

Ujian Tengah Semester

Backend Programming – Semester Genap 2023/2024

Deskripsi Umum

1. UTS ini dikerjakan secara **individu**. Jika ada pekerjaan yang dianggap plagiarisme, maka akan mendapatkan sanksi pengurangan nilai hingga nilai dianggap 0. Penilaian plagiarisme ini adalah kewenangan dosen dan tidak dapat diganggu gugat dengan alasan apapun.
2. Unduhlah *template project* yang telah diberikan melalui Microsoft Teams. Anda wajib menggunakan kode program pada *base project* ini sebagai template pengerjaan UTS ini.
3. Bahasa pemrograman yang wajib digunakan pada UTS ini adalah Javascript dan Node.js. Database yang digunakan adalah MongoDB. Anda dapat menggunakan MongoDB Atlas atau instalasi MongoDB lokal di PC anda masing-masing.
4. Kerjakan soal-soal di bawah ini dalam 1 project yang sama (bukan masing-masing soal menjadi project terpisah-pisah).

Soal

1. Pagination dan Filter

Endpoint seperti GET /user s digunakan untuk mendapatkan data semua user yang tersimpan di dalam database. Tentunya jumlah data user ini banyak dan tidak efektif jika langsung menampilkan seluruhnya.

Pada umumnya, endpoint untuk mendapatkan seluruh data ini memiliki fitur pagination yang dapat digunakan untuk membatasi jumlah data yang dihasilkan dengan cara mencantumkan jumlah data per halaman, nomor halaman, pengurutan data, dan filter atau pencarian data berdasarkan atribut tertentu.

Misalkan pada request ke `GET /users?page_number=2&page_number=10`, maka yang ditampilkan adalah daftar user pada halaman ke-2 dan dibatasi 10 user per halaman.

Contoh lainnya, request ke

`GET /users?page_number=1&page_number=5&search=email:test&sort=email:desc` akan menghasilkan daftar user yang:

- Mengandung substring “*test*” pada email
- Urutkan hasil berdasarkan email *descending*
- Setiap halaman hanya ada 5 user
- Tampilkan daftar user pada halaman pertama

Perhatikan urutan pengerjaan seperti pada contoh terakhir, yaitu filter `search` harus dilakukan terlebih dahulu, kemudian `sort`, baru dilakukan pagination menggunakan `page_size` dan `page_number`.

Penjelasan mengenai query masing-masing adalah sebagai berikut:

Request Query	
<code>page_number</code>	Harus bertipe integer (bilangan positif). Jika tidak diisi, maka default adalah menampilkan semua halaman/semua data.
<code>page_size</code>	Harus bertipe integer (bilangan positif). Jika tidak diisi, maka default adalah menampilkan semua data dalam 1 halaman.
<code>sort</code>	Memiliki format <code><field name>:<sort order></code> . Field name adalah field pada database user yang dapat diurutkan, yaitu <code>email</code> atau <code>name</code> . Field name ini tidak akan mengandung tanda titik dua.

	<p>Sort order yang dapat diisi adalah asc atau desc.</p> <p>Jika tidak diisi sama sekali atau format salah, maka dianggap sort berdasarkan email (<i>ascending</i>).</p>
search	<p>Memiliki format <field name>:<search key>.</p> <p>Field name adalah field pada database user yang dapat dicari, yaitu email atau name. Field name ini tidak akan mengandung tanda titik dua.</p> <p>Search key adalah sebuah string yang dapat mengandung huruf besar, huruf kecil, angka, maupun simbol.</p> <p>Jika tidak diisi sama sekali atau format salah, maka dianggap tidak ada.</p>

Response dari GET /users juga berubah. Jika sebelumnya berbentuk seperti contoh di bawah ini:

```
[{
  "id": "329710120d293",
  "name": "User 1",
  "email": "user01@gmail.com"
}, {
  "id": "293c420923af3",
  "name": "User 2",
  "email": "user02@gmail.com"
}, {
  "id": "83928afed2910",
  "name": "User 3",
  "email": "user03@gmail.com"
}]
```

berubah menjadi seperti contoh di bawah ini (dengan pagination):

```
{
  "page_number": 1,
  "page_size": 10,
  "count": 3,
  "total_pages": 1,
  "has_previous_page": false,
  "has_next_page": false,
  "data": [{
    "id": "329710120d293",
    "name": "User 1",
    "email": "user01@gmail.com"
  }, {
    "id": "293c420923af3",
    "name": "User 2",
    "email": "user02@gmail.com"
  }, {
    "id": "83928afed2910",
    "name": "User 3",
    "email": "user03@gmail.com"
  }]
}
```

Berikut penjelasan dari setiap atribut response di atas:

Response	
page_number	Nomor halaman yang ditampilkan pada response ini (integer)
page_size	Jumlah data yang dimunculkan per halaman (integer)
count	Jumlah total keseluruhan data (total seluruh halaman; integer)
total_pages	Jumlah total halaman yang ada (integer)
has_previous_page	Menunjukkan apakah ada halaman sebelumnya (boolean)
has_next_page	Menunjukkan apakah ada halaman selanjutnya (boolean)
data	Data yang ditampilkan (array of objects)

2. Login Attempts Limit

Implementasi batasan limit kesalahan pada saat login, sehingga jika user gagal melakukan login (karena kesalahan email atau password) sebanyak 5 kali, maka API akan mengembalikan error 403 Forbidden: Too many failed login attempts.

Batasan limit di atas dikenakan terhadap email user tersebut. Limit kesalahan login di atas akan hilang setelah jangka waktu 30 menit. Counter limit juga akan hilang pada saat user berhasil login.

Misalnya, perhatikan contoh kasus berikut ini:

- [2024-01-01 08:00:03] User test@example.com gagal login. Attempt = 1.
- [2024-01-01 08:00:35] User test@example.com gagal login. Attempt = 2.
- [2024-01-01 08:01:04] User test@example.com gagal login. Attempt = 3.
- [2024-01-01 08:03:11] User test@example.com berhasil login.
- [2024-01-01 08:10:15] User test@example.com logout.
- [2024-01-01 08:25:44] User test@example.com gagal login. Attempt = 1.
- [2024-01-01 08:26:14] User test@example.com gagal login. Attempt = 2.
- [2024-01-01 08:26:56] User test@example.com gagal login. Attempt = 3.
- [2024-01-01 08:27:04] User test@example.com gagal login. Attempt = 4.
- [2024-01-01 08:27:30] User test@example.com gagal login. Attempt = 5. Limit reached.
- [2024-01-01 08:28:00] User test@example.com mencoba login, namun mendapat error 403 karena telah melebihi limit attempt.
- [2024-01-01 08:58:00] User test@example.com bisa mencoba login kembali karena sudah lebih dari 30 menit sejak pengenaan limit. Attempt di-reset kembali ke 0.

3. Jika sistem yang dibuat adalah sebuah API untuk aplikasi di bidang:

- a. Online atau digital banking → untuk mahasiswa dengan **NIM genap**
- b. E-commerce atau marketplace → untuk mahasiswa dengan **NIM ganjil**

Tambahkan minimal masing-masing 1 endpoint untuk operasi *create*, *read*, *update*, dan *delete* (CRUD)!

Fitur apa yang mau dibuat diserahkan kepada mahasiswa, yang penting sesuai dengan bidang tersebut.

Penjelasan:

- Sebagai contoh, jika yang diminta adalah untuk sebuah aplikasi kasir, endpoint yang bisa dibuat:
 - *Create* → POST /purchases
Ada customer yang membeli barang dan diinput oleh kasir ke sistem, kemudian endpoint ini dipanggil untuk menyimpan pembelian tersebut ke database.
 - *Read* → GET /products
Mendapatkan daftar barang yang dijual, berisi nama produk, deskripsi, harga, *quantity* di gudang, dan sebagainya.
 - *Update* → PUT /purchases/{id}
Misalkan terjadi kesalahan input pembelian, maka manajer toko dapat mengubah pembelian tersebut, misalkan mengubah barang yang dibeli atau jumlahnya, dan sebagainya.
 - *Delete* → DELETE /purchases/{id}
Misalkan terjadi kesalahan input pembelian, maka manajer toko dapat menghapus pembelian tersebut dari sistem.
- Untuk input dari fitur tersebut, tentukan secara rasional dengan mengira-ngira kebutuhan user yang menggunakan aplikasi ini.
- Anda bisa juga mencari contoh kasus apps yang benar-benar ada, kemudian mengambil fitur-fitur yang ada untuk dibuat.

Ketentuan Pengumpulan

1. Pengumpulan dilakukan melalui Microsoft Teams pada assignment UTS yang telah dibuat.
2. Perhatikan batas tanggal dan jam pengumpulan yang tercantum pada assignment. Jika melewati batas waktu, maka pengumpulan anda tidak diterima dan tidak ada perpanjangan waktu dengan alasan apapun.

3. Mahasiswa tetap harus hadir tepat waktu di kampus pada tanggal dan jam UTS sesuai jadwal yang diberikan Program Studi untuk menandatangani daftar hadir UTS. Jika tidak ditandatangani, maka anda tetap tidak dianggap mengikuti UTS.
4. Pengumpulan yang diunggah ke assignment di Microsoft Teams berupa:
 - a. Link ke repository pada Github yang berisi source code.
 - b. File laporan dengan format PDF yang berisi bukti pengerjaan tiap soal berupa screenshot Bruno.
5. Pastikan link repository di Github tersebut bisa diakses setidaknya hingga semester ini berakhir dan nilai akhir anda telah muncul di Lintang.
6. Pengumpulan yang tidak sesuai dengan ketentuan tidak akan dinilai.

Kriteria Penilaian

- **40%** Program
- **20%** Fungsi
- **15%** Struktur program
- **15%** Material yang dicakup
- **5%** Presentasi
- **5%** Kompetensi Individu