BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

1.1 IMPLEMENTASI SISTEM

Pada tahap ini, implementasi difokuskan pada pengembangan backend berdasarkan rancangan yang telah dibuat sebelumnya. Proses implementasi ini bertujuan untuk menerjemahkan rancangan tersebut menjadi sebuah program aplikasi berbasis website. Backend akan dirancang untuk mendukung berbagai role pengguna dengan fitur dan fungsi yang berbeda sesuai dengan peran masing-masing, memastikan setiap role memiliki akses dan kemampuan yang sesuai dalam sistem.

1.1.1 Struktur Direktori

Struktur direktori backend berfungsi sebagai tempat untuk mengorganisasi kode dan file secara sistematis, sehingga memudahkan pengembangan, pemeliharaan, dan skala aplikasi. Struktur direktori backend cihuykids dapat kita lihat pada gambar berikut:

```
backend/
|— config/
|   ├— database.js  # Konfigurasi database menggunakan Sequelize
|— models/          # Model database
|— routes/          # Definisi endpoint API
|— controllers/     # Logika bisnis untuk masing-masing fitur
|— middlewares/     # Middleware autentikasi dan keamanan
|— uploads/        # Penyimpanan file yang diunggah
|— app.js          # Entry point aplikasi
|— .env            # Konfigurasi variabel lingkungan
```

Gambar 5. 1 Struktur Direktori

Dari gambar 5.1 diatas dapat digambarkan Struktur direktori backend dirancang untuk mengorganisasi berbagai komponen aplikasi secara rapi dan terstruktur. Direktori config/ berisi file konfigurasi, seperti database.js, yang mengatur koneksi database menggunakan Sequelize. Direktori models/ digunakan untuk menyimpan model database yang merepresentasikan tabel dalam sistem. Direktori routes/ berisi definisi endpoint API, sedangkan controllers/ mengelola logika bisnis untuk masing-masing fitur. Middleware, seperti autentikasi dan keamanan, ditempatkan dalam direktori middlewares/. File-file yang diunggah pengguna disimpan dalam direktori uploads/. Aplikasi dimulai dari file app.js, yang berfungsi sebagai entry point utama. Selain itu, file .env digunakan untuk menyimpan konfigurasi variabel lingkungan, seperti kredensial database atau kunci *API*, guna mendukung keamanan dan fleksibilitas aplikasi.

1.1.2 Implementasi *API*

Dari yang sudah dibahas pada bab 4 tentang tabel endpoint, cihuykids menyediakan berbagai endpoint *API*, diantaranya dapat kita lihat pada tabel dibawah ini:

Tabel 5. 1 Implementasi *API*

API	DESKRIPSI
/api/register	Registrasi pengguna baru.
/api/login	Login pengguna.
/api/profile	Mendapatkan data profil pengguna (dilindungi JWT).

/api/change-password	Mengubah kata sandi (dilindungi JWT).
/api/videos	Mengelola daftar video.
/api/music	Mengelola daftar musik.
/api/reading	Mengelola daftar bacaan.
/api/payments/create	Membuat transaksi pembayaran.
/api/payments/history	Melihat riwayat pembayaran.
/api/payments/approve/:id	Admin menyetujui pembayaran.

Pada tabel 5.1 diatas dapat kita ketahui bahwa Backend CihuyKids menyediakan berbagai endpoint API untuk mendukung fungsionalitas aplikasi. Pada fitur autentikasi dan pengelolaan pengguna, tersedia endpoint seperti /api/register untuk registrasi pengguna baru, /api/login untuk login, /api/profile untuk mendapatkan data profil pengguna, dan /api/change-password untuk mengubah kata sandi, di mana beberapa endpoint dilindungi menggunakan JWT untuk memastikan keamanan data. Dalam pengelolaan konten hiburan, terdapat endpoint seperti /api/videos, /api/music, dan /api/reading yang memungkinkan pengelolaan daftar video, musik, dan bacaan. Untuk sistem pembayaran, disediakan endpoint seperti /api/payments/create untuk membuat transaksi, /api/payments/history untuk melihat riwayat pembayaran, dan /api/payments/approve/:id bagi admin untuk menyetujui pembayaran. Selain itu, fitur favorit dan riwayat juga dilindungi menggunakan middleware JWT untuk

menjaga akses terhadap data sensitif. Keseluruhan sistem API ini dirancang untuk mendukung kebutuhan aplikasi secara aman dan efisien.

1.2 PENGUJIAN SISTEM

Pengujian sistem dilakukan untuk memastikan bahwa backend bekerja sesuai dengan spesifikasi yang telah dirancang. Pengujian ini mencakup pengujian fungsional, pengujian API, serta pengujian keamanan.

1.2.1 Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan bahwa fitur-fitur backend berjalan dengan baik sesuai dengan kebutuhan pengguna. Berikut adalah hasil pengujian beberapa fitur utama:

Tabel 5. 2 Tabel Pengujian

Fitur	Skenario	Hasil yang Diharapkan	Status
Registrasi Pengguna	Pengguna mengisi data registrasi dengan benar	Akun berhasil dibuat	Berhasil
Login	Pengguna memasukkan email dan password yang benar	Pengguna berhasil masuk dengan JWT Token	Berhasil
Menonton dan memutar video, musik, bacaan	Pengguna mengakses daftar video, musik, bacaan	Video, musik, dan bacaan ditampilkan dengan benar	Berhasil
Pembayaran Premium	Pengguna melakukan pembayaran premium	Status pembayaran tercatat dan pengguna bisa premium	Berhasil

CRUD Admin	Admin mengelola konten video, musik, bacaan	Admin dapat mengelola konten hiburan	Berhasil

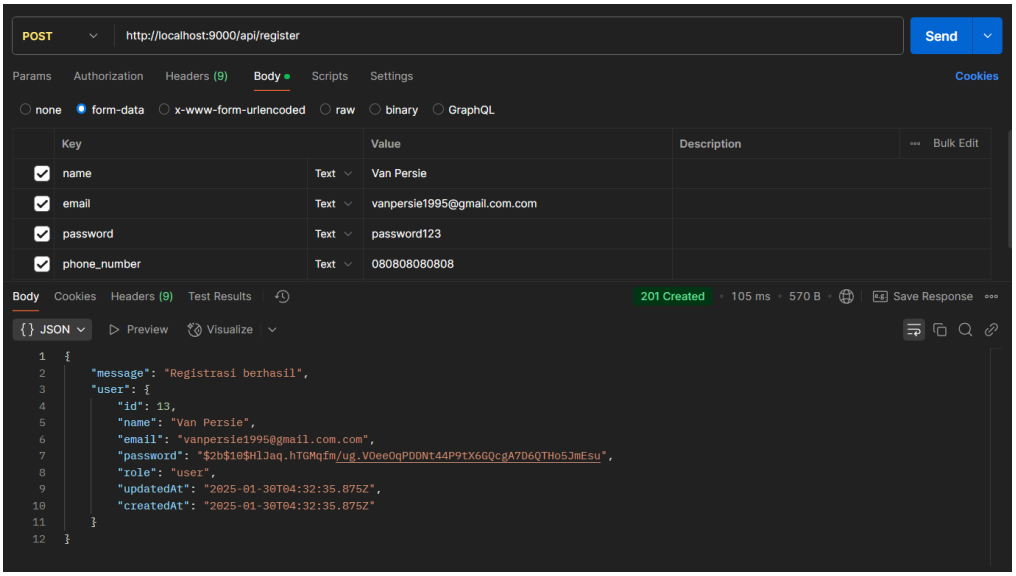
Dari tabel 5.2 diatas, dapat kita ketahui status dari tabel pengujian bahwa semuanya sukses dan berhasil, dimulai dari create akun sampai pengelolaan konten oleh admin semua dapat dilakukan.

1.2.2 Pengujian API

Pada tahap ini penulis melakukan pengujian API menggunakan Postman dengan berbagai skenario pengujian terhadap setiap endpoint. Adapun hasil dari masing-masing pengujian API sebagai berikut:

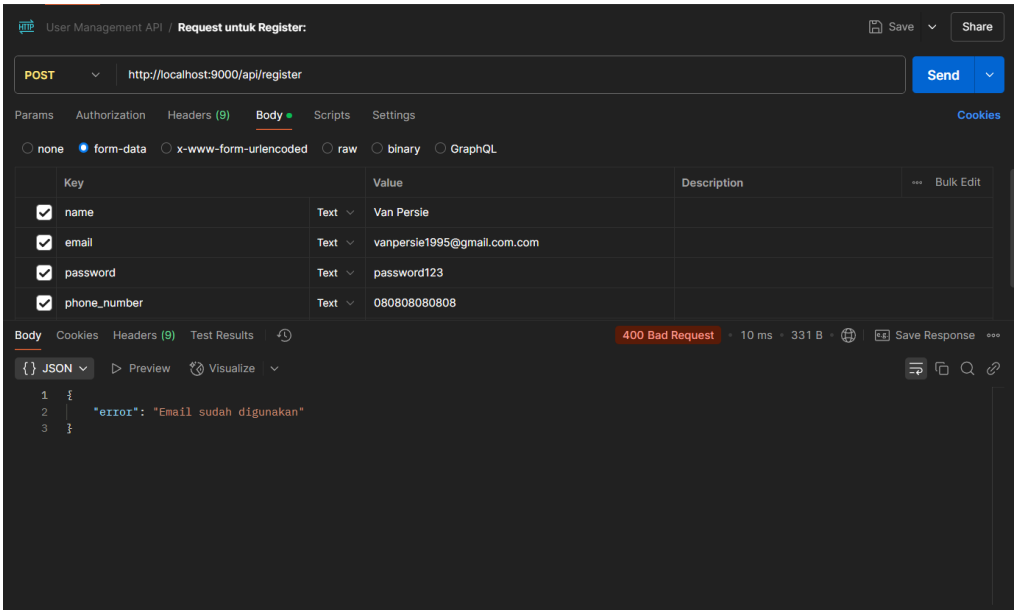
1. Pengujian API Register

Pengujian API Register dilakukan untuk memastikan bahwa sistem dapat menerima data pengguna baru dan menyimpannya ke dalam database. Data yang dimasukkan meliputi nama, email, kata sandi, dan nomor telepon (opsional). Permintaan dikirimkan ke endpoint POST dengan format form-data atau JSON sesuai dengan kebutuhan API. Pada pengujian ini, jika data valid, server akan memberikan respons dengan status kode 201 Created, pesan sukses, serta data pengguna yang baru saja dibuat (kecuali kata sandi yang telah di-hash untuk keamanan). Namun, jika data tidak valid, seperti email yang sudah terdaftar, server akan mengembalikan pesan error dengan status kode 400 Bad Request. Hasil pengujian dapat dilihat pada gambar 5.2:



Gambar 5. 2 Pengujian Register Berhasil

Dari gambar 5.2 diatas dapat kita lihat bahwa proses pendaftaran berhasil dan keterangan json dibawah tertulis lengkap, hal ini menandakan bahwa proses register berhasil sepenuhnya, jika email yang digunakan telah digunakan sebelumnya, maka proses pendaftaran gagal dan dapat kita lihat pada contoh gambar 5.3 dibawah ini.

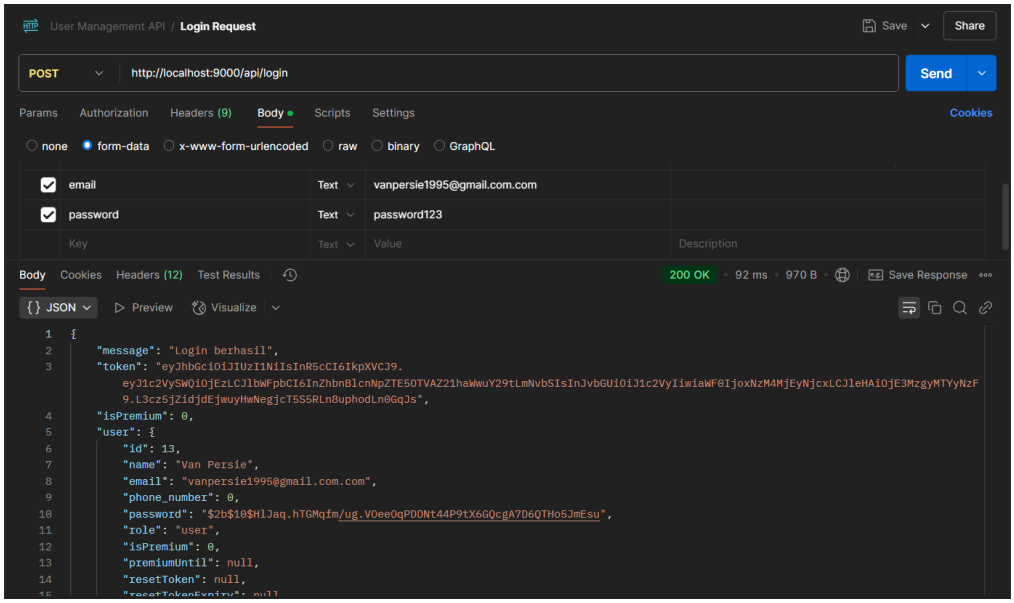


Gambar 5. 3 Pengujian Registrasi Gagal

Pada gambar 5.3 diatas, penulis mencoba memasukan data yang sama seperti pada gambar 5.2 sebelumnya, dan hasilnya gagal dan muncul keterangan bahwa email telah digunakan.

2. Pengujian API Login

Setelah registrasi berhasil, pengujian API Login dilakukan untuk memastikan bahwa pengguna dapat mengakses sistem dengan menggunakan email dan kata sandi yang telah terdaftar. Data login dikirimkan ke endpoint POST dalam format JSON atau form-data. Jika email dan kata sandi sesuai, server akan memberikan respons berupa status kode 200 OK, token autentikasi (JWT atau token lainnya), serta informasi pengguna. Token ini akan digunakan untuk keperluan autentikasi pada API lainnya. Jika email atau kata sandi salah, server akan mengembalikan pesan error dengan status kode 401 Unauthorized. Gambar hasil pengujian dapat dilihat pada gambar 5.4 dibawah ini:



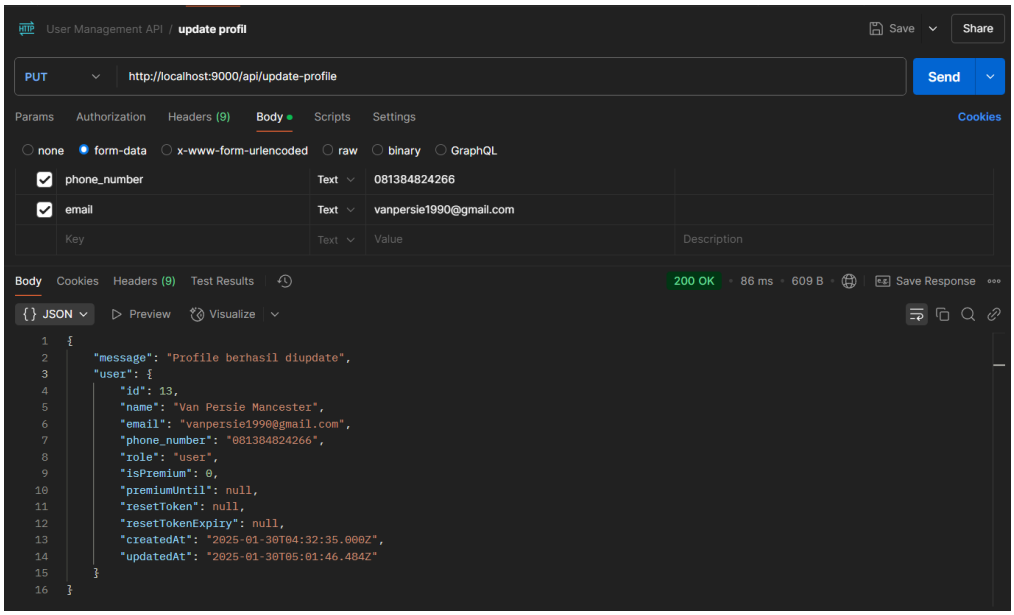
Gambar 5. 4 Pengujian API Login

Dari gambar 5.4 diatas dapat kita lihat bahwa pengujian API Login berhasil dengan menggunakan data email dari proses register sebelumnya, saat login berhasil token jwt otomatis akan terbuat random yang berfungsi sebagai keamanan dalam mengakses API berikutnya.

3. Pengujian API Update Profil

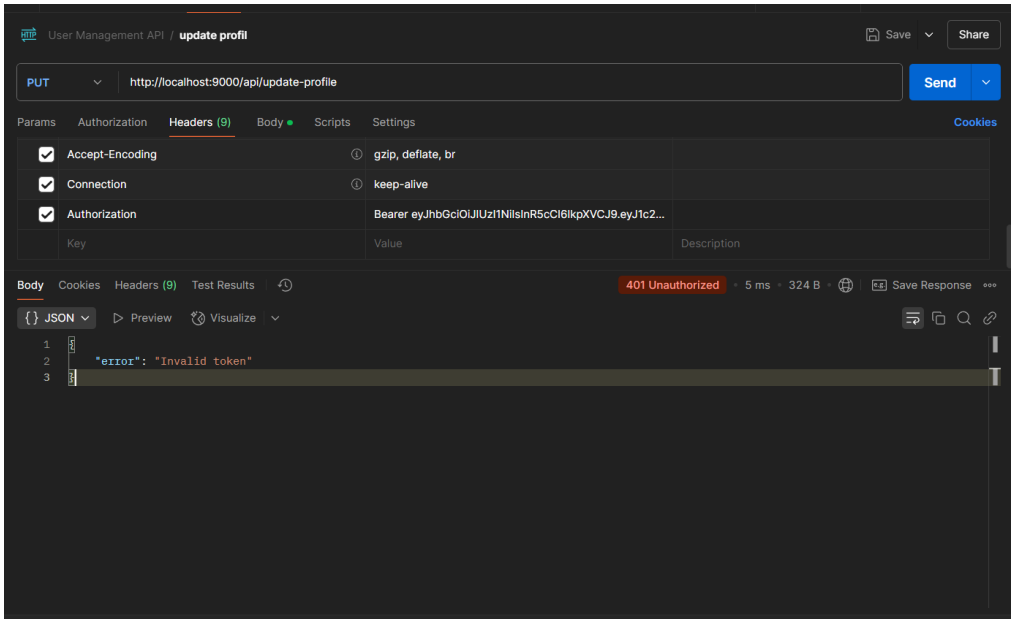
Setelah berhasil melakukan proses login, pengujian API Update Profil dilakukan menggunakan token autentikasi yang dihasilkan saat login. Token ini diperlukan untuk memastikan bahwa hanya pengguna yang telah terverifikasi dapat mengakses dan memperbarui profil mereka. Dalam pengujian ini, endpoint PUT digunakan untuk mengirimkan data pembaruan seperti nama, email, nomor telepon, atau informasi lainnya. Data yang dikirim dalam format JSON disertai dengan header Authorization yang berisi token. Hasil pengujian menunjukkan bahwa profil berhasil diperbarui jika token valid, dengan respons berupa status

kode 200 OK dan data pengguna terbaru. Jika token tidak valid atau tidak disertakan, server mengembalikan pesan error dengan status kode seperti 401 Unauthorized atau 403 Forbidden. Gambar hasil pengujian dapat dilihat pada ilustrasi berikut:



Gambar 5. 5 Pengujian API Update Profil

Dari gambar 5.5 diatas dapat kita lihat bahwa status update profil berhasil dengan menggunakan token yang kita dapatkan pada proses login sebelumnya, token berguna sebagai keamanan. Namun jika token yang dimasukan acak atau beda dengan token yang dihasilkan saat login, maka akan muncul pernyataan tidak memiliki izin atau Unauthorized yang dapat kita lihat pada gambar dibawah ini:



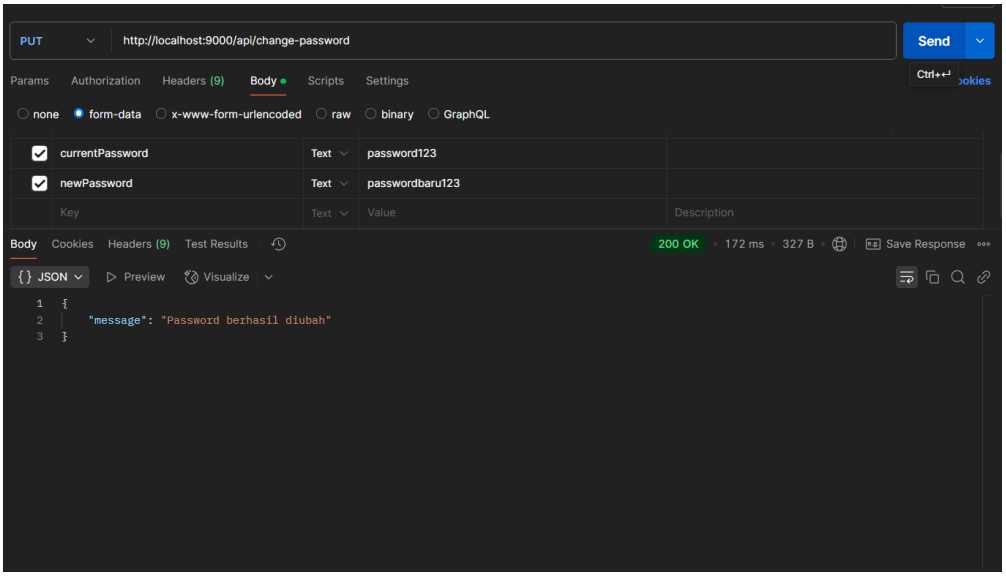
Gambar 5. 6 Pengujian API Update Profil Gagal

Pada gambar 5.6 diatas terlihat penulis gagal untuk memperbarui profil karena token yang dimasukan berbeda dengan token yang didapat saat login, keterangannya juga dapat kita lihat yaitu invalid token.

4. Pengujian API Ubah Kata Sandi

Pengujian API Ubah Kata Sandi dilakukan untuk memastikan bahwa pengguna dapat mengubah kata sandi mereka dengan aman. Permintaan dikirimkan ke endpoint PUT dengan menyertakan token autentikasi yang dihasilkan saat login, kata sandi lama, dan kata sandi baru. Token digunakan untuk memastikan bahwa hanya pengguna yang valid dapat mengakses fitur ini. Jika token valid dan kata sandi lama sesuai, server akan memperbarui kata sandi pengguna dan memberikan respons dengan status kode 200 OK serta pesan sukses. Jika token tidak valid atau kata sandi lama salah, server akan mengembalikan pesan

error dengan status kode seperti 401 Unauthorized. Hasil pengujian dapat dilihat pada gambar yang terkait:



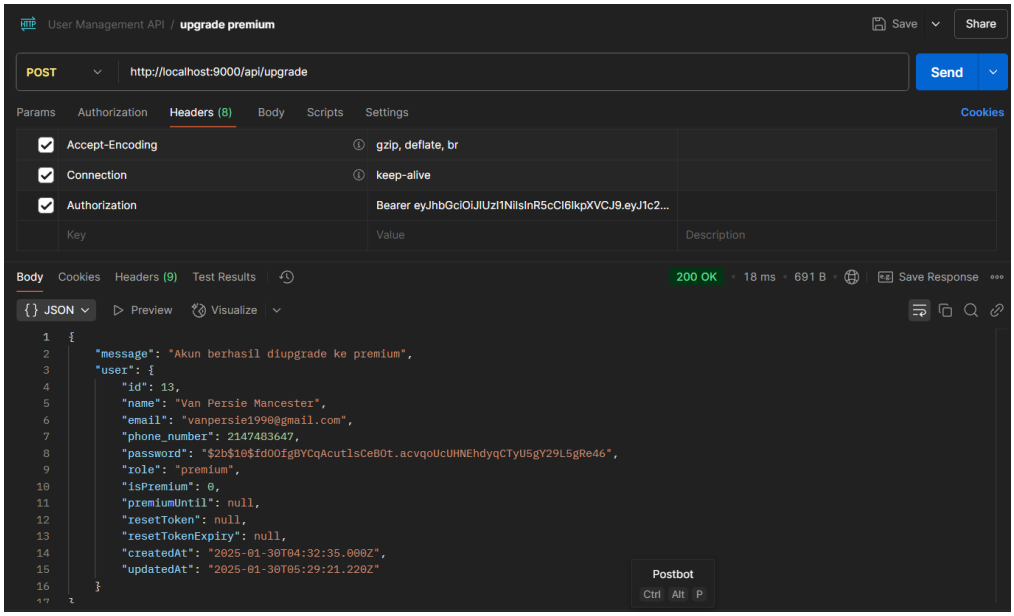
Gambar 5. 7 Pengujian API Change Password

Dari gambar diatas dapat kita lihat bahwa API Change Password berhasil dan tidak ada kendala, dengan memasukan token yang benar kita dapat mengakses semua API yang menggunakan token sebagai pengamannya. Apabila token dimasukan berbeda maka hasilnya sama seperti tadi yaitu gagal dan mendapat keterangan invalid token.

5. Pengujian API Upgrade Premium

Pengujian API Upgrade Premium dilakukan untuk memastikan bahwa pengguna dapat meningkatkan status akun mereka menjadi premium. Permintaan dikirimkan ke endpoint tertentu, misalnya PUT, dengan menyertakan token autentikasi yang didapat dari proses login. Berdasarkan token tersebut, sistem secara otomatis membaca data akun pengguna dan memperbarui statusnya menjadi premium. Jika proses berhasil, server memberikan respons berupa status kode 200

OK dengan pesan sukses dan data pengguna terbaru. Jika token tidak valid atau terjadi kesalahan lain, server mengembalikan pesan error dengan status kode seperti 401 Unauthorized atau 403 Forbidden. Pengujian ini memastikan bahwa fitur upgrade premium berjalan dengan benar dan aman. Pengujian dapat kita lihat pada gambar dibawah ini:



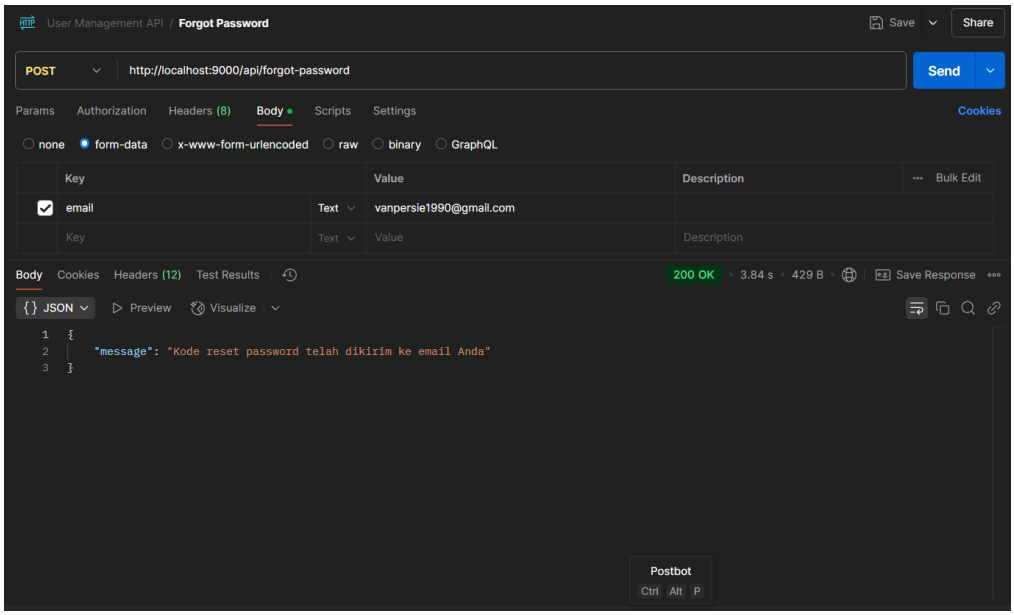
Gambar 5. 8 Pengujian API Upgrade Premium

Pada gambar 5.8 dapat kita lihat bahwa API upgrade akun ke premium berhasil dan memberikan keterangan akun yang premium. Role otomatis berubah dari biasa menjadi premium dan sekarang pengguna akun dapat mengakses konten premium.

6. Pengujian API Forgot Password

Pengujian API Forgot Password dilakukan untuk memastikan bahwa sistem dapat menangani permintaan pengguna yang lupa kata sandi mereka. Proses ini melibatkan pengiriman permintaan POST ke endpoint tertentu dengan

memasukkan email pengguna yang terdaftar. Server akan memverifikasi email tersebut, dan jika valid, sistem akan mengirimkan tautan atau kode reset kata sandi ke email pengguna. Respons server untuk permintaan yang valid biasanya berupa status kode 200 OK dengan pesan seperti "Email untuk reset kata sandi telah dikirim." Namun, jika email tidak terdaftar, responsnya dapat berupa status kode 404 Not Found dengan pesan error. Pengujian ini memastikan bahwa fitur lupa kata sandi berfungsi sesuai harapan dan aman dari penyalahgunaan. Gambar pengujian lupa password dapat kita lihat dibawah ini:

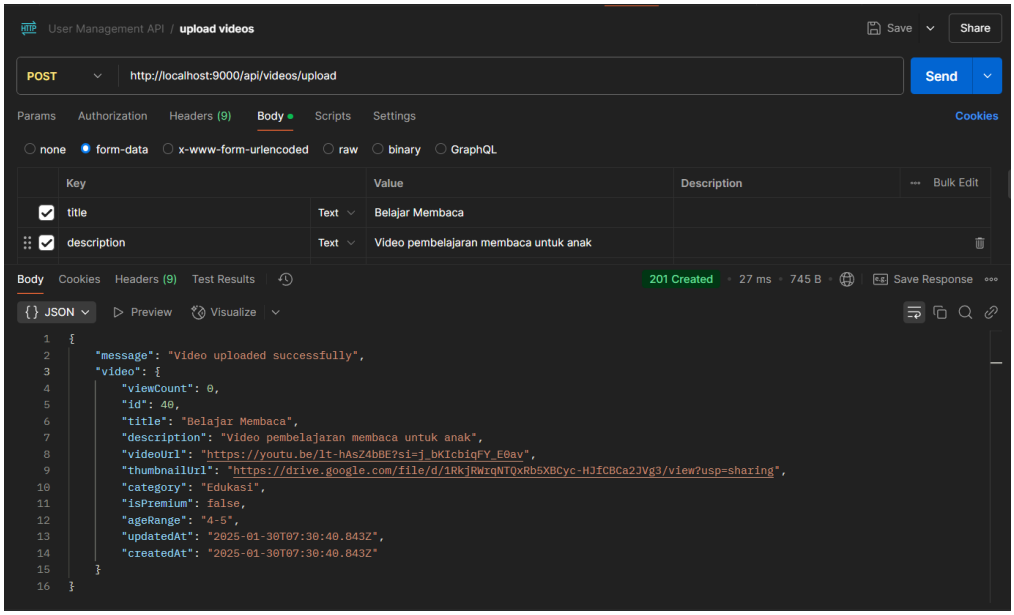


Gambar 5. 9 Pengujian API Forgot Password

Dari gambar 5.9 diatas dapat kita lihat bahwa API berhasil berjalan dengan baik. Pesan “kode reset password telah dikirim ke email anda” berarti kode authentikasi telah dikirim ke email pengguna sebagai bentuk bahwa user itu sendiri yang meminta request forgot password. Kode yang dikirim berisikan 5 nomor random yang dapat dilihat pada email yang meminta pengajuan forgot password.

7. Pengujian API Upload Videos

Pengujian API Upload Videos dilakukan untuk memastikan sistem dapat menerima dan memproses video yang diunggah. Pengujian menggunakan endpoint POST dengan URL http://localhost:9000/api/videos/upload, di mana data yang dikirim mencakup judul (title) dan deskripsi (description) melalui form-data. Token autentikasi diperlukan untuk memastikan hanya pengguna berizin yang dapat mengakses fitur ini. Pengujian API dapat kita lihat pada gambar dibawah ini:

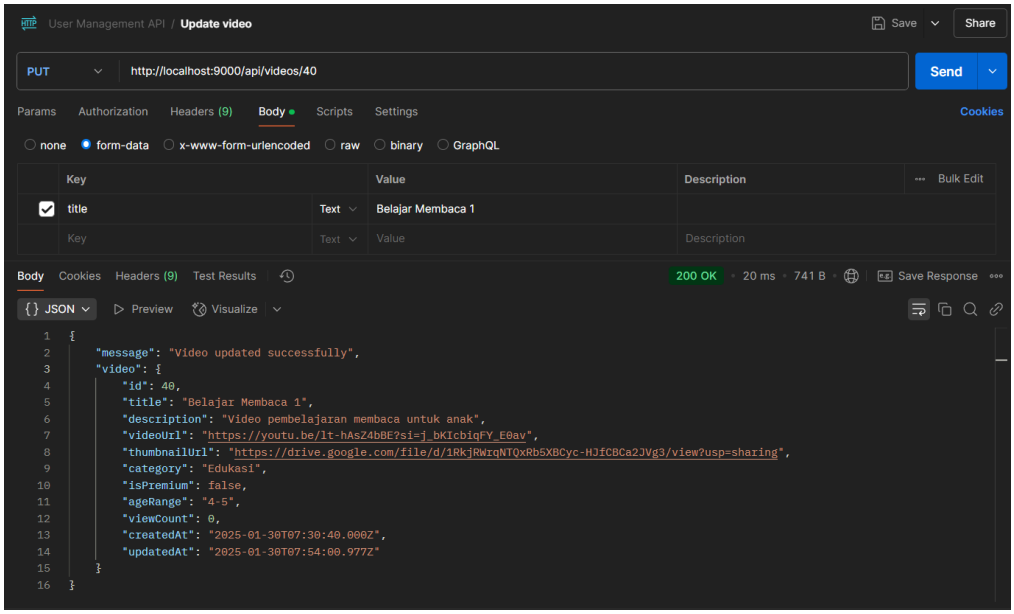


Gambar 5. 10 Pengujian API Upload Video

Dari gambar diatas dapat kita lihat pengujian API upload video. Pengujian berhasil dengan kode 201. Video masuk dan tersimpan didatabase dan siap di tampilkan ke user. Dapat kita ketahui bahwa proses memasukan video hanya bisa digunakan akun dengan role admin dan kemudian baru dapat memasukan video, musik dan bacaan.

8. Pengujian API Update Video

Pengujian API Update Video dilakukan untuk memastikan bahwa sistem dapat memperbarui data video berdasarkan ID yang diberikan. Kemudian ID tersebut berguna untuk memilih video dan kemudian dapat diupdate isi data videonya. Pengujian API dapat kita lihat pada gambar dibawah ini:



Gambar 5. 11 Pengujian API Update Video

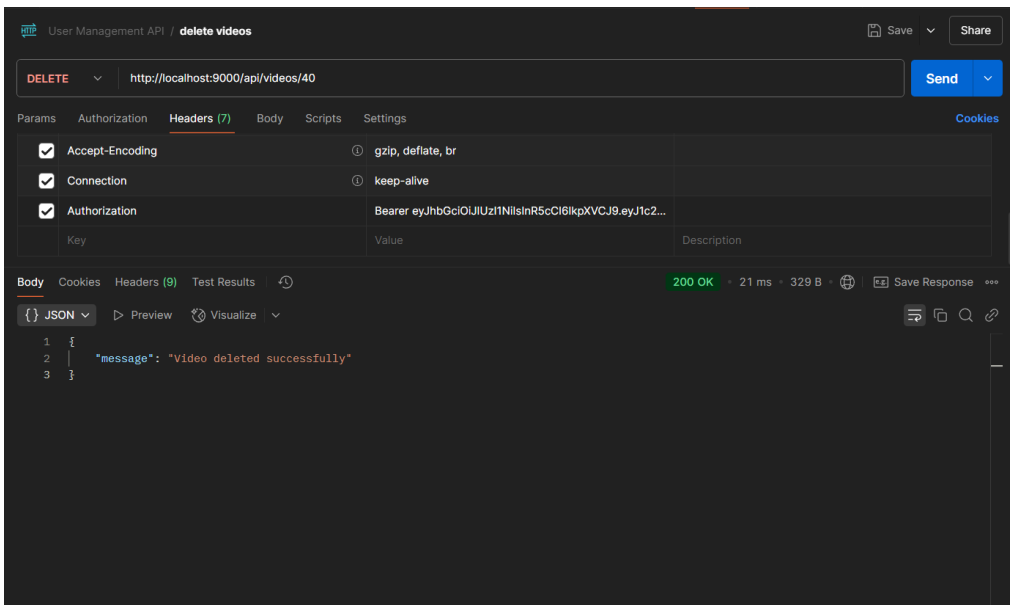
Dari gambar 5.11 diatas dapat kita lihat bahwa Pengujian ini menggunakan endpoint PUT dengan URL http://localhost:9000/api/videos/40, di mana 40 merupakan ID video yang akan diperbarui. Data yang dikirimkan berupa form-data, seperti judul baru (title) dengan nilai Belajar Membaca 1. Hasil pengujian menunjukkan bahwa video berhasil diperbarui dengan status kode 200 OK.

Respons server berisi pesan "Video updated successfully" dan detail video yang diperbarui, seperti ID (40), judul baru (Belajar Membaca 1), deskripsi (Video pembelajaran membaca untuk anak), URL video, thumbnail URL, kategori

(Edukasi), status premium (false), rentang usia (4-5 tahun), jumlah tampilan (0), waktu pembuatan (2025-01-30T07:30:40.000Z), dan waktu pembaruan terakhir (2025-01-30T07:54:00.977Z). Pengujian ini menunjukkan bahwa API mampu memproses pembaruan data video dengan baik. Token autentikasi juga diperlukan untuk memastikan keamanan akses fitur ini.

9. Pengujian API Delete Video

Pengujian API Delete Video dilakukan untuk memastikan bahwa sistem dapat menghapus data video berdasarkan ID yang diberikan. Hasil pengujian diharapkan menunjukkan respons dengan status kode 200 OK atau kode serupa yang menandakan keberhasilan penghapusan. respons server akan berisi pesan konfirmasi "Video deleted successfully" tanpa data video, untuk memastikan bahwa video dengan ID tersebut telah dihapus dari sistem. token autentikasi diperlukan untuk memastikan bahwa hanya pengguna yang memiliki izin yang dapat menghapus video. Pengujian API dapat kita lihat pada gambar dibawah ini:



Gambar 5. 12 Pengujian API Delete Video

Dari gambar 5.12 diatas, dapat kita lihat bahwa proses berjalan lancar. Penghapusan berhasil dengan ditandain kode 200 tanpa ada hambatan, hal ini berlaku juga untuk konten musik dan bacaan.

1.2.3 Pengujian Keamanan

Pada tahap ini penulis mendeskripsikan beberapa keamanan yang telah dijalankan. Keamanan dibuat untuk untuk mencegah potensi celah keamanan dalam sistem. Berikut keamanan yang telah diterapkan yaitu:

1. SQL Injection

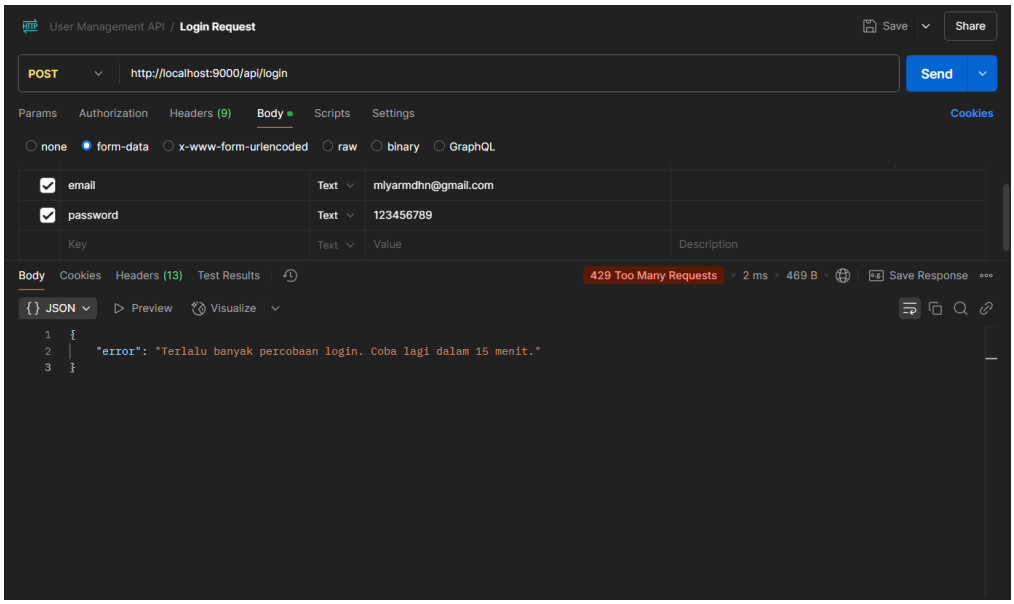
SQL Injection merupakan serangan yang dilakukan dengan menyisipkan perintah SQL berbahaya ke dalam input pengguna dengan tujuan mengakses atau memanipulasi database secara ilegal. Untuk mencegah serangan ini, backend CihuyKids menggunakan ORM (Object-Relational Mapping) Sequelize, yang secara otomatis melakukan parameterisasi query. Dengan demikian, setiap input dari pengguna tidak langsung dieksekusi sebagai perintah SQL, melainkan diproses dengan aman oleh ORM sebelum dikirim ke database.

2. Cross-Site Scripting (XSS)

XSS merupakan serangan di mana penyerang menyisipkan skrip berbahaya ke dalam halaman web yang kemudian dieksekusi di sisi pengguna. Untuk mengatasi hal ini, backend CihuyKids memastikan bahwa setiap input yang diterima dari pengguna divalidasi dan disanitasi sebelum disimpan ke dalam database atau ditampilkan kembali kepada pengguna. Teknik sanitasi ini mencegah skrip berbahaya dijalankan di dalam lingkungan aplikasi.

3. Rate Limiting

Serangan brute-force terjadi ketika penyerang mencoba berbagai kombinasi username dan password dalam waktu singkat untuk mendapatkan akses ke akun pengguna. Untuk mencegah serangan ini, backend telah mengimplementasikan rate limiter pada endpoint login dan reset password. Dengan pembatasan ini, pengguna hanya dapat mencoba login dalam jumlah terbatas dalam kurun waktu tertentu (misalnya 5 kali dalam 15 menit). Jika melebihi batas, sistem akan memblokir permintaan sementara, sehingga mencegah upaya brute-force yang dilakukan oleh penyerang. Kita dapat melihat bagaimana Rate Limiting bekerja pada gambar dibawah ini:



Gambar 5. 13 Penerapan Rate Limiting

Dari gambar 5.13 diatas dapat kita lihat bagaimana rate limiting bekerja dimana saat kita mencoba login dengan kesalahan email atau sandi lebih dari 5 kali maka akan muncul pesan terlalu banyak percobaan login dan coba lagi 15 menit,

hal ini juga sama ketika kita ingin mengajukan lupa password, apabila kita meminta lebih dari 3 kali kode reset maka akan muncul keterangan yang sama seperti gambar diatas.

1.3 ANALISA HASIL YANG DICAPAI

Sebuah sistem yang penulis buat pastinya memiliki kelebihan dan kekurangan, begitu pula dengan perancangan backend untuk website CihuyKids. Berikut adalah analisis kelebihan dan kekurangan sistem yang telah dikembangkan.

1.3.1 Kelebihan *Website*

1. CihuyKids dapat diakses di berbagai perangkat, seperti handphone dan laptop.
2. Website memiliki antarmuka yang ramah anak dan menarik, sehingga sesuai dengan target pengguna.
3. Website memiliki fitur shorting media berdasarkan usia, sehingga konten yang ditampilkan lebih sesuai dengan kategori umur pengguna.

1.3.2 Kekurangan *Website*

1. Tidak mengadopsi payment gateway dan Proses pembayaran untuk upgrade akun masih dilakukan secara manual, yang dapat memerlukan waktu lebih lama.
2. Pilihan konten masih terbatas, sehingga variasi hiburan dan edukasi bagi anak-anak belum maksimal.
3. Belum tersedia fitur kontrol orang tua yang lebih mendalam, seperti pengaturan waktu akses atau pembatasan konten tertentu.