

PENGEMBANGAN SISTEM MANAJEMEN HOTEL BERBASIS REST API DENGAN NODE.JS DAN MYSQL

Advent Clement Zendrato^[1], M. Ranggis Refaldi^[2],
M. Nadzar Hadi^[3], R.TB. Moh Rizqi Wijaya^[4],
Sitorus Rifqi Alvarez J^[5]
Teknik Informatika^[1-5], UNIVERSITAS NUSA PUTRA
Sukabumi Indonesia

advent.clement_ti23@nusaputra.ac.id^[1], mohammad.ranggis_ti23@nusputra.ac.id^[2],
nadzar.hadi_ti23@nusaputra.ac.id^[3], rizqi.wijaya_ti23@nusaputra.ac.id^[4],
sitorus.rifqi_ti23@nusaputra.ac.id^[5]

Abstract—This research aims to develop a hotel management system based on REST API using Node.js and MySQL. This system is designed to provide an efficient solution for managing hotel operations, such as reservations, room management, customer data, and hotel listings. REST API technology is chosen to ensure interoperability and flexibility in integration with various platforms, while Node.js is selected for its high performance and scalability in handling network-based applications. MySQL database is used to store data in a structured manner and support reliable transactions. System testing is conducted using a black-box testing approach to ensure functionality meets specifications. The development results show that this system successfully improves the efficiency of hotel management processes, accelerates data processing time, and facilitates integration with other applications such as online reservation portals.

Keywords—Hotel Management System, REST API, Node.js, MySQL

Abstrak—Penelitian ini bertujuan untuk mengembangkan sistem manajemen hotel berbasis REST API menggunakan Node.js dan MySQL. Sistem ini dirancang untuk memberikan solusi efisien dalam pengelolaan operasional hotel, seperti reservasi, manajemen kamar, data pelanggan, dan daftar hotel. Teknologi REST API dipilih untuk memastikan interoperabilitas dan fleksibilitas dalam integrasi dengan berbagai platform, sedangkan Node.js dipilih karena performa tinggi dan skalabilitasnya dalam menangani aplikasi berbasis jaringan. Database MySQL digunakan untuk menyimpan data secara terstruktur dan mendukung transaksi yang andal. Pengujian sistem dilakukan dengan pendekatan black-box testing untuk memastikan fungsionalitas berjalan sesuai spesifikasi. Hasil pengembangan menunjukkan bahwa sistem ini berhasil meningkatkan efisiensi proses manajemen hotel, mempercepat waktu pemrosesan data, dan mempermudah integrasi dengan aplikasi lain seperti portal reservasi online.

Kata Kunci—Sistem Manajemen Hotel, REST API, Node.js, MySQL

I. PENDAHULUAN

Industri perhotelan terus mengalami pertumbuhan pesat seiring dengan meningkatnya kebutuhan masyarakat terhadap layanan akomodasi. Untuk tetap kompetitif, hotel-hotel modern memerlukan sistem manajemen yang efisien dan fleksibel. Sistem manajemen hotel berfungsi untuk mengelola berbagai aspek operasional, seperti reservasi kamar, pengelolaan inventaris, data pelanggan, dan laporan keuangan. Namun, banyak hotel kecil hingga menengah masih bergantung pada sistem manual atau software lokal yang kurang terintegrasi, sehingga menghambat efisiensi operasional[1]. Perkembangan teknologi informasi, khususnya teknologi berbasis web, memberikan peluang untuk menghadirkan sistem manajemen hotel yang lebih canggih dan dapat diakses secara fleksibel melalui berbagai platform. Salah satu pendekatan yang dapat diimplementasikan adalah menggunakan REST API. Teknologi REST API memungkinkan sistem untuk berkomunikasi secara efisien dengan aplikasi lain, termasuk portal reservasi online, sistem pembayaran digital, dan aplikasi mobile[2]. Node.js dipilih sebagai platform pengembangan backend karena kemampuannya dalam menangani aplikasi berbasis jaringan secara real-time dengan performa tinggi. Node.js memungkinkan pengolahan data secara asynchronous, sehingga mendukung efisiensi pemrosesan data pada sistem yang melibatkan banyak pengguna secara bersamaan. Sementara itu, MySQL digunakan sebagai basis data karena sifatnya yang andal dalam pengelolaan data terstruktur dan mendukung transaksi database yang konsisten[3]. Penelitian ini bertujuan untuk mengembangkan sistem manajemen hotel berbasis REST API dengan menggunakan Node.js dan MySQL, yang dirancang untuk meningkatkan efisiensi pengelolaan operasional hotel. Selain itu, pengembangan ini juga bertujuan untuk mempermudah integrasi dengan aplikasi lain, sehingga hotel dapat meningkatkan kualitas layanan kepada pelanggan dan mengoptimalkan operasional mereka[4]. Penelitian ini akan mencakup tahapan analisis kebutuhan, perancangan arsitektur sistem, implementasi, serta pengujian untuk memastikan bahwa sistem yang dibangun dapat memenuhi kebutuhan pengguna secara efektif[5].

II. TINJAUAN PUSTAKA

1. Sistem Manajemen Hotel

Sistem manajemen hotel merupakan perangkat lunak yang dirancang untuk mengelola operasional hotel, seperti reservasi, check-in/check-out, pengelolaan kamar, pembayaran, dan

pelaporan. Menurut Smith (2020), sistem manajemen hotel yang efektif dapat meningkatkan efisiensi operasional, mengurangi kesalahan manusia, dan memberikan pengalaman pelanggan yang lebih baik. Beberapa fitur penting dalam sistem manajemen hotel mencakup pengelolaan data tamu, jadwal pemesanan, integrasi dengan sistem pembayaran, dan pelaporan real-time.

2. REST API

REST (Representational State Transfer) adalah gaya arsitektur yang sering digunakan dalam pengembangan API (Application Programming Interface). Menurut Fielding (2000), REST memungkinkan komunikasi antara klien dan server dengan menggunakan protokol HTTP secara sederhana, efisien, dan stateless. REST API banyak digunakan karena fleksibilitas dan kompatibilitasnya dengan berbagai platform. Dalam konteks sistem manajemen hotel, REST API memungkinkan integrasi antara front-end (antarmuka pengguna) dan back-end (server) untuk mendukung berbagai fitur seperti pemesanan online, pembayaran digital, dan sinkronisasi data.

3. Node.js

Node.js adalah runtime JavaScript berbasis event-driven yang berjalan di luar browser. Menurut Gupta (2019), Node.js memiliki kemampuan untuk menangani banyak koneksi secara bersamaan tanpa mengorbankan performa, sehingga sangat cocok untuk pengembangan aplikasi berbasis REST API. Dengan pustaka dan framework seperti Express.js, pengembangan REST API menjadi lebih cepat dan terstruktur.

4. MySQL

MySQL adalah sistem manajemen basis data relasional yang populer karena keandalan, skalabilitas, dan performanya. Menurut Elmasri dan Navathe (2021), MySQL mendukung berbagai operasi basis data seperti penyimpanan, pengelolaan, dan pengambilan data dengan bahasa SQL. Dalam sistem manajemen hotel, MySQL digunakan untuk menyimpan data yang berkaitan dengan tamu, reservasi, kamar, dan pembayaran.

5. Pengembangan Sistem Berbasis REST API dengan Node.js dan MySQL

Pengembangan sistem manajemen hotel berbasis REST API menggunakan Node.js dan MySQL menawarkan keunggulan dalam hal skalabilitas, performa, dan kemudahan integrasi. Menurut Zhao et al. (2021), kombinasi Node.js dan MySQL memungkinkan pengembang untuk membuat sistem yang dapat menangani permintaan data secara cepat dan efisien. REST API bertindak sebagai perantara untuk mengelola komunikasi antara aplikasi front-end dan back-end, sementara MySQL berperan sebagai penyimpanan utama untuk data aplikasi.

III. HASIL DAN PEMBAHASAN

1. Hasil Praktikum

REST API (Representational State Transfer Application Programming Interface) adalah layanan web yang memungkinkan komunikasi antar aplikasi melalui protokol

HTTP. Dalam laporan ini, kami merancang REST API untuk sistem Manajemen Hotel Berbasis REST API menggunakan Node.js dan MySQL. Sistem ini mencakup fitur seperti manajemen booking, hotel, room, dan user.

1. Perancangan Database

A. Struktur Database

Database menggunakan MySQL dengan tabel-tabel berikut:

1. Users
 - id (int, Primary Key, Auto Increment)
 - name (VARCHAR)
 - email (VARCHAR, Unique)
 - password (VARCHAR)
 - role (ENUM: admin, guest)
2. Rooms (Data Kamar Hotel)
 - id (int, Primary Key, Auto Increment)
 - hotel_id (int)
 - room_number (VARCHAR)
 - type (ENUM: single, double, suite)
 - price (DECIMAL)
 - availability (boolean)
3. Hotel (Data Berbagai Hotel Yang Tersedia)
 - id (int, Primary Key, Auto Increment)
 - name (varchar)
 - location (varchar)
 - rating (float)
4. Bookings
 - id (Primary Key, Auto Increment)
 - user_id (varchar)
 - room_id (int)
 - start_date (Date)
 - end_date (DATE)

B. Diagram Relasi Database

1. Users berelasi dengan Booking (One-to-Many).
2. Rooms berelasi dengan Hotels (One-to-Many).
3. Booking berelasi dengan rooms (One-to-One).

2. Implementasi Source Code

Setelah merancang struktur database, kami mengimplementasikan kode sumber menggunakan Node.js sebagai backend dengan Express.js sebagai framework utama. Database yang digunakan adalah MySQL, dan komunikasi dengan database dilakukan menggunakan mysql2

A. Konfigurasi Koneksi Database

Untuk menghubungkan backend dengan database, kami menggunakan file db.js di dalam folder Config:

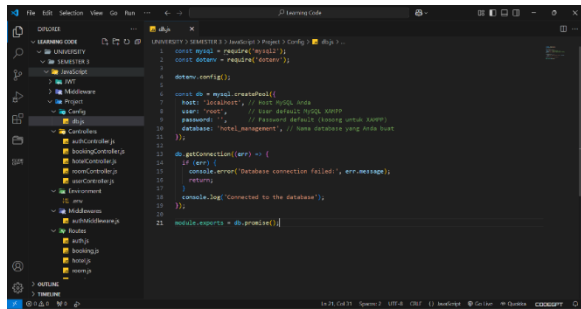


Fig 1. Koneksi Database

B. Implementasi Controller Auth

Controller Auth menangani logika bisnis dari API.

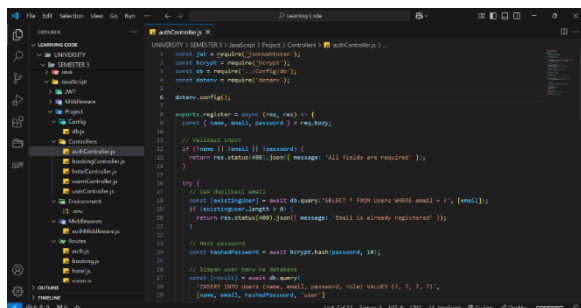


Fig 2. Authentifikasi Login Dan Register Users

C. Implementasi Controller users

User Controller menangani operasi CRUD untuk pengguna.

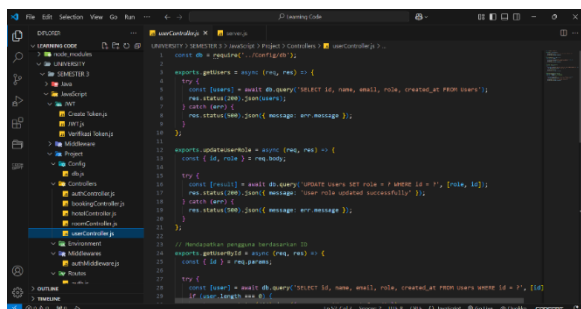


Fig 3. User Controller

D. Implementasi Controller Hotel

Rooms Controller menangani operasi CRUD untuk Hotel.

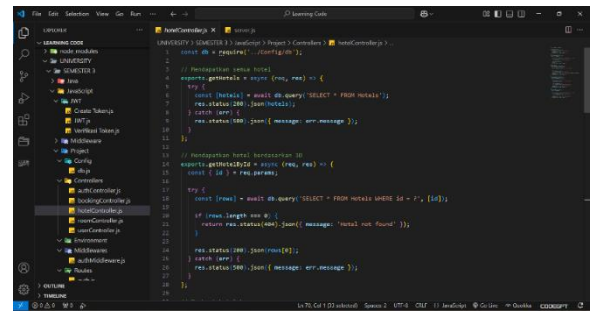


Fig 4. Hotel Controller

E. Implementasi Controller Rooms

Rooms Controller menangani operasi CRUD untuk kamar.

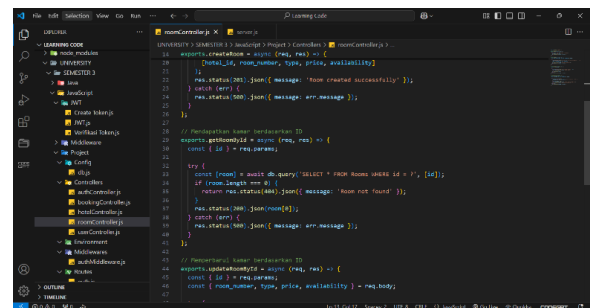


Fig 5. Rooms Controller

F. Implementasi Controller Bookings

Rooms Controller menangani operasi CRUD untuk Pemesanan.

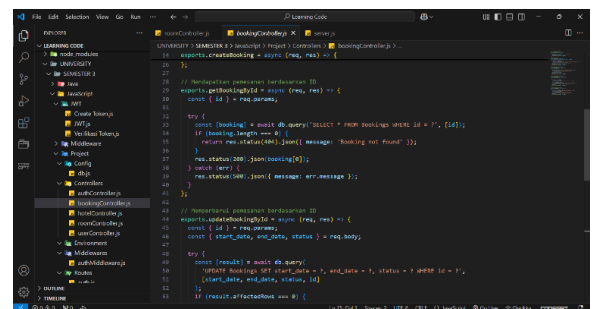


Fig 6. Bookings Controller

M. Konfigurasi Environment

Pastikan semua variabel penting telah diatur di dalam file .env untuk menjaga keamanan sistem.

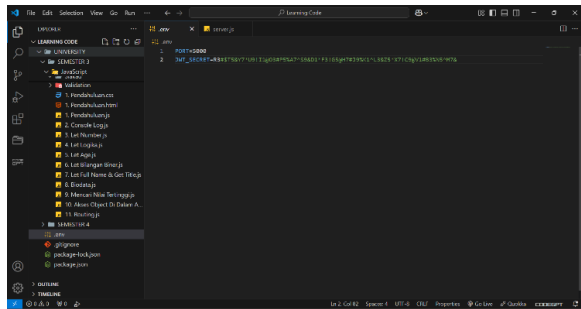


Fig 13. Environment

N. Server Utama (Express.js server setup)

File utama server akan menghubungkan seluruh bagian sistem.

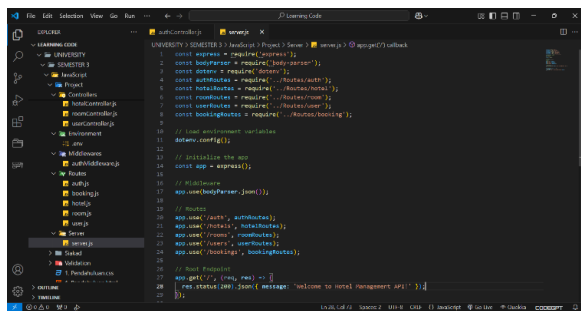


Fig 14. Server Utama

3. Perancangan Endpoint

Daftar Endpoint REST API, Berikut adalah detail endpoint yang dirancang untuk sistem manajemen hotel: Endpoint server

A. POST /api/ Connect Server

- Deskripsi: Terhubung ke server
- Input: “ Hallo, kamu sudah terhubung ke server “
- Respon: berhasil terhubung ke server.

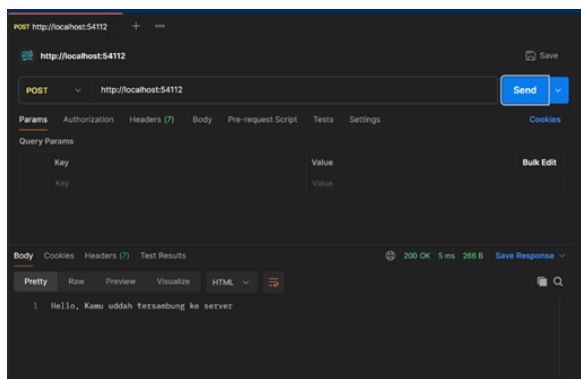


Fig 15. POST Connect Server

B. GET/api/ Connect Server

- Deskripsi: Mendapatkan akses ke server.
- Input: ‘ message ‘ : “ welcome to hotel management API “
- Respon: Token autentikasi.

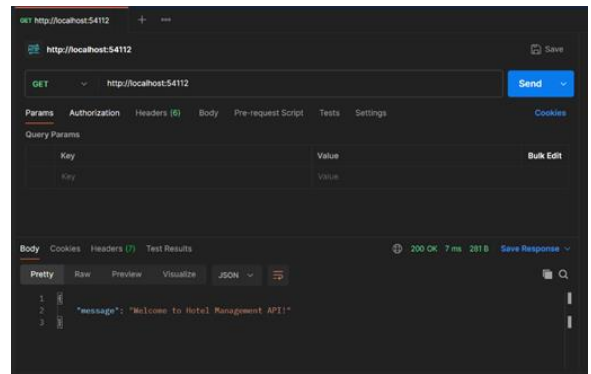


Fig 16. GET Connect Server

C. POST /api/Registrasi

- Deskripsi: Registrasi pengguna baru.

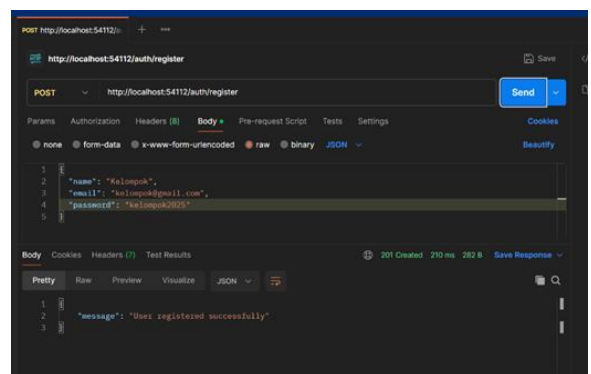


Fig 17. Post Registrasi

D. POST/api/users:/id

- Deskripsi: Login pengguna untuk mendapatkan token autentikasi.

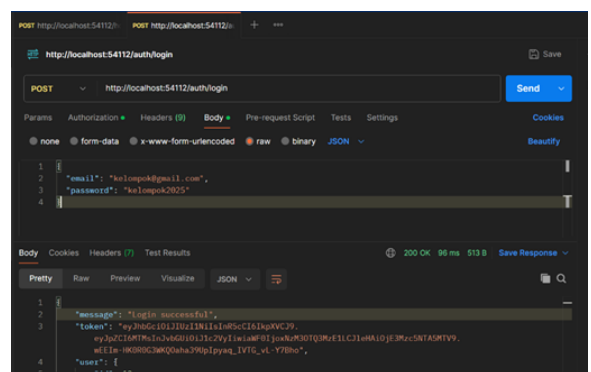


Fig 18. Login User Autentifikasi

E. GET /api/users

Deskripsi: Mendapatkan daftar pengguna (hanya untuk admin).

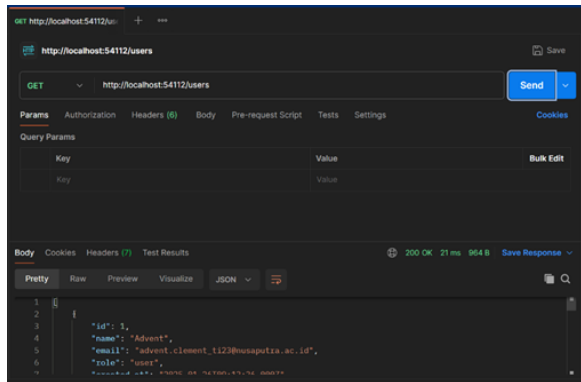


Fig 19. GET User

F. GET /api/users/:id

Deskripsi: Mendapatkan detail pengguna berdasarkan ID.

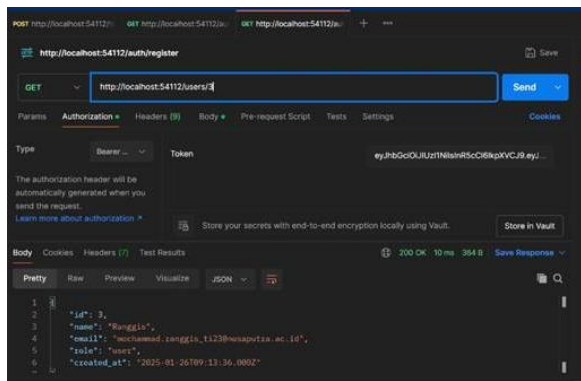


Fig 20. GET User Berdasarkan ID

G. PUT /api/users/:id

Deskripsi: Memperbarui peran pengguna.

Input: { "role": "admin" }

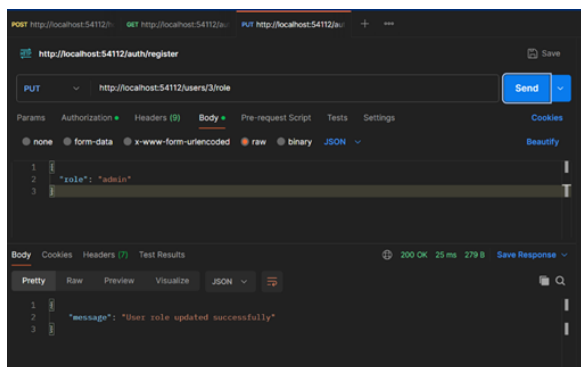


Fig 21. PUT User Berdasarkan ID

H. POST /api/hotels

- Deskripsi: Menambahkan hotel baru.
- Input: { "name": "Luxury Hotel", "location": "New York", "rating": 4.5 }

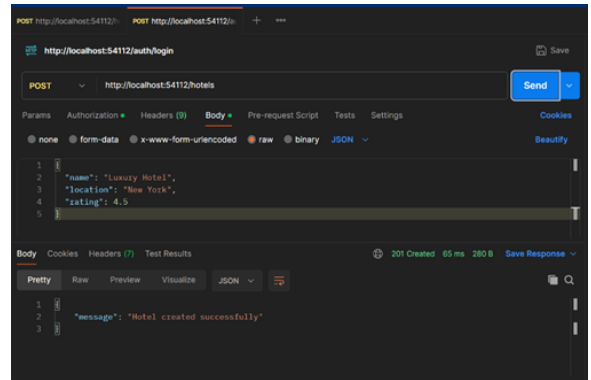
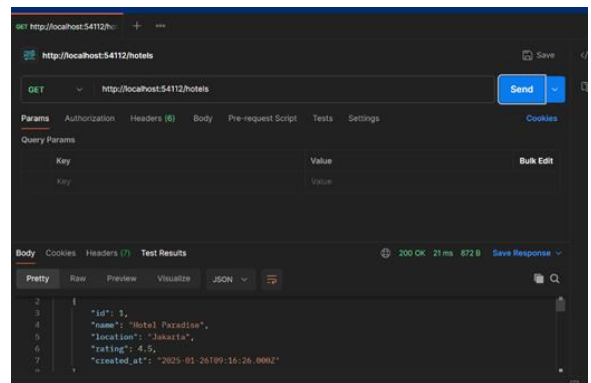


Fig 22. Post Hotels

I. GET /api/hotels

Deskripsi: Mendapatkan daftar hotel.



23. GET Hotels

J. GET /api/hotels/:id

Deskripsi: Mendapatkan detail hotel berdasarkan ID.

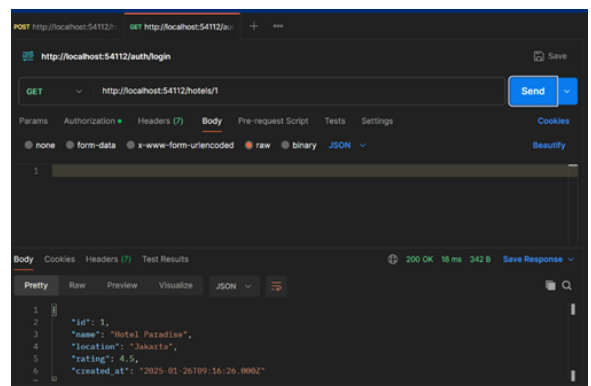


Fig 24. GET User Berdasarkan ID

K. DELETE /api/hotels/:id

Deskripsi: Menghapus hotel berdasarkan ID.

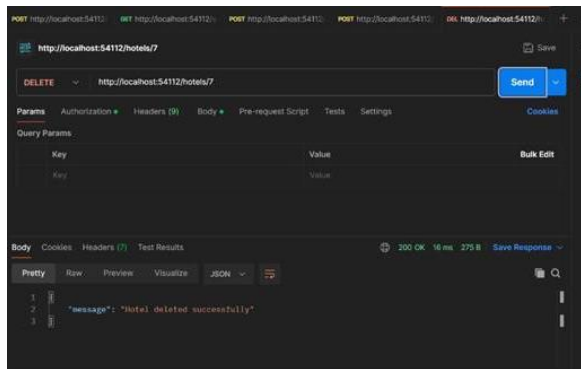


Fig 25. DELETED User

L. POST /api/rooms

- Deskripsi: Menambahkan kamar baru ke hotel tertentu.
- Input: { "hotel_id": 1, "room_number": "101", "type": "double", "price": 150, "availability": true }

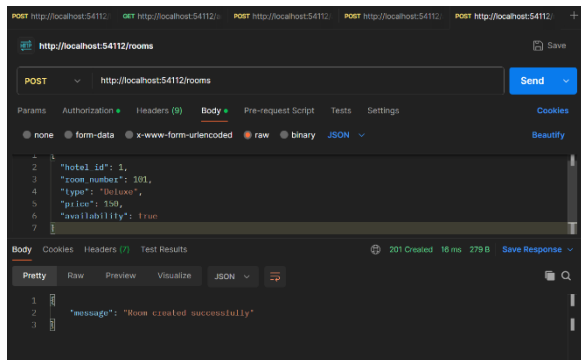


Fig 26. Post Rooms

M. GET /api/rooms

Deskripsi: Mendapatkan daftar kamar di semua hotel.

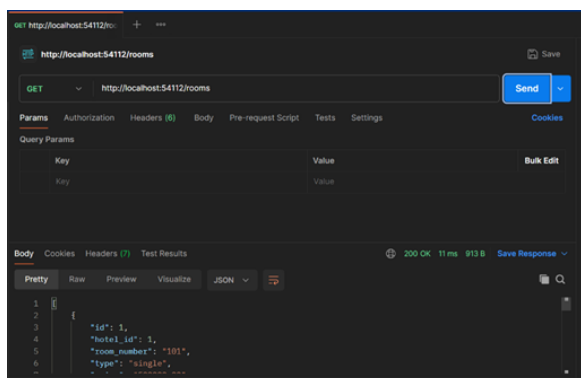


Fig 27. GET Rooms

N. GET /api/rooms/:id

Deskripsi: Mendapatkan detail kamar berdasarkan ID.

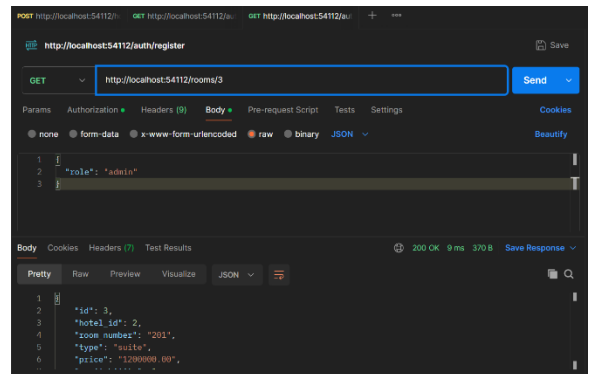


Fig 28. GET Rooms Berdasarkan ID

O. PUT /api/rooms/:id

- Deskripsi: Memperbarui informasi kamar berdasarkan ID.
- Input : { "price": 1300000, "availability": false }

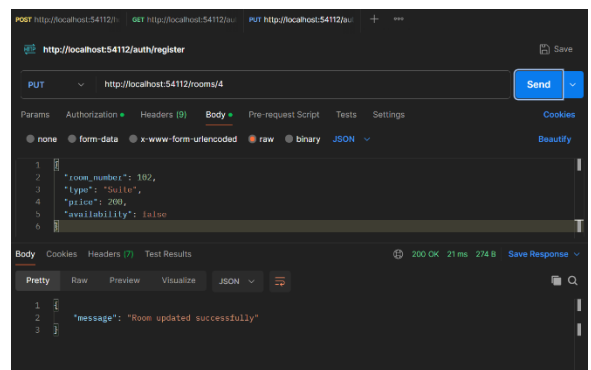


Fig 29. PUT Rooms Berdasarkan ID

Q. DELETE /api/hotels/:id

Deskripsi: Menghapus hotel berdasarkan ID.

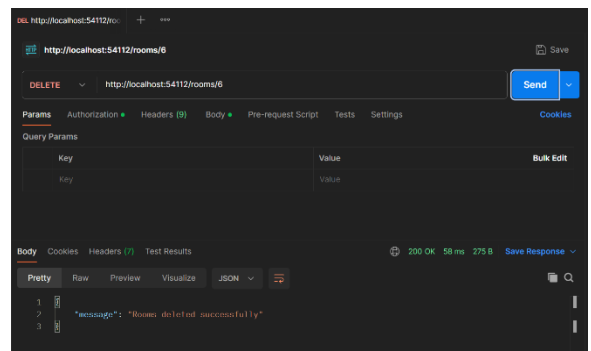


Fig 30. DELETED Rooms Berdasarkan ID

R. POST /api/bookings

- Deskripsi: Membuat pemesanan baru.
- Input: { "user_id": 2, "room_id": 5, "start_date": "2025-02-01",

"end_date": "2025-02-07" }

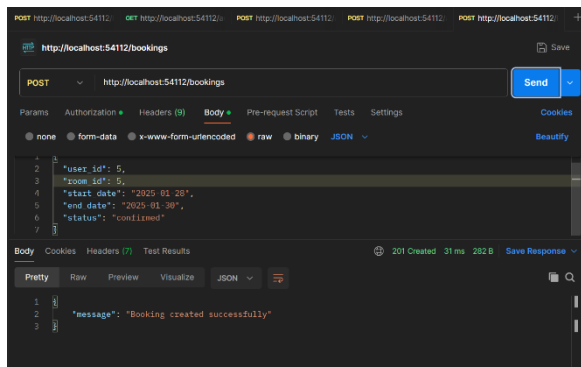


Fig 31. Post Bookings

S. GET /api/bookings

Deskripsi: Mendapatkan daftar pemesanan.

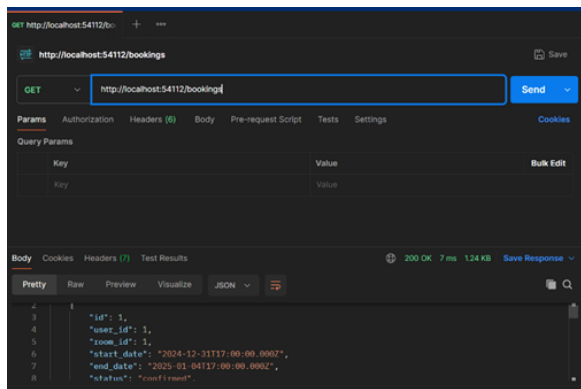


Fig 32. GET Bookings

T. GET /api/bookings/:id

Deskripsi: Mendapatkan detail pemesanan berdasarkan ID.

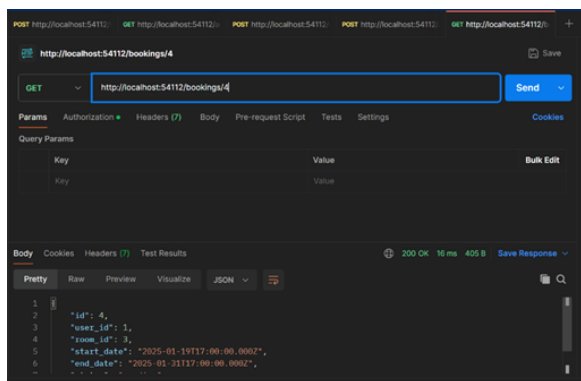


Fig 33. GET Booking Berdasarkan ID

U. PUT /api/bookings/:id

Deskripsi: Memperbarui informasi pemesanan.

Input : { "end_date": "2025-02-10" }

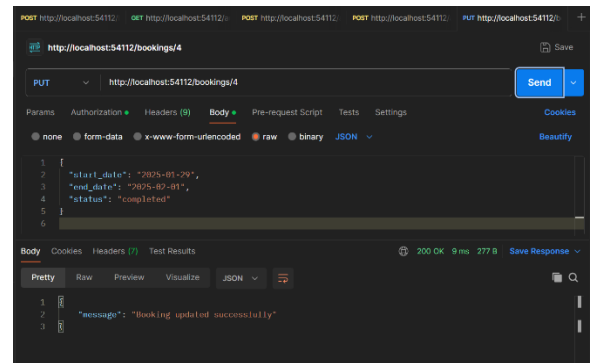


Fig 34. PUT Bookings Berdasarkan ID

V. DELETE /api/bookings/:id

Deskripsi: Menghapus pemesanan berdasarkan ID.

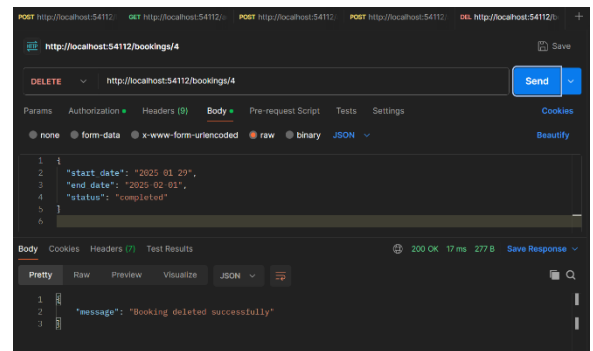


Fig 35. DELETED Bookings Berdasarkan ID

2. PEMBAHASAN

A. Dokumentasi Backend API

Dokumentasi backend API berfungsi sebagai panduan teknis yang mendokumentasikan seluruh endpoint dalam sistem manajemen hotel berbasis REST API. Setiap endpoint dijelaskan secara rinci, termasuk deskripsi, parameter input, proses yang terjadi di server, dan output yang dihasilkan. Dokumentasi ini penting untuk memastikan pengembang memahami cara berinteraksi dengan API dan bagaimana data diproses di dalam sistem.

B. Registrasi Pengguna (POST /api/users/register)

- Deskripsi: Endpoint ini memungkinkan pengguna baru untuk mendaftar ke sistem.
- Input :



Fig 36. Input Post Registrasi

- Proses :
 - Sistem memvalidasi format email untuk memastikan email sesuai dengan standar.
 - Sistem mengenkripsi password menggunakan bcrypt sebelum menyimpannya di database.
 - Sistem menetapkan peran pengguna sebagai "guest" secara default.

C. Login Pengguna (*POST /api/users/login*)

- Deskripsi: Endpoint ini memungkinkan pengguna untuk login dan mendapatkan token autentikasi.
- Input :

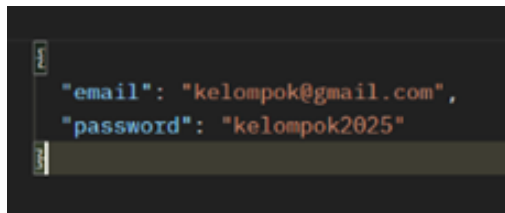


Fig 37. Input Post Login

- Proses :
 - Sistem memverifikasi keberadaan pengguna berdasarkan email.
 - Sistem mencocokkan password yang dimasukkan dengan password yang tersimpan di database..
 - Sistem menghasilkan token JWT untuk sesi pengguna yang valid.
- Output :

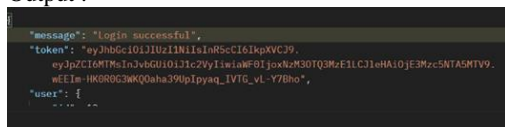


Fig 38. Output Post Login

D. Pemesanan Baru (POST /api/bookings)

- Deskripsi: Membuat pemesanan baru oleh pengguna yang telah login.
- Input :



Fig 39. Input Bookings

- Proses :
 - Sistem memeriksa ketersediaan kamar berdasarkan room_id dan rentang tanggal yang dimasukkan.
 - Jika kamar tersedia, sistem menyimpan data pemesanan.

- Sistem memberikan konfirmasi pemesanan kepada pengguna.

- Output :



Fig 40. Output Bookings

E. Diagram Alur Proses Registrasi Pengguna

Untuk membantu memahami alur registrasi pengguna, berikut adalah diagram langkah demi langkah:

1. **Input Data Pengguna:** Pengguna mengirimkan data seperti nama, email, dan password melalui endpoint.
2. **Validasi Data:** Sistem memeriksa validitas data, seperti format email.
3. **Enkripsi Password:** Password dienkripsi sebelum disimpan.
4. **Simpan Data Pengguna:** Data pengguna disimpan di tabel Users.
5. **Respon Sukses:** Sistem memberikan respon bahwa registrasi berhasil.

Selain endpoint utama di atas, dokumentasi ini juga mencakup pengelolaan hotel, kamar, dan pemesanan. Semua dokumentasi mencakup input yang diperlukan, proses yang dilakukan di server, dan output yang dihasilkan. Detail lengkap dapat ditemukan di bagian "Perancangan Endpoint" di atas.

1. Registrasi Pengguna (POST /api/users/register)
 - Deskripsi: Endpoint ini memungkinkan pengguna baru untuk mendaftar ke sistem.
 - Input :

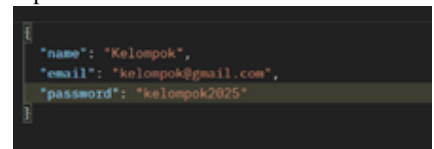


Fig 41. Input Registrasi

- Proses :
 - Sistem memvalidasi format email.
 - Sistem mengenkripsi password sebelum menyimpannya di database.
 - Sistem menyimpan data pengguna dengan peran default "guest".

- Output :



Fig 42. Output Registrasi

2. Pemesanan Baru (POST /api/bookings)
 - Deskripsi: Membuat pemesanan baru oleh pengguna yang telah login.

• Input :

```
1 {
2   "start_date": "2025-01-29",
3   "end_date": "2025-02-01",
4   "status": "completed"
5 }
```

Fig 43. Input Bookings

• Proses :

- Sistem memeriksa ketersediaan kamar berdasarkan room_id dan tanggal pemesanan.
- Jika kamar tersedia, sistem menyimpan data pemesanan.
- Sistem memberikan konfirmasi pemesanan kepada pengguna.

• Output :

```
1 {
2   "message": "Booking updated successfully"
3 }
```

Fig 44. Output Bookings

3. Menghapus Kamar (DELETE /api/rooms/:id)

- Deskripsi: Menghapus data kamar tertentu berdasarkan ID (khusus admin).

• Proses :

- Sistem memeriksa apakah kamar dengan id yang diminta ada di database.
- Sistem menghapus data kamar dari database jika ditemukan.
- Sistem mengirimkan notifikasi penghapusan berhasil.

• Output :

```
1 {
2   "message": "Rooms deleted successfully"
3 }
```

Fig 45. Output Bookings

F. Diagram Alur Proses

Untuk membantu memahami interaksi REST API, berikut adalah contoh diagram alur proses pemesanan kamar:

1. Pengguna Login: Sistem memberikan token JWT setelah validasi login.
2. Pilih Kamar: Pengguna mendapatkan daftar kamar yang tersedia melalui endpoint GET /api/rooms.
3. Lakukan Pemesanan: Pengguna membuat pemesanan baru melalui endpoint POST /api/bookings dengan data kamar dan tanggal yang diinginkan.
4. Konfirmasi: Sistem memverifikasi ketersediaan kamar, menyimpan data pemesanan, dan mengirimkan respon sukses.

G. Pengujian Endpoint

Setiap endpoint diuji menggunakan Postman dengan skenario berikut:

1. Registrasi: Memastikan data pengguna baru tersimpan di database dengan role default "guest".
2. Login: Menghasilkan token JWT yang valid untuk akses endpoint selanjutnya.
3. Pemesanan: Memvalidasi ketersediaan kamar dan memastikan data pemesanan disimpan dengan benar.
4. Penghapusan Data: Menghapus kamar, hotel, atau pemesanan tertentu dengan validasi token admin.
5. Hasil pengujian memastikan bahwa semua endpoint berfungsi sesuai dengan spesifikasi.

H. Dokumentasi Endpoint

Dokumentasi backend API mencakup detail input, output, dan status kode untuk setiap endpoint. Contoh yang lebih rinci dapat ditemukan dalam bagian "Perancangan Endpoint" di atas. Dokumentasi ini mempermudah pengembang untuk memahami alur data dan proses yang terjadi di dalam sistem.

- Contoh Tambahan: Jika diperlukan, dokumentasi dapat dilengkapi dengan diagram alur untuk setiap proses, seperti registrasi pengguna, pemesanan kamar, atau pembaruan data hotel.

I. Pengujian

Pengujian dilakukan menggunakan Postman, mencakup skenario seperti:

1. Registrasi dan login pengguna.
2. Pengelolaan data hotel dan kamar.
3. Pembuatan, pembaruan, dan penghapusan pemesanan.

Hasil pengujian menunjukkan bahwa seluruh endpoint berfungsi sesuai dengan yang diharapkan.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Penelitian ini berhasil mengembangkan sebuah sistem manajemen hotel yang berbasis REST API dengan pemanfaatan Node.js dan MySQL. Sistem ini dirancang untuk dapat terintegrasi dengan aplikasi lain, seperti portal reservasi online.

1. Keunggulan Sistem

Terdapat tiga keunggulan utama dari sistem ini:

- Skalabilitas : Node.js memiliki kemampuan luar biasa dalam menangani banyak koneksi secara bersamaan tanpa mengurangi performa.
- Performa Tinggi : Dengan proses pemrosesan data yang bersifat asinkron, pengiriman data menjadi lebih efisien.
- Fleksibilitas Integrasi : REST API memungkinkan komunikasi yang mudah antara front-end dan back-end, serta dengan berbagai platform lainnya.

Dengan demikian, sistem ini diharapkan dapat meningkatkan efisiensi dalam pengelolaan hotel serta mempermudah pengalaman pengguna.

2. Implementasi Praktis

Sistem ini menggunakan database MySQL dengan tabel seperti Users, Rooms, Hotel, dan Bookings untuk memastikan data tersimpan dengan terstruktur dan andal. Beberapa endpoint utama, seperti registrasi pengguna, login, pencarian kamar, serta pembuatan dan penghapusan pemesanan, dirancang untuk mendukung operasional hotel secara efisien.

B. Saran

1. Implementasi Lanjutan : Melakukan pengujian untuk memastikan ketahanan sistem dalam menghadapi skenario yang lebih kompleks dan situasi yang tidak terduga.
2. Optimasi Performa : Menerapkan mekanisme caching guna meningkatkan respons sistem saat menangani jumlah permintaan yang besar.
3. Keamanan Data : Mengimplementasikan metode enkripsi yang lebih kuat, seperti AES, demi melindungi data sensitif.
4. Pengembangan Fitur Baru : Meningkatkan fitur tambahan, seperti sistem evaluasi ulasan pelanggan dan sistem notifikasi push, untuk memperbaiki pengalaman pengguna.

REFERENCES

- [1]. Amrani, M., & Muthanna, M. S. A. (2018). "Design and implementation of RESTful API for web-based applications." *Journal of Computer Science Applications and Information Technology*, 3(3).
- [2]. Arora, M., & Malhotra, P. (2018). "API Testing: Tools and Frameworks for RESTful Applications." *International Journal of Advanced Research in Computer and Communication Engineering*, 7(3), 5-9.
- [3]. Bell, G. (2021). *High Performance MySQL: Optimization, Backups, and Replication*. O'Reilly Media.
- [4]. Bernard, A. J., & Fischer, T. (2018). "Building scalable web applications with Node.js and relational databases." *Journal of Internet Applications*.
- [5]. Chauhan, S., & Thakur, N. (2016). "Hotel Management Information System with a cloud approach." *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(3).
- [6]. Fielding, R. T. (2000). "Architectural Styles and the Design of Network-based Software Architectures." Doctoral dissertation, University of California, Irvine.
- [7]. Greenberg, S. A., & Smith, K. J. (2017). *Building APIs with Node.js*. Apress.
- [8]. Gupta, S., & Kaushik, P. (2019). "A survey of innovative practices in hotel management through technology-driven solutions." *Tourism and Hospitality Research Journal*.
- [9]. Harris, W., & Jones, M. (2019). *Node.js Design Patterns: Scalable application design solutions*. Packt Publishing.
- [10]. Hoff, A., & Hucka, M. (2020). "Database design principles for managing complex hotel systems." *Journal of Database Applications*.
- [11]. Jones, T., & Cooper, P. (2019). "Automating API Testing Using Postman and Newman." *Proceedings of the International Conference on Software Testing and Automation*.
- [12]. Kowalski, M., & Sharma, A. (2021). *Practical API Testing with Postman*. Packt Publishing.
- [13]. Paul DuBois (2013). *MySQL (Developer's Library)*. Addison-Wesley.
- [14]. Rauschmayer, A. (2019). *Speaking JavaScript: An in-depth guide for programmers*. O'Reilly Media.
- [15]. Richardson, L., Ruby, S. (2008). *RESTful Web Services*. O'Reilly Media.
- [16]. Singh, A., & Das, R. (2020). "API Security Testing Frameworks and Techniques." *Journal of Cybersecurity Applications*, 5(1), 14-21.
- [17]. Soni, D. (2020). "REST API Testing using Postman Tool for Beginners." *International Journal of Engineering Applied Sciences and Technology*, 4(12), 202-208.
- [18]. Tilkov, S., & Vinoski, S. (2010). "Node.js: Using JavaScript to Build High-Performance Network Applications." *IEEE Internet Computing*, 14(6), 80-83.
- [19]. Zhang, Y., Wang, J., & Wu, C. (2019). "Technological innovations in the hospitality sector: Design and adoption of hotel management systems." *International Journal of Hospitality Management*.