Bagian 1: Auth Routes (Untuk Autentikasi Pengguna)

Endpoint-endpoint ini digunakan untuk mendaftarkan, login, dan logout pengguna, serta untuk memverifikasi informasi pengguna yang sedang login.

1. POST /register: Digunakan untuk mendaftarkan pengguna baru. Pengguna perlu mengirimkan username, password, dan email. Jika data valid, pengguna baru akan dibuat dan menerima pesan "Registrasi berhasil!".
2. POST /login: Digunakan untuk login pengguna dengan username dan password. Jika kredensial benar, server mengembalikan JWT token yang bisa digunakan untuk akses endpoint yang dilindungi (seperti untuk mengakses data game, genre, dll.).
3. GET /auth: Digunakan untuk mengambil informasi pengguna yang sedang login, jika pengguna sudah mempunyai token valid. Jika token valid, server akan mengembalikan data pengguna.
4. DELETE /logout: Digunakan untuk logout. Server akan menghapus sesi pengguna yang sedang aktif.

Bagian 2: Game Routes (Untuk Mengelola Game)

Endpoint ini digunakan untuk mengelola data game yang ada di backend.

1. GET /games: Digunakan untuk mengambil daftar semua game yang ada di sistem. Hanya pengguna yang sudah login dan memiliki token valid yang bisa mengaksesnya.
2. GET /games/:id: Digunakan untuk mengambil informasi game berdasarkan ID. Pengguna harus memberikan ID game yang ingin diambil.
3. POST /games: Digunakan untuk menambahkan game baru ke sistem. Pengguna harus mengirimkan data game baru, termasuk judul, deskripsi, harga, dan tahun rilis.
4. PATCH /games/:id: Digunakan untuk memperbarui informasi game berdasarkan ID. Pengguna dapat memperbarui data seperti judul atau deskripsi.
5. DELETE /games/:id: Digunakan untuk menghapus game berdasarkan ID. Pengguna yang sudah login dapat menghapus game yang ada di sistem.

Bagian 3: Genre Routes (Untuk Mengelola Genre Game)

Genre digunakan untuk mengklasifikasikan game ke dalam kategori tertentu (misalnya, Adventure, Action, dll.).

1. GET /genres: Digunakan untuk mengambil daftar semua genre yang ada.
2. POST /genres: Digunakan untuk menambahkan genre baru ke sistem.
3. DELETE /genres/:id: Digunakan untuk menghapus genre berdasarkan ID.

Bagian 4: Pembelian Routes (Untuk Mengelola Pembelian Game)

Pembelian digunakan untuk merekam transaksi game yang dibeli oleh pengguna.

1. GET /pembelian: Digunakan untuk mengambil daftar semua pembelian game yang dilakukan oleh pengguna.
2. POST /pembelian: Digunakan untuk menambahkan pembelian baru game oleh pengguna.
3. PATCH /pembelian/:id: Digunakan untuk memperbarui pembelian yang sudah ada berdasarkan ID.
4. DELETE /pembelian/:id: Digunakan untuk menghapus pembelian game berdasarkan ID.

Bagian 5: Review Routes (Untuk Mengelola Ulasan Game)

Ulasan memberikan pengguna kemampuan untuk memberi rating dan komentar pada game yang sudah mereka mainkan.

1. GET /reviews: Digunakan untuk mengambil semua ulasan yang ada untuk game yang telah dibeli atau dimainkan.
2. POST /reviews: Digunakan untuk menambahkan ulasan baru tentang game. Pengguna memberikan rating dan komentar tentang game.
3. PATCH /reviews/:id: Digunakan untuk memperbarui ulasan berdasarkan ID.
4. DELETE /reviews/:id: Digunakan untuk menghapus ulasan berdasarkan ID.

Penjelasan Umum tentang Status Code:

* 200 OK: Permintaan berhasil dan data yang diminta dikembalikan.
* 201 Created: Data baru berhasil dibuat (biasanya untuk POST request).
* 400 Bad Request: Ada masalah dengan data yang dikirimkan, seperti validasi input yang gagal.
* 404 Not Found: Endpoint tidak ditemukan atau ID yang diminta tidak ada.
* 401 Unauthorized: Pengguna tidak terautentikasi atau token tidak valid.
* 500 Internal Server Error: Terjadi kesalahan pada server, biasanya disebabkan oleh bug di backend.

Bagaimana API ini Bekerja:

* Setiap endpoint API (misalnya /games, /genres, /register) berfungsi untuk mengakses atau mengubah data tertentu (seperti game, genre, pembelian, atau review).
* Token JWT digunakan untuk autentikasi pengguna. Untuk dapat mengakses data yang dilindungi, pengguna perlu mengirimkan token yang valid di header Authorization.
* POST digunakan untuk menambahkan data baru, PATCH untuk memperbarui data yang ada, dan DELETE untuk menghapus data.
* GET digunakan untuk mengambil data dari server.

Contoh Penggunaan di Aplikasi Frontend:

1. Mendaftar Pengguna:  
   Pengguna bisa mendaftar melalui formulir dengan mengirimkan username, email, dan password. API /register akan membuat akun pengguna baru.
2. Login Pengguna:  
   Setelah registrasi, pengguna dapat login melalui API /login untuk mendapatkan token JWT. Token ini digunakan untuk akses lebih lanjut ke data yang dilindungi.
3. Mengambil Daftar Game:  
   Pengguna dapat mengakses semua game melalui API /games setelah login. API ini akan mengembalikan daftar game yang tersedia.

**Dokumentasi API - Project TCC**

**Base URL**

<http://localhost:5000>

**Authentication**

Semua endpoint **/games, /genres, /pembelian, /reviews** membutuhkan autentikasi menggunakan **JWT (JSON Web Token)**. Token harus ditambahkan di header Authorization dengan format:

Authorization: Bearer <your\_token>

**1. Auth Routes**

**POST /register**

Mendaftarkan pengguna baru.

|  |  |
| --- | --- |
| **Request:** | POST /register  Content-Type: application/json |
| **Body Request:** | {  "username": "testUser",  "password": "testPassword123",  "email": "test@example.com"  } |
| **Response:** | * **201 Created** - Jika registrasi berhasil.   {  "msg": "Registrasi berhasil!"  }   * **400 Bad Request** - Jika username atau email sudah terdaftar.   {  "msg": "Username sudah terdaftar"  } |

**POST /login**

Login dan mendapatkan token JWT.

|  |  |
| --- | --- |
| **Request:** | POST /login  Content-Type: application/json |
| **Body Request:** | {  "username": "testUser",  "password": "testPassword123"  } |
| **Response:** | * **200 OK** - Jika login berhasil, Anda akan menerima token.   {  "token": "<JWT\_Token>"  }   * **401 Unauthorized** - Jika kredensial salah.   {  "msg": "Invalid credentials"  } |

**GET /auth**

Mengambil informasi autentikasi pengguna yang sudah login.

|  |  |
| --- | --- |
| **Request:** | GET /auth  Authorization: Bearer <your\_token> |
| **Response:** | * **200 OK** - Mengembalikan informasi pengguna.   {  "user": {  "id": 1,  "username": "testUser",  "email": "test@example.com"  }  }   * **401 Unauthorized** - Jika token tidak valid atau tidak ada.   {  "msg": "No token provided"  } |
|  |  |

**DELETE /logout**

Logout dan menghapus sesi.

|  |  |
| --- | --- |
| **Request:** | DELETE /logout  Authorization: Bearer <your\_token> |
| **Response:** | * **200 OK** - Jika logout berhasil.   {  "msg": "Logout successful"  } |

**2. Game Routes (Protected)**

**GET /games**

Mendapatkan daftar semua game.

|  |  |
| --- | --- |
| **Request:** | GET /games  Authorization: Bearer <your\_token> |
| **Response:** | * **200 OK** - Daftar game.   {  "games": [  {  "id": 1,  "judul": "Game 1",  "harga": 100,  "tahun\_rilis": 2023  },  {  "id": 2,  "judul": "Game 2",  "harga": 150,  "tahun\_rilis": 2024  }  ]  } |

**GET /games/:id**

Mendapatkan detail game berdasarkan ID.

|  |  |
| --- | --- |
| **Request:** | GET /games/:id  Authorization: Bearer <your\_token> |
| **Response:** | * **200 OK** - Detail game.   {  "game": {  "id": 1,  "judul": "Game 1",  "harga": 100,  "tahun\_rilis": 2023,  "deskripsi": "This is a description of Game 1."  }  }   * **404 Not Found** - Jika game dengan ID tidak ditemukan.   {  "msg": "Game not found"  } |

**POST /games**

Menambahkan game baru.

|  |  |
| --- | --- |
| **Request:** | POST /games  Authorization: Bearer <your\_token>  Content-Type: application/json |
| **Body Request:** | {  "judul": "New Game Title",  "deskripsi": "Description of the new game",  "harga": 100,  "tahun\_rilis": 2023,  "id\_genre": 1  } |
| **Response:** | * **201 Created** - Game berhasil dibuat.   {  "game": {  "id": 1,  "judul": "New Game Title",  "deskripsi": "Description of the new game",  "harga": 100,  "tahun\_rilis": 2023,  "id\_genre": 1  }  } |

**PATCH /games/:id**

Memperbarui game berdasarkan ID.

|  |  |
| --- | --- |
| **Request:** | PATCH /games/:id  Authorization: Bearer <your\_token>  Content-Type: application/json |
| **Body Request:** | {  "judul": "Updated Game Title"  } |
| **Response:** | * **200 OK** - Game berhasil diperbarui.   {  "game": {  "id": 1,  "judul": "Updated Game Title",  "deskripsi": "Updated description",  "harga": 100,  "tahun\_rilis": 2023  }  }   * **404 Not Found** - Jika game dengan ID tidak ditemukan.   {  "msg": "Game not found"  } |

**DELETE /games/:id**

Menghapus game berdasarkan ID.

|  |  |
| --- | --- |
| **Request:** | DELETE /games/:id  Authorization: Bearer <your\_token> |
| **Response:** | * **200 OK** - Game berhasil dihapus.   {  "message": "Game deleted successfully"  }   * **404 Not Found** - Jika game dengan ID tidak ditemukan.   {  "msg": "Game not found"  } |

**3. Genre Routes (Protected)**

**GET /genres**

Mendapatkan semua genre.

|  |  |
| --- | --- |
| **Request:** | GET /genres  Authorization: Bearer <your\_token> |
| **Response:** | * **200 OK** - Daftar genre.   {  "genres": [  {  "id": 1,  "nama\_genre": "Adventure"  },  {  "id": 2,  "nama\_genre": "Action"  }  ]  } |

**POST /genres**

Menambahkan genre baru.

|  |  |
| --- | --- |
| **Request:** | POST /genres  Authorization: Bearer <your\_token>  Content-Type: application/json |
| **Body Request:** | {  "nama\_genre": "Puzzle"  } |
| **Response:** | * **201 Created** - Genre berhasil ditambahkan.   {  "genre": {  "id": 3,  "nama\_genre": "Puzzle"  }  } |

**DELETE /genres/:id**

Menghapus genre berdasarkan ID.

|  |  |
| --- | --- |
| **Request:** | DELETE /genres/:id  Authorization: Bearer <your\_token> |
| **Response:** | * **200 OK** - Genre berhasil dihapus.   {  "message": "Genre deleted successfully"  } |

**4. Pembelian Routes (Protected)**

**GET /pembelian**

Mendapatkan semua pembelian.

|  |  |
| --- | --- |
| **Request:** | GET /pembelian  Authorization: Bearer <your\_token> |
| **Response:** | * **200 OK** - Daftar pembelian.   {  "pembelian": [  {  "id": 1,  "id\_game": 1,  "id\_user": 1,  "tanggal\_pembelian": "2023-01-01T00:00:00Z"  }  ]  } |

**POST /pembelian**

Menambahkan pembelian baru.

|  |  |
| --- | --- |
| **Request:** | POST /pembelian  Authorization: Bearer <your\_token>  Content-Type: application/json |
| **Body Request:** | {  "id\_game": 1,  "id\_user": 1,  "tanggal\_pembelian": "2023-01-01T00:00:00Z"  } |
| **Response:** | * **201 Created** - Pembelian berhasil ditambahkan.   {  "message": "Pembelian berhasil"  } |

**5. Review Routes (Protected)**

**GET /reviews**

Mendapatkan semua ulasan.

|  |  |
| --- | --- |
| **Request:** | GET /reviews  Authorization: Bearer <your\_token> |
| **Response:** | * **200 OK** - Daftar ulasan.   {  "reviews": [  {  "id": 1,  "id\_game": 1,  "id\_user": 1,  "komentar": "Great game!",  "rating": 5  }  ]  } |

**POST /reviews**

Menambahkan ulasan untuk game.

|  |  |
| --- | --- |
| **Request:** | POST /reviews  Authorization: Bearer <your\_token>  Content-Type: application/json |
| **Body Request:** | {  "id\_game": 1,  "id\_user": 1,  "komentar": "Great game!",  "rating": 5  } |
| **Response:** | * **201 Created** - Ulasan berhasil ditambahkan.   {  "review": {  "id": 1,  "komentar": "Great game!",  "rating": 5  }  } |