:59
2	13	Rina	rina123	rina@gmail.com	Kasir	Aktif	2025-12-29 15:02:35
3	15	Aurel	manajer	manajer@gmail.com	Manajer	Aktif	2025-12-29 15:20:15

Pengujian dilakukan menggunakan aplikasi Postman dengan mengirimkan request GET ke endpoint yang telah disediakan. Hasil pengujian menunjukkan bahwa REST API berhasil menampilkan data pengguna dalam format JSON dengan status response berhasil (200 OK).

5.3. Pengujian REST API Metode POST

Metode POST digunakan untuk menambahkan data pengguna baru ke dalam database. Data pengguna dikirim melalui body request dalam format JSON yang berisi informasi seperti nama, username, password, email, role, dan status.



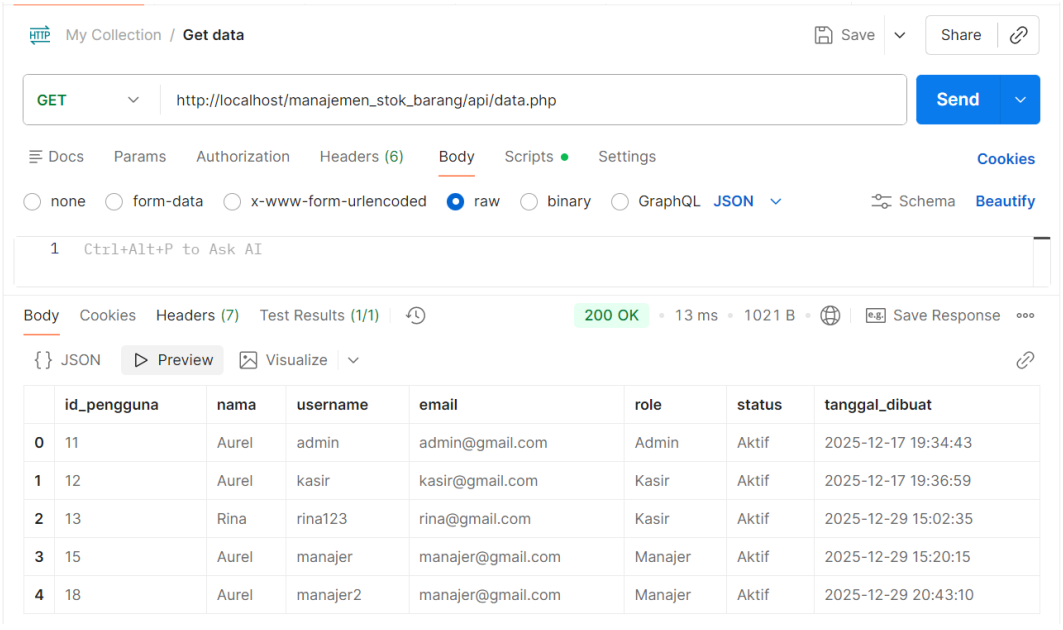


	id_pengguna	nama	username	email	role	status	tanggal_dibuat
0	11	Aurel	admin	admin@gmail.com	Admin	Aktif	2025-12-17 19:34:43
1	12	Aurel	kasir	kasir@gmail.com	Kasir	Aktif	2025-12-17 19:36:59
2	13	Rina	rina123	rina@gmail.com	Kasir	Aktif	2025-12-29 15:02:35
3	15	Aurel	manajer	manajer@gmail.com	Manajer	Aktif	2025-12-29 15:20:15
4	18	Aurel	manajer2	manajer@gmail.com	Manajer	Aktif	2025-12-29 20:43:10

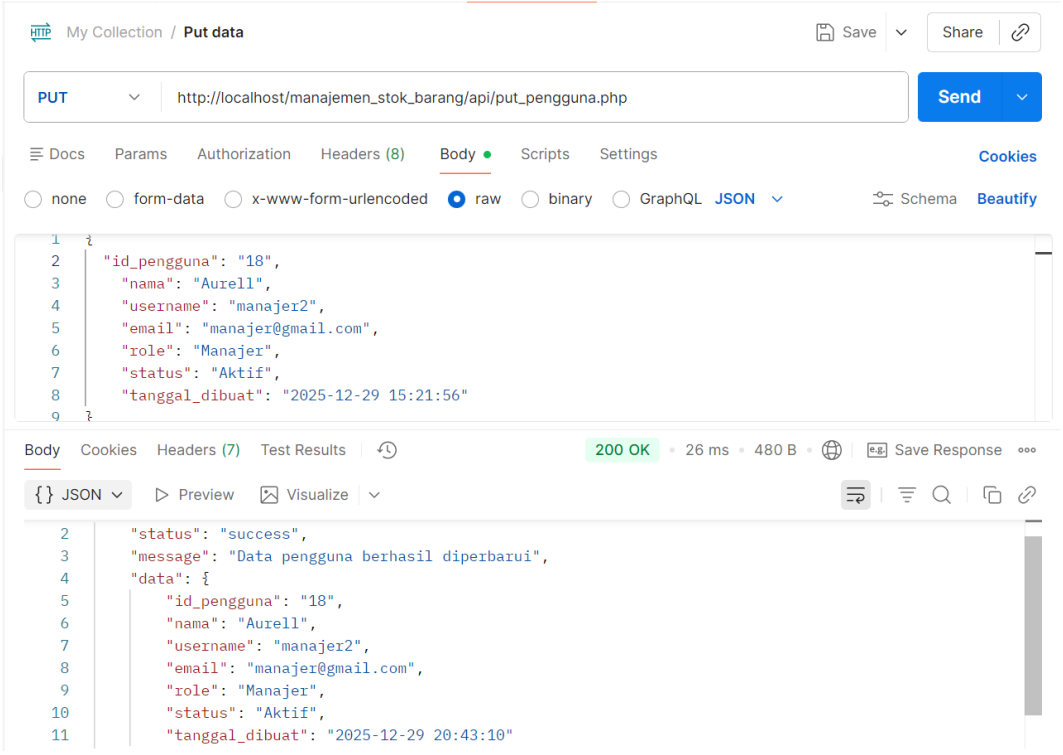
Pada tahap pengujian, REST API mampu menerima data dengan baik, melakukan validasi data, serta menyimpan data ke dalam database. Setelah proses penyimpanan berhasil, server mengembalikan response berupa pesan keberhasilan. Pengujian ini membuktikan bahwa REST API dapat menjalankan proses input data secara benar.

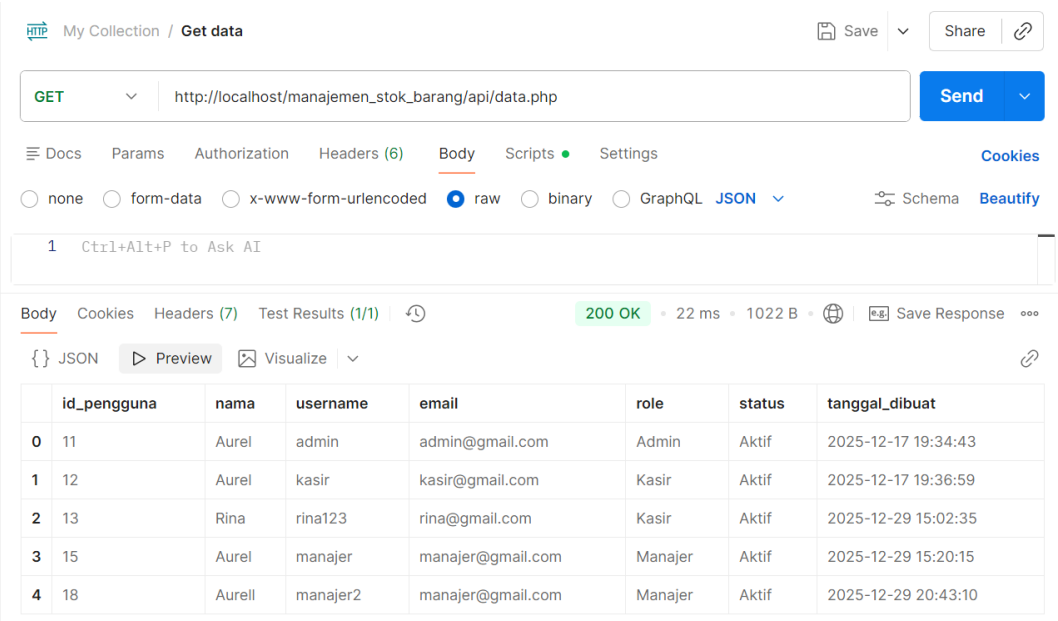
5.4. Pengujian REST API Metode PUT

Pengujian metode PUT dilakukan untuk memperbarui data pengguna yang sudah ada di dalam database. Proses pembaruan dilakukan berdasarkan ID pengguna yang dikirimkan melalui body request.



	id_pengguna	nama	username	email	role	status	tanggal_dibuat
0	11	Aurel	admin	admin@gmail.com	Admin	Aktif	2025-12-17 19:34:43
1	12	Aurel	kasir	kasir@gmail.com	Kasir	Aktif	2025-12-17 19:36:59
2	13	Rina	rina123	rina@gmail.com	Kasir	Aktif	2025-12-29 15:02:35
3	15	Aurel	manajer	manajer@gmail.com	Manajer	Aktif	2025-12-29 15:20:15
4	18	Aurel	manajer2	manajer@gmail.com	Manajer	Aktif	2025-12-29 20:43:10



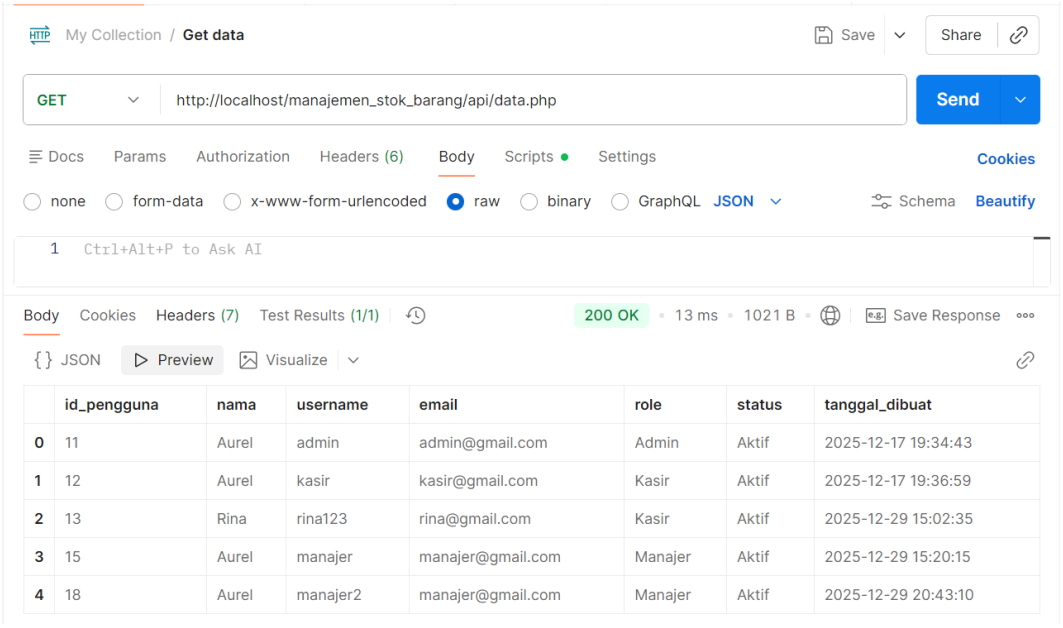


	id_pengguna	nama	username	email	role	status	tanggal_dibuat
0	11	Aurel	admin	admin@gmail.com	Admin	Aktif	2025-12-17 19:34:43
1	12	Aurel	kasir	kasir@gmail.com	Kasir	Aktif	2025-12-17 19:36:59
2	13	Rina	rina123	rina@gmail.com	Kasir	Aktif	2025-12-29 15:02:35
3	15	Aurel	manajer	manajer@gmail.com	Manajer	Aktif	2025-12-29 15:20:15
4	18	Aurell	manajer2	manajer@gmail.com	Manajer	Aktif	2025-12-29 20:43:10

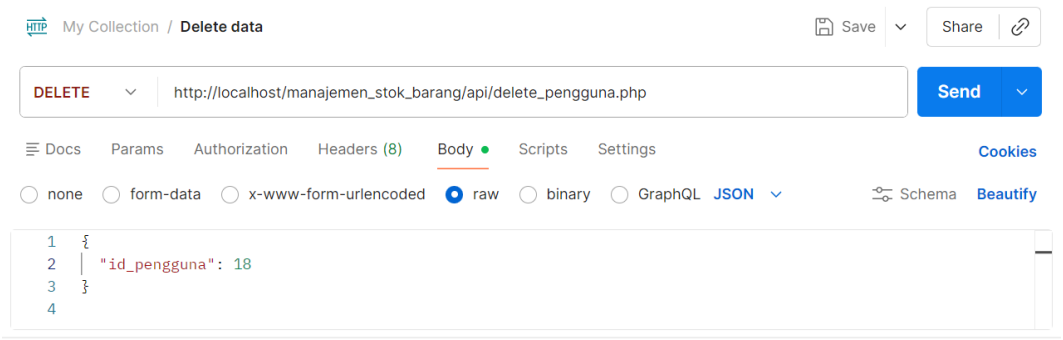
Hasil pengujian menunjukkan bahwa REST API berhasil memperbarui data pengguna sesuai dengan perubahan yang dikirimkan, seperti perubahan username, email, atau role. Server juga mengembalikan data terbaru sebagai response untuk memastikan bahwa proses update berjalan dengan baik.

5.5. Pengujian REST API Metode DELETE

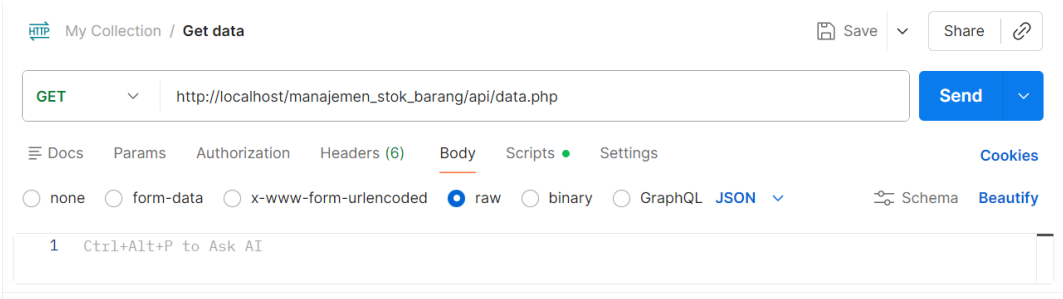
Metode DELETE digunakan untuk menghapus data pengguna dari database berdasarkan ID pengguna. Pengujian dilakukan dengan mengirimkan request DELETE melalui Postman dengan menyertakan ID pengguna yang akan dihapus.



	id_pengguna	nama	username	email	role	status	tanggal_dibuat
0	11	Aurel	admin	admin@gmail.com	Admin	Aktif	2025-12-17 19:34:43
1	12	Aurel	kasir	kasir@gmail.com	Kasir	Aktif	2025-12-17 19:36:59
2	13	Rina	rina123	rina@gmail.com	Kasir	Aktif	2025-12-29 15:02:35
3	15	Aurel	manajer	manajer@gmail.com	Manajer	Aktif	2025-12-29 15:20:15
4	18	Aurel	manajer2	manajer@gmail.com	Manajer	Aktif	2025-12-29 20:43:10



status	success
message	Pengguna berhasil dihapus



	id_pengguna	nama	username	email	role	status	tanggal_dibuat
0	11	Aurel	admin	admin@gmail.com	Admin	Aktif	2025-12-17 19:34:43
1	12	Aurel	kasir	kasir@gmail.com	Kasir	Aktif	2025-12-17 19:36:59
2	13	Rina	rina123	rina@gmail.com	Kasir	Aktif	2025-12-29 15:02:35
3	15	Aurel	manajer	manajer@gmail.com	Manajer	Aktif	2025-12-29 15:20:15

Berdasarkan hasil pengujian, REST API berhasil menghapus data pengguna dari database dan mengembalikan response berupa pesan keberhasilan. Data yang telah dihapus tidak lagi ditemukan pada tabel `pengguna`, sehingga fungsi DELETE dapat dinyatakan berjalan sesuai dengan perancangan.

5.6. Hasil Pengujian Sistem

Secara keseluruhan, hasil pengujian menunjukkan bahwa seluruh metode REST API yang diimplementasikan dapat berjalan dengan baik. REST API mampu mengelola data pengguna melalui proses pengambilan, penambahan, pembaruan, dan penghapusan data tanpa kendala berarti. Penggunaan Postman sebagai alat pengujian mempermudah proses verifikasi endpoint serta memastikan response yang dihasilkan sesuai dengan yang diharapkan.

BAB VI
KESIMPULAN

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa REST API berhasil diimplementasikan sebagai media komunikasi antara client dan server dalam pengelolaan data pengguna. REST API mampu menangani pertukaran data secara terstruktur menggunakan format JSON serta memanfaatkan metode HTTP sesuai fungsinya.

Metode GET digunakan untuk mengambil data pengguna dari database, sedangkan metode POST digunakan untuk menambahkan data baru. Metode PUT berperan dalam memperbarui data pengguna yang sudah ada, dan metode DELETE digunakan untuk menghapus data pengguna berdasarkan ID tertentu. Seluruh metode tersebut telah diuji menggunakan Postman dan menunjukkan hasil yang sesuai dengan perancangan sistem.

Selain itu, integrasi REST API dengan database MySQL yang dikelola melalui phpMyAdmin berjalan dengan baik. Proses validasi data yang diterapkan pada setiap metode membantu mencegah terjadinya kesalahan seperti data kosong atau duplikasi data. Dengan demikian, praktikum ini memberikan pemahaman yang jelas mengenai konsep dan penerapan REST API berbasis CRUD dalam pengembangan aplikasi web.