



Institut Sains dan Teknologi Terpadu Surabaya

Jl. Ngagel Jaya Tengah 73 - 77, Surabaya 60284

Telp. (031) 5027920 Fax. (031) 5041509



Laboratorium : L-404

Waktu : Kamis / 10.45-12.45

Minggu Ke : 3

Materi : Object-Relational Mapping

Praktikum : Web Service

Jurusan : S1 – Informatika

Tanggal : 9 Maret 2023

Jenis Soal : Materi dan Tugas

MATERI (TOTAL = 40)

Wajib berikan response code: **400** jika bad request, **404** jika not found, **200** jika sukses, **201** bila created atau berhasil menambahkan data baru. Pastikan semua endpoint yang ada **HARUS** menggunakan Object-Relational Mapping dan **BUKAN** raw query.

Buatlah sebuah web service Portal Berita Indonesia. Endpoint yang harus dibuat adalah sebagai berikut:

1. [POST] /api/users/

Endpoint ini berguna untuk mendaftarkan user ke dalam database menggunakan Model.

- Data yang diperlukan adalah username, email, dan tanggal_lahir.
- Beri pengecekan field harus ada dan tidak kosong.

Contoh Request	Contoh Response
<pre>body: - username: "odiodi123" - email: "ignatiusodi@gmail.com" - tanggal_lahir: "06/06/2001"</pre>	<pre>status: 201 body: { username: "odiodi123", email: "ignatiusodi@gmail.com", tanggal_lahir: "06/06/2001", tanggal_join: "09/03/2023" }</pre>
<pre>body: - email: "" - tanggal_lahir: "06/06/2001"</pre>	<pre>status: 400 body: { message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

2. [GET] /api/users

Endpoint ini berguna untuk mendapatkan semua user yang terdaftar. Tampilkan data user beserta dengan jumlah berita yang sudah ia publish.

Contoh Request	Contoh Response
----------------	-----------------

body:	status: 200
-	body: [
{
username: "odiodi123",
tanggal_lahir: "06/06/2001",
jumlah_bertia: "12"
},
{
username: "hartononickolas",
tanggal_lahir: "26/10/2002",
jumlah_bertia: "0"
},

]

3. [POST] /api/berita

Endpoint ini berguna untuk membuat berita baru yang dibuat oleh user yang sudah terdaftar. Masukkan data ke dalam database menggunakan model.

- Data yang diperlukan adalah headline, konten, kategori berita, dan user yang membuat.
- Apabila berhasil, buatkan id berita dengan format "NEWS-" + 3 digit nomor urut serta tambahkan jumlah berita milik user yang mengupload sebanyak 1.
- Sebuah berita nantinya bisa memiliki lebih dari 1 komentar.
- Beri pengecekan user yang membuat harus ada dan semua field ada dan terisi.

Contoh Request	Contoh Response
body: - headline: "Panas, kontroversi bantengdengan segitiga" - content: "Dikabarkan pada dini hariseekor banteng sedang bertarungdengan segitiga besi" - kategori: "Politik" - user_pembuat: "bintangtujuh6"	status: 201 body: {id_berita: "NEWS-001",headline: "Panas, kontroversi banteng dengan segitiga",content: "Dikabarkan pada dini hari seekor bantengsedang bertarung dengan segitiga besi",kategori: "Politik",user_pembuat: "bintangtujuh6",created_at: "09/03/2023" }
body: - headline: "" - content: "" - kategori: "Keuangan" - user_pembuat: "abc"	status: 400 body: {message: "Terdapat field yang kosong/format salah/tidak ada!" }

4. [GET] /api/berita/:id

Endpoint ini berguna untuk mendapatkan data berita yang dimaksud dari parameter id. Tampilkan satu spesifik berita dengan data yang ditampilkan dapat di lihat di bawah. Apabila berita tidak ditemukan, berikan pesan tidak ada.

Contoh Request	Contoh Response
----------------	-----------------

<pre>param: - id: "NEWS-002"</pre>	<pre>status: 200 body: { id_berita: "NEWS-002", headline: "Dahsyatnya Dampak Ekonomi Gempa Turki dan Suriah, Kerugian Sentuh Rp 60,4 Triliun", content: "Sebanyak lebih dari 17.500 korban tewas dalam gempa berkekuatan 7,8 yang melanda Turki dan Suriah pada Senin (6/2/2023)", kategori: "Ekonomi", user_pembuat: "natasha223", created_at: "10/02/2023", komentar: [{ user: "bintangtujuh6", konten: "Wah kasian sekali mereka :(" }, { user: "hartononickolas", konten: "Semoga mereka diberikan ketabahan" }] }</pre>
<pre>param: - id: "NEWS12"</pre>	<pre>status: 404 body: { message: "Berita tidak ditemukan!" }</pre>

5. [GET] /api/berita?kategori=

Endpoint ini berguna untuk mendapatkan data berita yang dimaksud dari parameter kategori yang sifatnya adalah like dan case insensitive.

- Tampilkan format data yang sama seperti pada nomor 4.
- Beri pengecekan kalau kategori yang dicari tidak ditemukan

Contoh Request	Contoh Response
<pre>param: - kategori: "oli"</pre>	<pre>status: 200 body: [{ id_berita: "NEWS-002", ... kategori: "Politik", ... }, { id_berita: "NEWS-004", ... kategori: "Poligami", ... }, { id_berita: "NEWS-007", ... kategori: "Politik", ... }]</pre> <p>*Note: titik-titik hanya untuk mempersingkat penulisan soal</p>
<pre>param: - kategori: "zxa"</pre>	<pre>status: 404 body: { message: "Berita tidak ditemukan!" }</pre>

6. [POST] /api/berita/comment

Endpoint ini berguna untuk melakukan comment ke sebuah berita yang mana akan masuk ke database menggunakan model.

- Data yang diminta adalah id_berita, user_komentar, dan isi_komentar
- Apabila berhasil, tambahkan komentar ke berita tersebut

- User yang mengupload berita bisa melakukan komentar di beritanya sendiri
- Beri pengecekan id berita dan user komentar terdaftar di database, dan semua field ada dan terisi.
- Nomor ini akan dinilai apabila komentar benar-benar masuk saat dilihat di endpoint No 4

Contoh Request	Contoh Response
<pre>body: - id_berita: "NEWS-001" - user_komentar: "odiodi123" - isi_komentar: "Tak kusangka ada ... kejadian seperti itu, menyedihkan ... sekali teman-teman😭"</pre>	<pre>status: 201 body: { message: "Berhasil menambahkan komentar!" }</pre>
<pre>body: - id_berita: "NIUS21" - user_komentar: "" - isi_komentar: "😭"</pre>	<pre>status: 400 body: { message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

7. [DELETE] /api/berita

Endpoint ini berguna untuk melakukan delete berita yang sudah pernah tayang.

- Data yang dibutuhkan adalah id berita, user yang menghapus, dan alasan menghapus
- Beri pengecekan apabila user pembuat berita = user yang menghapus. Apabila tidak maka batalkan
- Pastikan data terhapus dari database (hapus juga commentnya).

Contoh Request	Contoh Response
<pre>body: - id_berita: "NEWS-001" - user_penghapus: "bintangtujuh6"</pre>	<pre>status: 200 body: { message: "Berita dan komentarnya berhasil dihapus!" }</pre>
<pre>body: - id_berita: "NEWS-002" - user_penghapus: "hartononickolas" <p>*pembuat NEWS-002 = natasha223</p></pre>	<pre>status: 404 body: { message: "Berita tidak ditemukan!" }</pre>

Struktur Database Minggu 3 (PRAK_M3_WS):

Users	Keterangan
username	varchar, PK
email	varchar, NN
tanggal_lahir	varchar, NN
jumlah_berita	int, NN
tanggal_join	date, NN

Berita	Keterangan
id	varchar, PK
headline	varchar, NN
content	varchar, NN
kategori	varchar, NN
user_pembuat	varchar, NN, FK Users(username)
created_at	date, NN

Komentar	Keterangan
id	int, auto inc, PK
berita_id	varchar, NN
username	varchar, NN
konten	varchar, NN

**DILARANG MENGGUNAKAN MATERI YANG BELUM DIAJARKAN PADA MINGGU INI
JIKA MELANGGAR MAKA NILAI MATERI : 0**

PERHATIKAN KETENTUAN DI BAWAH:

- Nama Endpoint dan Nama Key untuk request HARUS SAMA dengan ketentuan SOAL
- Dilarang mengganti struktur DB yang diberikan. Melanggar = Materi 0
- Wajib menyertakan response code yang sesuai, jika tidak disertakan / sesuai maka kriteria yang berhubungan tidak akan dinilai
- Apabila data tidak tersimpan maka nomor bersangkutan tidak akan diperiksa
- Highlight kriteria yang dikerjakan dengan warna kuning dan kumpulkan word beserta dengan file materi, apabila tidak dikumpulkan maka materi tidak akan diperiksa.
- Akan ada pengurangan nilai sebesar -5 untuk setiap kriteria yang dihighlight namun tidak dikerjakan.

• **MENCONTEK = Nilai MOD 2**

MATERI : 40

SCORE	KRITERIA
0/2/4/6	[POST] /api/users/ Berjalan dengan sesuai, pengecekan sesuai
0/2/4	[GET] /api/users Berjalan dengan sesuai, data diminta sesuai
0/2/4/6	[POST] /api/berita - Berjalan dengan sesuai - Format id benar - Pengecekan sesuai - Jumlah post user bertambah
0/2/4/6	[GET] /api/berita:id - Berjalan dengan sesuai - Data diminta sesuai (beserta komentar) - Pengecekan sesuai
0/2/4/6	[GET] /api/berita?kategori= - Berjalan dengan sesuai - Data diminta sesuai (beserta komentar) - Pengecekan sesuai
0/2/4/6	[POST] /api/berita/comment - Berjalan dengan sesuai - Komentar bertambah - Pengecekan sesuai
0/2/4/6	[DELETE] /api/berita - Berjalan dengan sesuai - Berita terhapus (berita dan komentar) - Pengecekan sesuai
-40/0	Tidak menggunakan model
Total : 40	

TUGAS (TOTAL = 30)

Wajib berikan response code: **400** jika bad request, **404** jika not found, **200** jika sukses, **201** bila created atau berhasil menambahkan data baru. Pastikan semua endpoint yang ada **HARUS** menggunakan Object-Relational Mapping dan **BUKAN** raw query. Rancanglah databasenya sendiri.

Buatlah sebuah web service Event Planning Online seperti Eventbrite. Endpoint yang harus dibuat adalah sebagai berikut:

1. [POST] /api/user

Endpoint ini berguna untuk membuat user baru. Data yang diminta untuk membuat user baru adalah email, nama, jenis kelamin (P/L), nomor telepon, dan password. Generatekan id user dengan format “EX” + 5 digit nomor urut. Berikan pengecekan sebuah email hanya dapat membuat 1 akun saja dan data harus ada dan ada isinya!

Contoh Request	Contoh Response
<pre>body: - email: "richbrian@gmail.com" - nama: "Brian Imanuel" - jenis_kelamin: "P" - nomor_HP: "08144556677" - password: "cobalagi"</pre>	<pre>status: 201 body: { id: "EX00001", email: "richbrian@gmail.com", nama: "Brian Imanuel", jenis_kelamin: "P", nomor_HP: "08144556677" }</pre>
<pre>body: - id_berita: "NIUS21" - user_komentar: "" - isi_komentar: "☹"</pre>	<pre>status: 400 body: { message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

2. [POST] /api/event

Endpoint ini berguna untuk membuat sebuah event baru oleh seorang user. Seorang user bisa membuat satu atau lebih event dan seorang user bisa register ke satu atau lebih event.

- Data yang diminta untuk membuat event baru adalah id user organizer, nama, deskripsi, lokasi, tanggal.
- Generate-kan format event dengan format “EV” + 5 digit nomor urut.
- Beri pengecekan tanggal event harus lebih besar dari hari ini dan memiliki format (DD/MM/YYYY).
- Beri pengecekan id user organizer harus sudah terdaftar juga.

Contoh Request	Contoh Response
<pre>body: - id_user: "EX00003" - nama: "Seminar KSP Episode 1" - deskripsi: "The Future of Digital Art" - lokasi: "Kampus ISTTS N-110, Surabaya" - tanggal: "10/03/2023"</pre>	<pre>status: 201 body: { id_event: "EV00001", id_user: "EX00003", nama: "Seminar KSP Episode 1", deskripsi: "The Future of Digital Art", lokasi: "Kampus ISTTS N-110, Surabaya", tanggal: "10/03/2023", peserta: [], created_at: "09/03/2023" }</pre>
<pre>body: - id_user: "EB0021" - nama: "" - deskripsi: "The Future of Digital Art" - lokasi: "Kampus ISTTS N-110, Surabaya" - tanggal: "02/01/2023"</pre>	<pre>status: 400 body: { message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

3. [PUT] /api/event/:id

Endpoint ini berguna untuk melakukan update event sesuai id yang diberikan pada parameter.

- Data yang dapat diupdate adalah nama, deskripsi, lokasi, dan tanggal.
- Field yang diupdate boleh hanya 1,2,3,4 data
- Beri pengecekan yang sama seperti nomor 2. Apabila berhasil, maka updatelah data event yang ada di database.

Contoh Request	Contoh Response
<pre>param: - id: "EV00001" body: - nama: "Ini contoh update yaa" - deskripsi: "Deksripsi terupdate"</pre>	<pre>status: 201 body: { id_event: "EV00001", id_user: "EX00003", nama: "Ini contoh update yaa", deskripsi: "Deksripsi terupdate", lokasi: "Kampus ISTTS N-110, Surabaya", tanggal: "10/03/2023", peserta: [], created_at: "09/03/2023" }</pre>
<pre>param: - id: "EV00001" body: - id_user: "EX00001" - nama: "" - deskripsi: "Deksripsi terupdate"</pre>	<pre>status: 400 body: { message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

4. [GET] /api/event/all

Endpoint ini berguna untuk melihat semua event yang sudah terdaftar dan masih aktif (tanggal eventnya >= hari ini). Tampilkan data yang dibutuhkan sesuai dengan di contoh bawah. Apabila tidak ada event terdaftar maka tampilkan seperti contoh di bawah.

Contoh Request	Contoh Response
----------------	-----------------

body: -	<pre> status: 200 body: [{ id_event: "EV00001", id_user: "EX00003", nama: "Seminar KSP Episode 1", deskripsi: "The Future of Digital Art", lokasi: "Kampus ISTTS N-110, Surabaya", tanggal: "10/03/2023", peserta: [], created_at: "09/03/2023" }, { id_event: "EV00002", id_user: "EX00006", nama: "Parade Bunga Surabaya 2023", deskripsi: "Building Culture in Youth", lokasi: "Alun-Alun Surabaya", tanggal: "16/05/2023", peserta: [{ id_user: "EX00001", nama: "Brian Imanuel" }, { id_user: "EX00002", nama: "Lawrence Patrick" }], created_at: "08/03/2023" }] </pre>
body: -	<pre> status: 404 body: { message: "Tidak ada event!" } </pre>

5. [GET] /api/event?nama=&lokasi=

Endpoint ini berguna untuk mencari nama event yang dicari berdasarkan nama dan lokasi yang bersifat like dan case insensitive. Tampilkan response seperti pada nomor 3.

Contoh Request	Contoh Response
----------------	-----------------

<pre> param: - nama: "Bunga" - lokasi: "Alun-Alun" </pre>	<pre> status: 200 body: [{ id_event: "EV00002", id_user: "EX00006", nama: "Parade Bunga Surabaya 2023", deskripsi: "Building Culture in Youth", lokasi: "Alun-Alun Surabaya", tanggal: "16/05/2023", peserta: [{ id_user: "EX00001", nama: "Brian Imanuel" }, { id_user: "EX00002", nama: "Lawrence Patrick" }], created_at: "08/03/2023" }] </pre>
<pre> param: - nama: "bungaaaaa" - lokasi: "rumaaaaah" </pre>	<pre> status: 404 body: { message: "Tidak ada event!" } </pre>

6. [POST] /api/event/register

Endpoint ini berguna untuk melakukan register terhadap suatu event. Data yang dibutuhkan adalah id user dan id event. Beri pengecekan user dan event terdaftar atau tidak.

Contoh Request	Contoh Response
<pre> body: - id_user: "EX00003" - id_event: "EV00001" </pre>	<pre> status: 201 body: { message: "User berhasil melakukan registrasi event" } </pre>
<pre> body: - id_user: "AA00003" - id_event: "BB00001" </pre>	<pre> status: 404 body: { message: "User/Event tidak terdaftar" } </pre>

7. [GET] /api/user/:id

Endpoint ini berguna untuk melihat data user yang dipilih dari parameter id yang diberikan. Tampilkan data yang diberikan seperti contoh di bawah. Beri pengecekan id user harus terdaftar

Contoh Request	Contoh Response
<pre>param: - id: "EX00003"</pre>	<pre>status: 200 body: { id: "EX00003", nama: "Ivan Susanto", tiket: [{ id_event: "EV00001", nama_event: "Seminar KSP Episode 1" }, { id_event: "EV00002", nama_event: "Parade Bunga Surabaya 2023" }] }</pre>
<pre>param: - id: "EX00005"</pre>	<pre>status: 200 body: { id: "EX00005", nama: "Williandy Takhta", tiket: [] }</pre>
<pre>param: - id: "EB00005"</pre>	<pre>status: 404 body: { message: "User tidak ditemukan!" }</pre>

**DILARANG MENGGUNAKAN KONSEP MAUPUN MATERI YANG BELUM DIAJARKAN
JIKA MELANGGAR MAKA NILAI TUGAS: 0**

PERHATIKAN KETENTUAN DI BAWAH:

- **WAJIB MENGHAPUS FOLDER "node_modules" sebelum mengumpulkan! Melanggar nilai tugas = 0!**
- **Nama Endpoint dan Nama Key untuk request HARUS SAMA dengan ketentuan SOAL**
- **Wajib menyertakan response code yang sesuai, jika tidak disertakan / sesuai maka kriteria yang berhubungan tidak akan dinilai**
- **Apabila data tidak tersimpan maka nomor bersangkutan tidak akan diperiksa**
- **Highlight kriteria yang dikerjakan dengan warna kuning dan kumpulkan word beserta dengan file tugas, apabila tidak dikumpulkan maka tugas tidak akan diperiksa.**
- **Akan ada pengurangan nilai sebesar -5 untuk setiap kriteria yang dihighlight namun tidak dikerjakan.**
- **MENCONTEK = Nilai MOD 2**
- **Nama database: T3_WS_9DigitNRP**

TUGAS : 30

SCORE	KRITERIA
0/2/4	[POST] /api/user - Berjalan dengan sesuai - Format id benar

	- Pengecekan sesuai
0/2/4	[POST] /api/event - Berjalan dengan sesuai - Format id benar - Pengecekan sesuai
0/2/4	[PUT] /api/event/:id - Berjalan dengan sesuai - Dapat mengupdate field spesifik - Pengecekan sesuai
0/3/5	[GET] /api/event/all - Berjalan dengan sesuai - Data diberikan sesuai (beserta peserta) - Pengecekan sesuai
0/3/5	[GET] /api/event?name= - Berjalan dengan sesuai - Data diberikan sesuai - Pengecekan sesuai
0/2/4	[POST] /api/event/register - Dapat mengupdate event - Pengecekan sesuai
0/2/4	[GET] /api/user/:id - Berjalan dengan sesuai - Data diberikan sesuai (beserta tiket) - Pengecekan sesuai
-40/0	Tidak menggunakan model
TUGAS: 30	

Menyetujui

Mengetahui

Penyusun Soal

(Evan Kusuma Susanto,
S.Kom., M.Kom.)
Koordinator Kuliah

(Grace Levina Dewi, M.Kom.)
Koordinator Laboratorium

(Lawrence Patrick)
Asisten