

Soal : Game Browser Online

Pendahuluan

Sebuah perusahaan yang baru didirikan sedang mencari pengembang tumpukan penuh untuk membuat platform game browser online. Pengembang game dapat mengunggah game mereka ke platform dan pengguna dapat memainkannya secara online di browser.

Ada tiga bagian pada platform:

- Portal Pengembang: Aplikasi web bagi pengembang game untuk mengunggah game mereka ke platform.
- Portal Administrator: Aplikasi web bagi administrator untuk mengelola platform dengan pengguna dan gamenya.
- Portal Permainan: Aplikasi web bagi pemain untuk bermain game online di browser.

Perusahaan ingin membuat produk minimum yang layak (MVP) untuk platform tersebut. MVP seharusnya sudah berisi bagian-bagian yang disebutkan di atas, tetapi dapat diterima bahwa Portal Pengembang Game dan Administrator.

Portal belum disempurnakan. Portal Gaming harus berfungsi penuh, sehingga pengguna dapat bermain game online di browser.

Deskripsi Proyek

Proyek ini dibagi menjadi dua fase:

- Fase pertama untuk membangun API dan halaman statis menggunakan kerangka kerja PHP dan database MySQL.
- Fase kedua untuk membangun bagian frontend menggunakan HTML/CSS dan kerangka kerja JavaScript, menggunakan API yang dikembangkan pada fase satu.

Anda dapat menemukan file media yang disediakan untuk mendukung pekerjaan Anda:

- Kerangka kerja yang disediakan (laravel, vuejs, reactjs)
- Postman Collection dan Environment
- Template GUI (untuk membangun UI frontend)
- lks-server.sql (database dengan struktur dan data dummy)

Fase 1 : API RESTful

Pada fase ini, Anda harus membangun RESTful API menggunakan framework Laravel sesuai dengan dokumentasi di bawah ini.

Pastikan pengguna dapat login menggunakan kredensial di bawah ini:

Administrator

Nama pengguna	Kata sandi
admin1	halouniverse1!
admin2	halouniverse2!

Pengembang

Nama pengguna	Kata sandi
dev1	halobyte1!
dev2	halobyte2!

Pemain

Nama pengguna	Kata sandi
pemain1	halodunia1!
pemain2	halodunia2!

Spesifikasi REST API

Informasi umum:

- Isi respons berisi beberapa contoh data statis. Data dinamis dari database harus digunakan.
- Parameter placeholder di URL ditandai dengan titik dua sebelumnya (misalnya :slug atau :id).
- Urutan properti dalam objek tidak masalah, tetapi urutan dalam array tidak masalah.
- Header Content-Type permintaan harus selalu application/json untuk POST, PUT, PATCH.
- Header Content-Type respons selalu application/json kecuali ditentukan lain.
- Tanda waktu diformat sebagai string ISO-8601. Contohnya: 2032-01-31T21:59:35.000Z.
- URL yang diberikan relatif terhadap URL dasar API. Misalnya /api/v1/games adalah URL untuk mendapatkan semua game.
- URL API tidak boleh diakhiri dengan .php atau .html atau ekstensi file lainnya. File game adalah pengecualian untuk ini.
- Token untuk titik akhir yang dilindungi harus ditentukan sebagai token Pembawa di header 'Otorisasi'. Yaitu 'Otorisasi: Pembawa <token>'

1. Otentikasi

Anda harus membuat titik akhir Login dan Logout. AccessToken harus dihasilkan oleh sanctum dan akan ditempatkan di header permintaan Pembawa Otorisasi.

Sign Up / mendaftar

POST [domain]/api/v1/auth/signup

Titik akhir ini membuat pengguna baru dan mengembalikan token sesi.

Request Body / Badan Permintaan:

```
{
  "nama pengguna": "pengguna pengujian",
  "kata sandi": "asdf1234"
}
```

HARTA BENDA	KOMENTAR
nama pengguna	Diperlukan, Unik, Panjang Min 4, Panjang Maks 60
kata sandi	Diperlukan, panjang min 5, panjang maks 10

Jawaban / Response:

Respons pembuatan yang berhasil:

Kode Status: 201

Badan Respons:

```
{
  "status": "berhasil",
  "token": "xxx"
}
```

Sign In / Masuk

POST [domain]/api/v1/auth/signin

Tindakan ini akan memeriksa nama pengguna dan sandi terhadap semua pengguna yang dikenal. Jika ditemukan, token sesi dikembalikan.

Tanggapan yang valid:

Kode Status: 200

Badan Permintaan:

```
{
  "nama pengguna": "pengguna pengujian",
  "kata sandi": "asdf1234"
}
```

HARTA BENDA	DESKRIPSI
nama pengguna	Diperlukan, Unik, Panjang Min 4, Panjang Maks 60
kata sandi	Diperlukan, panjang min 5, panjang maks 10

Badan Respons:

```
{
  "status": "berhasil",
  "token": "xxx"
}
```

Respons nama pengguna / kata sandi salah:

Kode Status: 401

Badan Respons:

```
{
  "status": "tidak valid",
  "message": "Nama pengguna atau sandi salah"
}
```

Keluar

POST [domain]/api/v1/auth/signout

Menghapus token sesi saat ini.

Tanggapan yang valid:

Kode Status: 200

Badan Respons:

```
{
  "status": "keberhasilan"
}
```

2. Pengguna

Mendapatkan semua data admin

DAPATKAN [domain]/api/v1/admin

Menampilkan data admin.

jawaban

Kode Status: 200

```
{
  "totalElemen": 2,

  "konten": [
    {
      "nama pengguna": "admin1",
      "last_login_at": "",
      "created_at": "2025-03-19 20:55:40",
      "updated_at": "2025-03-19 20:55:40",
    },
    {
```

```
"nama pengguna": "admin2",
"last_login_at": "2025-03-19 20:55:40",
"created_at": "2025-03-19 20:55:40",
"updated_at": "2025-03-19 20:55:40",
}
]
```

Pengguna bukan respons administrator:

Kode Status: 403

Badan Respons:

```
{
  "status": "dilarang",
  "message": "Anda bukan administrator"
}
```

POST [domain]/api/v1/pengguna

Titik akhir ini dapat digunakan untuk membuat pengguna.

Badan Permintaan:

```
{
  "nama pengguna": "pengguna pengujian",
  "kata sandi": "asdf1234"
}
```

HARTA BENDA	KOMENTAR
nama pengguna	Diperlukan, Unik, Panjang Min 4, Panjang Maks 60
kata sandi	Diperlukan, panjang min 5, panjang maks 10

Jawaban:

Respons pembuatan yang berhasil:

Kode Status: 201

Badan Respons:

```
{
  "status": "berhasil",
  "username": "pengguna uji"
}
```

Nama pengguna yang ada:

Jika nama pengguna tidak unik, pengguna admin tidak dapat dibuat dan sebagai gantinya respons berikut dikembalikan.

jawaban

Kode Status: 400

```
{
  "status": "tidak valid",
  "message": "Nama pengguna sudah ada"
}
```

Pengguna bukan respons administrator:

Kode Status: 403

Badan Respons:

```
{
  "status": "dilarang",
  "message": "Anda bukan administrator"
}
```

DAPATKAN [domain]/api/v1/pengguna

Mengembalikan detail pengguna.

jawaban

Kode Status: 200

```
{
  "totalElemen": 2,

  "konten": [
    {
      "nama pengguna": "pemain1",
      "last_login_at": "",
      "created_at": "2025-03-19 20:55:40",
      "updated_at": "2025-03-19 20:55:40",
    },
    {
      "nama pengguna": "pemain2",
      "last_login_at": "2025-03-19 20:55:40",
      "created_at": "2025-03-19 20:55:40",
      "updated_at": "2025-03-19 20:55:40",
    }
  ]
}
```

Pengguna bukan respons administrator:

Kode Status: 403

Badan Respons:

```
{
  "status": "dilarang",
  "message": "Anda bukan administrator"
}
```

```
}
```

PUT [domain]/api/v1/users/:id

Titik akhir ini dapat digunakan untuk memperbarui pengguna.

Badan Permintaan:

```
{  
  "nama pengguna": "pengguna pengujian",  
  "kata sandi": "asdf1234"  
}
```

HARTA BENDA	KOMENTAR
nama pengguna	Diperlukan, Unik, Panjang Min 4, Panjang Maks 60
kata sandi	Diperlukan, panjang min 5, panjang maks 10

Jawaban:

Respons pembuatan yang berhasil:

Kode Status: 201

Badan Respons:

```
{  
  "status": "berhasil",  
  "username": "pengguna uji"  
}
```

Nama pengguna yang ada:

Jika nama pengguna tidak unik, pengguna admin tidak dapat dibuat dan sebagai gantinya respons berikut dikembalikan.

jawaban

Kode Status: 400

```
{  
  "status": "tidak valid",  
  "message": "Nama pengguna sudah ada"  
}
```

Pengguna bukan respons administrator:

Kode Status: 403

Badan Respons:

```
{  
  "status": "dilarang",  
  "message": "Anda bukan administrator"
```

```
}
```

HAPUS [domain]/api/v1/users/:id

Titik akhir ini dapat digunakan untuk memperbarui pengguna.

Respons penghapusan yang berhasil:

Kode Status: 204

Ini mengembalikan isi kosong. Header 'Content-Type' tidak harus 'application/json'.

Pengguna tidak ditemukan tanggapan:

Kode Status: 403

Badan Respons:

```
{
  "status": "tidak ditemukan",
  "message": "Pengguna Tidak ditemukan"
}
```

Pengguna bukan respons administrator:

Kode Status: 403

Badan Respons:

```
{
  "status": "dilarang",
  "message": "Anda bukan administrator"
}
```

3. Permainan

DAPATKAN [domain]/api/v1/games

Mengembalikan daftar game yang dipaginasi halaman.

Parameter Kueri:

HARTA BENDA	DESKRIPSI	DEFAULT
halaman	Nomor halaman. Mulai dari 0	0
tingginya	Ukuran halaman. Harus lebih besar atau sama dengan 1	10
sortBy	Bidang untuk diurutkan. Harus salah satu dari "judul", "populer", "tanggal unggah"	titel
sortDir	Menjelaskan arah pengurutan. Harus salah satu dari "asc" atau "desc"	Asc

Bidang pengurutan dijelaskan di sini:

KEBUN	DESKRIPSI
titel	Judul permainan
populer	Menghitung jumlah total skor per game dan mengurutkan berdasarkan hitungan ini
tanggal unggah	Stempel waktu unggahan versi game terbaru

Di 'konten', kolom 'thumbnail' dan 'uploadTimestamp' hanya merujuk ke versi terbaru.

Bidang 'scoreCount' adalah jumlah skor di semua versi.

Jawaban:

Kode Status: 200

Badan Respons:

```
{
  "halaman": 0,
  "ukuran": 10,
  "totalElemen": 15,

  "konten": [
    {
      "siput": "demo-game-1",
      "title": "Permainan Demo 1",
      "description": "Ini adalah game demo 1",
      "thumbnail": "/games/:slug/:version/thumbnail.png",
      "unggah Stempel waktu": "2032-01-31T21:59:35.000Z",
      "penulis": "dev1",
      "scoreCount": 5
    }
  ]
}
```

Bidang halaman respons dijelaskan:

KEBUN	DESKRIPSI
halaman	Nomor halaman yang diminta. Mulai dari 0
tingginya	Ukuran halaman yang sebenarnya. Harus kurang atau sama dengan ukuran halaman yang diminta
Elemen total	Jumlah total elemen terlepas dari halaman
puas	Serangkaian permainan di halaman

Dapat dihitung berapa banyak halaman yang ada:

$\text{pageCount} = \text{ceil}(\text{totalElements} / \text{requestedSize})$

Ini juga dapat dihitung jika halaman yang dikembalikan adalah halaman terakhir dengan mengalikan (halaman+1) dengan ukuran halaman yang diminta dan memeriksa apakah hasilnya kurang dari atau sama dengan total elemen.

$\text{isLastPage} = (\text{halaman} + 1) * \text{requestedSize} \geq \text{totalElements}$

Catatan 1: Jika ada game yang belum memiliki versi game, itu tidak termasuk dalam respons atau jumlah total.

Catatan 2: Jika tidak ada thumbnail, bidang thumbnail adalah null.

POST [domain]/api/v1/games

Endpoint ini dapat digunakan untuk membuat game. Namun, versi game perlu diunggah dalam langkah terpisah. Jika game belum memiliki versi, game tersebut tidak dikembalikan di titik akhir ini.

Badan Permintaan:

```
{
  "title": "Permainan Demo 3",
  "description": "Ini adalah game demo 3"
}
```

HARTA BENDA	DESKRIPSI
titel	Diperlukan, panjang min 3, panjang maks 60
deskripsi	Diperlukan, panjang min 0, panjang maks 200

Jawaban:

Respons pembuatan yang berhasil:

Kode Status: 201

Badan Respons:

```
{
  "status": "berhasil",
  "slug": "siput game-yang dihasilkan"
}
```

Siput yang ada:

Jika siput yang dihasilkan tidak unik, game tidak dapat dibuat dan sebagai gantinya respons berikut dikembalikan.

jawaban

Kode Status: 400

```
{
  "status": "tidak valid",
  "slug": "Judul game sudah ada"
}
```

DAPATKAN [domain]/api/v1/games/:slug

Mengembalikan detail permainan.

jawaban

Kode Status: 200

```
{
  "siput": "demo-game-1",
  "title": "Permainan Demo 1",
  "description": "Ini adalah game demo 1",
  "thumbnail": "/games/:slug/:version/thumbnail.png",
  "unggah Stempel waktu": "2032-01-31T21:59:35.000Z",
  "penulis": "dev1",
  "skorHitungan": 5,
  "jalur permainan": "/game/demo-game-1/1/"
}
```

Jika tidak ada thumbnail, bidang thumbnail adalah null.

Kolom 'gamePath' menunjuk ke jalur URL yang dapat digunakan browser untuk merender game. Ini berarti ini adalah jalur aset yang dapat dijangkau.

Unggah file game

POST [domain]/api/v1/games/:slug/upload

Pengguna dapat mengunggah versi baru game jika mereka adalah penulis game tersebut.

- Ini bukan titik akhir REST dan lebih menerima unggahan file. Nama parameternya adalah 'zipfile'.
- Versi gim ini adalah bilangan bulat dan bertambah. Versi pertama adalah '1'. Pengguna tidak dapat mengontrol versi.
- Token sesi perlu disediakan sebagai parameter formulir 'token'.
- Jalur harus disimpan dalam catatan permainan, sehingga pemain dapat menemukan file game.

Jika upload gagal karena salah satu kemungkinan alasan ini, respons harus berupa penjelasan teks biasa tentang error tersebut.

- Pengguna bukan penulis game

Menyajikan file game

DAPATKAN /games/:slug/:version/

File game yang diunggah disajikan di bawah jalur yang bersifat publik.

PUT [domain]/api/v1/games/:slug

Endpoint ini memungkinkan penulis game untuk memperbarui judul dan deskripsi game.

Badan Permintaan:

```
{
  "title": "Demo Game 1 (diperbarui)",
  "description": "Deskripsi yang diperbarui"
}
```

Catatan: Ini tidak memperbarui siput game.

jawaban

Respons pembaruan yang berhasil:

Kode Status: 200

Badan Respons:

```
{
  "status": "keberhasilan"
}
```

Pengguna bukan tanggapan penulis game:

Kode Status: 403

Badan Respons:

```
{
  "status": "dilarang",
  "message": "Anda bukan penulis game"
}
```

HAPUS [domain]/api/v1/games/:slug

Penulis dapat menghapus game mereka. Ini menghapus permainan, semua versi dan semua skor.

jawaban

Respons penghapusan yang berhasil:

Kode Status: 204

Ini mengembalikan isi kosong. Header 'Content-Type' tidak harus 'application/json'.

Pengguna bukan tanggapan penulis game:

Kode Status: 403

Badan Respons:

```
{
  "status": "dilarang",
  "message": "Anda bukan penulis game"
}
```

DAPATKAN [domain]/api/v1/users/:username

Mengembalikan detail pengguna.

jawaban

Kode Status: 200

Badan Respons:

```
{
  "nama pengguna": "dev1",
  "registeredTimestamp": "2032-01-31T21:59:35.000Z",
  "authoredGames": [
    {
      "siput": "demo-game-1",
      "title": "Permainan Demo 1",
      "description": "Ini adalah game demo 1"
    }
  ],
  "skor tertinggi": [
    {
      "permainan": {
        "siput": "demo-game-1",
        "title": "Permainan Demo 1",
        "description": "Ini adalah game demo 1"
      },
      "skor": 15,
      "stempel waktu": "2032-01-31T21:59:35.000Z"
    }
  ]
}
```

authoredGames adalah array yang mengembalikan semua game dengan setidaknya satu versi di mana pengguna ini adalah penulisnya. Jika pengguna yang meminta detail pengguna adalah pengguna itu sendiri, ini juga mengembalikan game yang belum memiliki versi.

Skor tertinggi adalah susunan skor tertinggi per game yang dimainkan.

DAPATKAN [domain]/api/v1/games/:slug/scores

Mengembalikan skor tertinggi dari setiap pemain yang memainkan versi game apa pun, diurutkan berdasarkan skor (menurun).

jawaban

Kode Status: 200

Badan Respons:

```
{
  "skor": [
    {
      "nama pengguna": "pemain2",
      "skor": 20,
      "stempel waktu": "2032-01-31T21:59:35.000Z"
    },
    {
      "nama pengguna": "pemain1",
      "skor": 15,

```

```

    "stempel waktu": "2032-01-31T21:59:35.000Z"
  }
]
}

```

POST [domain]/api/v1/games/:slug/scores

Saat pengguna mengakhiri game berjalan, skor dapat diposting ke titik akhir ini.

Badan Permintaan:

```

{
  "skor": 100
}

```

Versi game yang terkait dengan skor adalah yang terbaru yang tersedia.

jawaban

Respons pembuatan yang berhasil:

Kode Status: 201

Badan Respons:

```

{
  "status": "keberhasilan"
}

```

Isi permintaan tidak valid

Jika permintaan POST atau PUT memiliki bidang yang tidak valid, mereka divalidasi dan respons dikembalikan yang mencantumkan pelanggaran.

Kode Status: 400

Badan Respons:

```

{
  "status": "tidak valid",
  "message": "Isi permintaan tidak valid.",
  "pelanggaran": {
    "field_name": {
      "message": "wajib"
    },
    "field_name": {
      "message": "harus memiliki panjang minimal 4 karakter"
    },
    "field_name": {
      "message": "harus paling banyak 60 karakter"
    }
  }
}

```

Dalam contoh di atas, semua kemungkinan pelanggaran ditampilkan. Pelanggaran yang dikembalikan sebenarnya hanya boleh berupa bidang yang sebenarnya tidak valid. Paling banyak satu validasi per bidang ditampilkan. Validasi dijalankan dalam urutan kemunculan pada contoh di atas. field_name harus diganti dengan nama bidang yang sebenarnya.

Pesan untuk panjang harus menyertakan nilai persyaratan panjang yang sebenarnya.

Header autentikasi tidak ada atau tidak valid

Jika konsumen melakukan panggilan ke jalur yang mengharuskan header auth hadir, ini harus menjadi respons:

Kode Status: 401

Badan Respons:

```
{
  "status": "tidak diautentikasi",
  "message": "Token hilang"
}
```

Jika konsumen melakukan panggilan ke jalur yang mengharuskan header autentikasi ada, tetapi memberikan token yang tidak valid atau tidak ada, ini harus menjadi respons:

Kode Status: 401

Badan Respons:

```
{
  "status": "tidak diautentikasi",
  "message": "Token tidak valid"
}
```

Jika konsumen melakukan panggilan ke jalur yang mengharuskan header auth hadir, tetapi pengguna diblokir, ini harus menjadi respons:

Kode Status: 403

Badan Respons:

```
{
  "status": "diblokir",
  "message": "Pengguna diblokir",
  "reason": "Anda telah diblokir oleh administrator"
}
```

Catatan: Alasannya adalah nilai dinamis, yang dipilih oleh admin yang memblokir pengguna.

Metode dan pola jalur berikut memerlukan header sesi yang valid untuk hadir:

- POST /api/v1/auth/signout
- POSTING, PUT, HAPUS /api/v1/games/**
- DAPATKAN, POSTING, PUT, HAPUS /api/v1/users/**
- DAPATKAN /api/v1/admins/**

Jalur API yang tidak ada

Jika konsumen melakukan panggilan ke jalur yang tidak ada, atau sumber daya yang tidak ada, ini harus menjadi respons:

Kode Status: 404

Badan Respons:

```
{  
  "status": "tidak ditemukan",  
  
  "message": "Tidak ditemukan"  
}
```

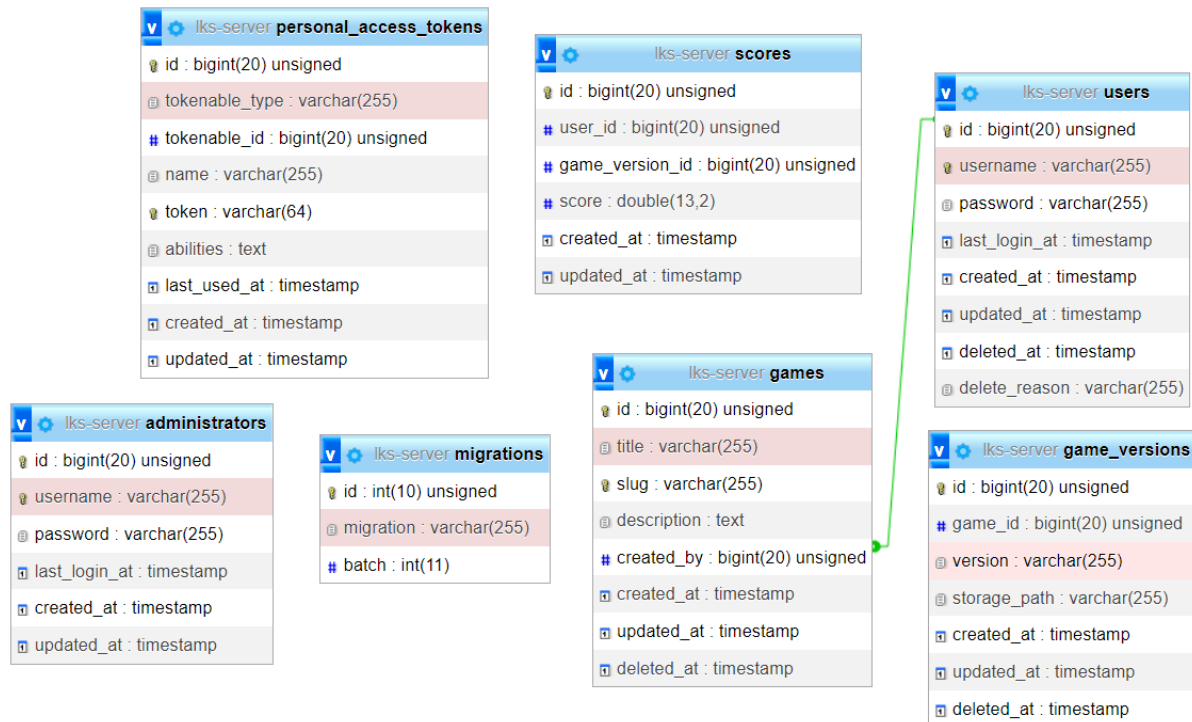

Tahap 2 : Pengembangan Front-end

Di bagian ini, Anda harus membangun aplikasi front-end menggunakan salah satu kerangka kerja yang disediakan (vue js atau react js). Anda dapat menggunakan gui template yang disediakan pada file media untuk membangun ui front-end.

Halaman / Fitur	Deskripsi
Sign Up, Sign In, Sign Out	<ul style="list-style-type: none"> - Pengguna dapat masuk menggunakan kredensial yang benar (nama pengguna dan kata sandi) - Kesalahan peringatan harus ditampilkan saat login gagal - Pengguna dapat keluar dengan mengklik tombol keluar di navbar. - Pengguna 2 level dapat login (administrator dan pengguna) - Menu Administrator (Daftar Admin, Daftar Pengguna) - Menu pengguna (Temukan Game, Kelola Game, Profil Pengguna)
Home	<ul style="list-style-type: none"> - Pengguna dapat melihat info login mereka.
Daftar Admin	<ul style="list-style-type: none"> - Admin dapat melihat semua pengguna admin dalam database dengan nama pengguna, stempel waktu yang dibuat, dan stempel waktu login terakhir.
Daftar Pengguna	<ul style="list-style-type: none"> - Admin dapat melihat semua pengguna platform dengan nama pengguna, stempel waktu pendaftaran, stempel waktu login terakhir dan mengklik tautan ke halaman profil mereka - Admin dapat memblokir dan membuka blokir pengguna - Admin dapat membuat, memperbarui, dan menghapus pengguna
Temukan game	<ul style="list-style-type: none"> - Lihat daftar game. Untuk setiap permainan, berikut ini ditampilkan: <ul style="list-style-type: none"> o Titel o Jumlah Skor: Jumlah skor yang dikirimkan o Deskripsi: Deskripsi yang diberikan oleh penulis. o Thumbnail: Jika tersedia, jika tidak, grafik default. o Daftar dapat diurutkan berdasarkan popularitas, yang baru diperbarui, atau judul. Opsi naik dan turun dapat dipilih oleh pengguna. - Tautan ke game harus memiliki tujuan tautan khusus tautan yang diterapkan untuk aksesibilitas <p>Permainan ditampilkan dalam daftar gulir tanpa batas. Dengan daftar gulir tak terbatas, tidak ada halaman, tetapi menggulir halaman memicu pemuatan lebih banyak game jika posisi gulir mendekati akhir game yang saat ini dimuat. Dalam mode gulir tak terbatas, tidak ada halaman, tetapi pengguliran memicu pemuatan lebih banyak game.</p>

Detail Permainan	<ul style="list-style-type: none"> - Halaman ini merender game menunjukkan skor tertinggi saat ini (yang diperbarui secara otomatis). - Jika pengguna berada dalam sepuluh besar, pengguna saat ini disorot dan ditandai sebagai pengguna saat ini (misalnya dengan menunjukkannya tebal). Jika pengguna saat ini memiliki skor yang berperingkat di bawah sepuluh besar, maka skor tersebut ditambahkan, tetapi ditampilkan tanpa peringkat. - Deskripsi game juga ditampilkan.
Profil Pengguna	<p>Halaman ini menampilkan profil pengguna dengan:</p> <ul style="list-style-type: none"> - Nama pengguna - Game yang Ditulis <ul style="list-style-type: none"> o Daftar diurutkan berdasarkan kapan versi terakhir diunggah. Game yang terakhir diperbarui ada di bagian atas. o Jika pengguna belum mengunggah game apa pun, bagian "Game yang Ditulis" dihilangkan. o Per game yang ditulis, berikut ini ditampilkan: Judul Game, Deskripsi, Thumbnail, Jumlah skor yang dikirimkan - Penulis tidak ditampilkan per game karena ini adalah informasi yang berlebihan. - Skor tertinggi <ul style="list-style-type: none"> o Daftar tersebut menunjukkan skor tertinggi per game. o Daftar ini diurutkan berdasarkan judul game menurut abjad. o Setiap game memiliki tautan yang dapat diklik untuk masuk ke game.
Kelola Game	<p>Di halaman ini, developer dapat memperbarui judul dan deskripsi game, menghapus game, atau mengunggah versi baru. Di halaman ini, pengembang dapat melakukan hal berikut:</p> <ul style="list-style-type: none"> - Perbarui judul dan deskripsi. - Unggah versi baru. Ini mengharuskan pengguna untuk mengunggah file ZIP. - Hapus game. Game hanya dihapus setelah pengguna mengonfirmasi. <p>Jika pengguna mencoba mencapai halaman ini tanpa menjadi penulisnya, mereka akan diarahkan ke game itu sendiri.</p>

ERD



Peraturan

- Simpan REST Api Anda di direktori bernama **"SERVER_MODULE/backend"** dan simpan di folder repositori.
- Simpan aplikasi front-end Anda di direktori bernama **"SERVER_MODULE/frontend"** dan simpan di folder repositori.
- Saat mengembangkan Frontend, Anda dapat **menggunakan/menggunakan REST API yang disediakan**.
- Anda harus **membangun aplikasi frontend ke mode produksi** dan **menyimpan file produksi di folder "SERVER_MODULE/frontend/build"** sebelum Anda mendorongnya ke repositori github.
- Pastikan **folder "/build" tidak tertulis di file .gitgnor**.
- Anda harus melakukan dan mendorong perubahan Anda ke repositori github setidaknya setiap 30 menit.

