4.2 Pengujian Sistem

Pengujian sistem ini bertujuan untuk memastikan bahwa sistem autentikasi berbasis JSON Web Token (JWT) yang dirancang memenuhi standar keamanan dan efektivitas yang diharapkan. Fokus utama pengujian akan berada pada aspek backend dari aplikasi, sesuai dengan ruang lingkup penelitian yang mencakup penggunaan JWT dan bahasa pemrograman Golang dengan framework Echo. Framework Echo dipilih karena kemampuannya dalam membangun aplikasi backend, serta mendukung pengelolaan autentikasi berbasis JWT. Echo dikenal dalam menangani banyak permintaan secara bersamaan untuk aplikasi yang membutuhkan verifikasi token dalam waktu nyata, seperti pada sistem autentikasi.

Pengujian fungsi autentikasi akan dilakukan untuk memverifikasi bahwa proses autentikasi berjalan dengan standar, termasuk validasi token, verifikasi akses, dan penanganan sesi penggun. Echo menyediakan sistem middleware yang fleksibel yang dapat mengelola autentikasi, termasuk validasi JWT dan pengelolaan sesi. Selain itu, pengujian keamanan akan menjadi prioritas utama untuk mengidentifikasi dan mengatasi potensi kerentanan, seperti manipulasi token dan serangan pemalsuan, serta memastikan perlindungan data sensitif. Echo juga mendukung enkripsi dan perlindungan token, yang membantu menjaga integritas data dan mencegah potensi ancaman keamanan.

Pengujian kinerja akan dilakukan untuk mengevaluasi responsivitas dan kemampuan sistem dalam menangani berbagai beban pengguna, memastikan bahwa sistem dapat mengelola trafik tinggi dengan efisien dan memberikan

performa yang optimal. Dengan Echo, sistem dapat menangani banyak permintaan API berkat optimisasi internal yang memastikan aplikasi tetap responsif meskipun pada beban tinggi. Terakhir, pengujian integrasi akan mengonfirmasi bahwa sistem autentikasi berfungsi dengan baik dalam konteks situs penjualan tiket online dan dapat berinteraksi dengan komponen backend lainnya.

4.2.1 Pengujian Endpoint "/user/register"

```
{
    "name": "string",
    "username": "string",
    "password": "string",
    "email": "string",
    "phonenumber": "string"
}
```

Gambar 4. 13 Rancangan API Endpoint "/user/register"

Atribut name digunakan untuk menyimpan nama lengkap pengguna, yang bertujuan untuk memberikan identifikasi personal dalam aplikasi. Atribut username adalah identitas unik yang dirancang untuk membedakan setiap pengguna dalam sistem, terutama dalam proses autentikasi. Atribut password menyimpan kata sandi yang diberikan oleh pengguna, yang akan diolah menggunakan metode hashing sebelum disimpan dalam basis data untuk menjaga kerahasiaan dan keamanan data pengguna. Selanjutnya, atribut email berfungsi sebagai alat komunikasi yang juga dapat digunakan untuk verifikasi akun atau pemulihan akses, sedangkan phonenumber merupakan nomor telepon pengguna yang dapat digunakan untuk pengiriman notifikasi atau sebagai metode autentikasi tambahan, seperti pengiriman kode OTP (One-Time Password).

```
{
    "name": "Kibar",
    "username": "Iballl",
    "password": "***",
    "email": "kibar@gmail.com",
    "phonenumber": "082154500431"
}
```


Gambar 4. 14 Endpoint "/user/register"

Gambar 4.13 merupakan data yang dimasukkan oleh *user* untuk melakukan registrasi pada sistem penjualan tiket online agar dapat masuk dan dapat mengakses fitur yang tersedia didalamnya.


```
{
    "code": 200,
    "status": "You have successfully registered as User",
    "data": {
        "id": "USER-004",
        "name": "Kibar",
        "username": "Iballl",
        "email": "kibar@gmail.com",
        "phonenumber": ""
    }
}
```


Gambar 4. 15 Hasil Pengujian Endpoint "/user/register"

Pengujian endpoint /user/register berhasil dilakukan dengan kode status HTTP 200, yang menandakan bahwa pendaftaran pengguna berhasil. Respons dari endpoint ini menunjukkan status "You have successfully registered as User", mengonfirmasi bahwa pengguna baru telah terdaftar dengan sukses. Hasil ini menunjukkan bahwa proses pendaftaran berjalan dengan baik dan tidak ada kesalahan yang terdeteksi, sehingga pengguna baru dapat mulai menggunakan sistem dengan informasi yang telah diberikan.

4.2.2 Pengujian Endpoint “/admin/register”

```
{
    "name": "string",
    "username": "string",
    "password": "string",
    "email": "string",
    "phonenumber": "string"
}
```

Gambar 4. 16 Rancangan API Endpoint "/admin/register"

Atribut name digunakan untuk menyimpan nama lengkap pengguna, yang bertujuan untuk memberikan identifikasi personal dalam aplikasi. Atribut username adalah identitas unik yang dirancang untuk membedakan setiap pengguna dalam sistem, terutama dalam proses autentikasi. Atribut password menyimpan kata sandi yang diberikan oleh pengguna, yang akan diolah menggunakan metode hashing sebelum disimpan dalam basis data untuk menjaga kerahasiaan dan keamanan data pengguna. Selanjutnya, atribut email berfungsi sebagai alat komunikasi yang juga dapat digunakan untuk verifikasi akun atau pemulihan akses, sedangkan phonenumber merupakan nomor telepon pengguna yang dapat digunakan untuk pengiriman notifikasi atau sebagai metode autentikasi tambahan, seperti pengiriman kode OTP (One-Time Password).

```
{
    "name": "Topek",
    "username": "coba",
    "password": "888",
    "email": "Topek@gmail.com",
    "phonenumber": "081234567890"
}
```

Gambar 4. 17 Endpoint "/admin/register"

Gambar 4.13 merupakan data yang dimasukkan oleh *admin* untuk melakukan registrasi pada sistem penjualan tiket online agar dapat masuk dan dapat mengakses fitur yang tersedia didalamnya.

```
{
    "code": 200,
    "status": "You have successfully registered as Admin",
    "data": {
        "id": "ADMIN-004",
        "name": "Topek",
        "username": "coba",
        "email": "Topek@gmail.com",
        "created_at": "2024-07-23T04:53:22.2918966+07:00"
    }
}
```


Gambar 4. 18 Hasil Pengujian Endpoint "/admin/register"

Pengujian endpoint /user/register menunjukkan hasil yang sukses dengan kode status HTTP 200, yang menandakan bahwa pendaftaran pengguna baru dengan peran Admin telah berhasil dilakukan. Respons yang diterima mengonfirmasi keberhasilan pendaftaran dengan pesan "You have successfully registered as Admin", menegaskan bahwa pengguna kini terdaftar sebagai Admin di sistem. Hasil ini juga menegaskan bahwa sistem berhasil mengelola pendaftaran pengguna.

4.2.3 Pengujian Endpoint “/login-user”


```
{
    "username": "string",
    "password": "string",
    "email": "string",
}
```


Gambar 4. 19 Rancangan API Endpoint "/login-user"

Endpoint /login-user berfungsi untuk mengautentikasi pengguna dalam sistem. Proses autentikasi dimulai dengan pengguna mengirimkan permintaan berbentuk JSON yang memuat atribut username, password dan email. Atribut ini digunakan untuk memvalidasi identitas pengguna dengan mencocokkan data yang dimasukkan dengan informasi yang tersimpan dalam basis data. Jika data cocok, sistem akan memberikan token JWT (JSON Web Token) sebagai respons.

```
{
    "username": "Iballl",
    "password": "123",
    "email": "kibar@gmail.com",
}
```


Gambar 4. 20 Endpoint "/login-user"

Pada gambar 4.17 *user* memasukkan data yang sebelumnya sudah didaftarkan pada *endpoint* registrasi dan nantinya akan diperiksa kredibilitasnya oleh sistem sebelum diberikan akses untuk masuk kedalam sistem.


```
{
    "code": 200,
    "status": "login berhasil",
    "data": {

        "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX
2lkIjoiVVNFUi0wMDQiLCJuYW1lIjoiS2liYXIiLCJlbWFpbCI6ImtpYmFyQGdtY
WlsLmNvbSIsImlzcyI6ImtlbG9tcG9rLTAzIiwiZXhwIjoxNzIxNjg5MDYyfQ.Zk
aG5thy1T0BYNfYyyjNpwS7Ocnmj5KsU6ouDaNB0uc"
    }
}
```


Gambar 4. 21 Hasil Pengujian Endpoint "/login-user"

Respons dari endpoint ini menunjukkan pesan "login berhasil", yang menegaskan bahwa kredensial pengguna telah diterima dan diproses dengan benar. Dalam data respons, terdapat token autentikasi yang diberikan dalam format JSON Web Token (JWT). Token ini berfungsi sebagai bukti bahwa pengguna telah berhasil login dan dapat digunakan untuk mengakses layanan yang memerlukan autentikasi. Token ini dapat digunakan untuk melakukan permintaan ke endpoint yang memerlukan otorisasi.

4.2.4 Pengujian Endpoint “/login-admin”

```
{
    "username": "string",
    "password": "string",
    "email": "string",
}
```

Gambar 4. 22 Rancangan API Endpoint "/login-user"

Endpoint /login-user berfungsi untuk mengautentikasi pengguna dengan role admin dalam sistem. Proses autentikasi dimulai dengan pengguna mengirimkan permintaan berbentuk JSON yang memuat atribut username, password dan email. Atribut ini digunakan untuk memvalidasi identitas pengguna dengan mencocokkan data yang dimasukkan dengan informasi yang tersimpan dalam basis data. Jika data cocok, sistem akan memberikan token JWT (JSON Web Token) sebagai respons.

```
{
    "name": "Topek",
    "username": "coba",
    "password": "888",
    "email": "Topek@gmail.com",
    "phonenumber": "081234567890"
}
```

Gambar 4. 23 Endpoint "/login-admin"

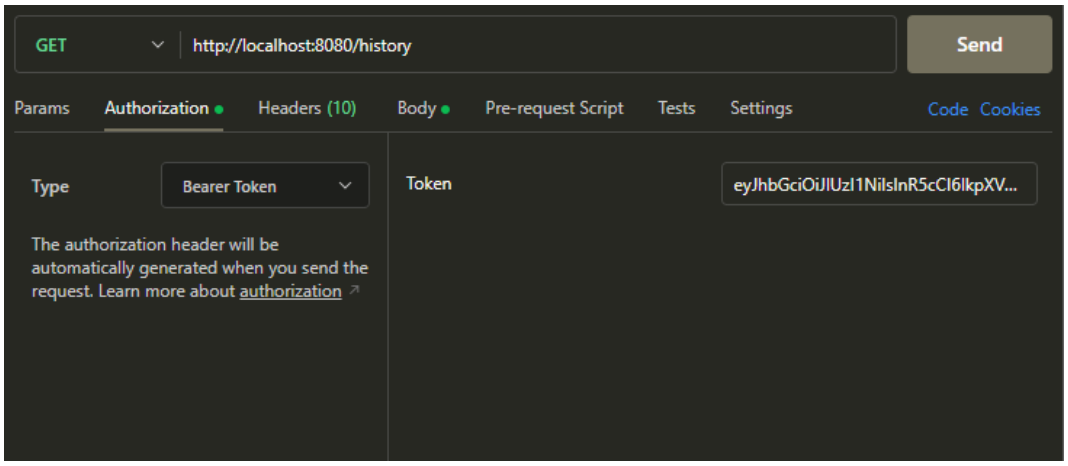
Pada gambar 4.17 *admin* memasukkan data yang sebelumnya sudah didaftarkan pada *endpoint* registrasi dan nantinya akan diperiksa kredibilitasnya oleh sistem sebelum diberikan akses untuk masuk kedalam sistem.

```
{
    "code": 200,
    "status": "login berhasil",
    "data": {
        "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX
2lkIjoiQURNSU4tMDA0IiwibmFtZSI6IlRvcGVrIiwiZW1haWwiOiJUb3Bla0Bnb
WFpbC5jb20iLCJpc3MiOiJLZWxvbXBvay0wMyIsImV4cCI6MTcyMTY4OTI4MH0.t
DiD6wvaGQB5_Kuq46fi2ciuyoxiKECRc8XtiNaiC-w"
    }
}
```


Gambar 4. 24 Hasil Pengujian Endpoint "/login-admin"

JWT token berfungsi sebagai bukti bahwa pengguna telah berhasil melakukan autentikasi. Token ini digunakan untuk memverifikasi identitas pengguna dan memberi akses yang sesuai ke berbagai fitur atau layanan dalam aplikasi. Token yang diberikan dalam respons login menyertakan informasi penting yang dapat digunakan oleh server untuk mengidentifikasi pengguna dan memvalidasi permintaan yang dilakukan oleh pengguna tersebut, serta memastikan bahwa akses diberikan hanya kepada pengguna yang sah dan terautentikasi. Dengan Hasil diatas dapat di katakan bahwa pengujian berhasil dalam memvalidasi dan melakukan autentikasi pada user yang melakukan login ke sistem.

4.2.5 Pengujian Endpoint “/history” Oleh Admin



Gambar 4. 25 Bearer Token Validation Admin

Gambar 4.21 merupakan tampilan dari pengujian menggunakan postman menggunakan endpoint yang sudah disebutkan diatas, pengujian ini bertujuan untuk melihat apakah fitur yang dibuat dapat berjalan dengan semestinya.

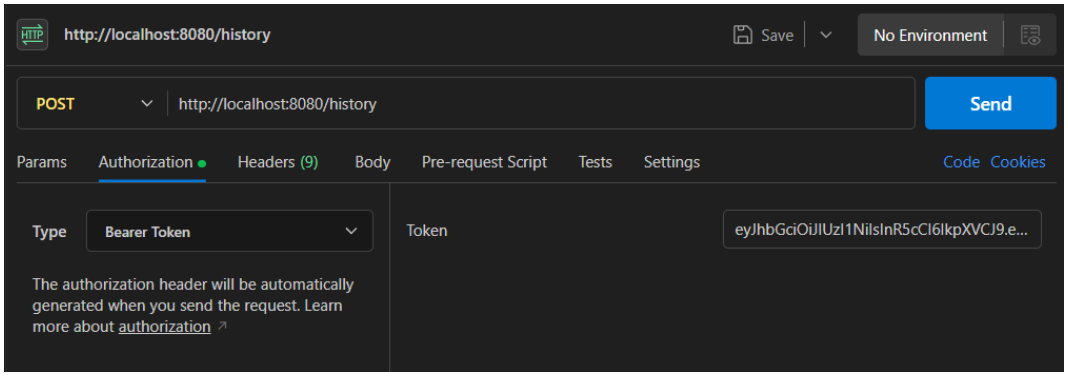

```
[
    {
        "ID": "trans1",
        "UserID": "user1",
        "EventID": "event1",
        "Date": "2024-06-02",
        "Quantity": 2,
        "TotalPrice": 200000,
        "Status": "confirmed",
        "CreatedAt": "2024-05-28T21:55:57.125949+07:00"
    },
    {
        "ID": "trans2",
        "UserID": "user2",
        "EventID": "event2",
        "Date": "2024-07-02",
        "Quantity": 1,
        "TotalPrice": 150000,
        "Status": "pending",
        "CreatedAt": "2024-05-28T21:55:57.125949+07:00"
    },
    {
        "ID": "trans3",
        "UserID": "user3",
        "EventID": "event3",
        "Date": "2024-08-02",
        "Quantity": 3,
        "TotalPrice": 600000,
        "Status": "confirmed",
        "CreatedAt": "2024-05-28T21:55:57.125949+07:00"
    }
]
```


Gambar 4. 26 Tampilan History Setelah Validasi Admin

Setelah memasukkan JWT token ke bagian Authorization Bearer Token, tampilan yang muncul adalah daftar transaksi yang dapat diakses oleh admin yang terautentikasi. Informasi yang ditampilkan memberikan gambaran yang jelas tentang transaksi yang telah dilakukan oleh pengguna yang terautentikasi, termasuk detail harga, kuantitas, tanggal acara, dan status transaksi. Keseluruhan tampilan ini

menunjukkan bagaimana sistem memproses dan mengelola data transaksi untuk pengguna yang valid dan terautentikasi.

4.2.6 Pengujian Endpoint "/history" Oleh User



Gambar 4. 27 Bearer Token Validation User

Gambar 4.21 merupakan tampilan dari pengujian menggunakan postman menggunakan endpoint yang sudah disebutkan diatas, pengujian ini bertujuan untuk melihat apakah role yang digunakan untuk mengakses sesuai dengan ketentuan dari system atau tidak, jika hasilnya role yang digunakan tidak diizinkan maka akan muncul peringatan dari sistem.

```
{
    "code": 405,
    "status": "Method Not Allowed",
    "data": null
}
```

Gambar 4. 28 Tampilan History Setelah Validasi User

Muncul pesan bahwa metode yang digunakan tidak diperbolehkan, hal ini terjadi karena user memaksa masuk ke endpoint "/history", yang hanya bisa di akses dengan token dari admin, sehingga aksesnya ditolak dan tidak ada data yang di tampilkan.