Institut Sains dan Teknologi Terpadu Surabaya
Jl. Ngagel Jaya Tengah 73 - 77, Surabaya 60284
Telp. (031) 5027920 Fax. (031) 5041509

Laboratorium	: L-304	Praktikum	: Service Oriented Architecture
Waktu	: Senin / 08.15-10.15	Jurusan	: S1 – Informatika
Minggu Ke	: 4	Tanggal	: 13 Maret 2023
Materi	: Form Validation	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.

Buatlah sebuah web service Freelance Online seperti Fiver. Web Service ini dapat membuat user baru, melakukan CRUD gigs, serta menerima dan request gigs. Endpoint yang harus dibuat adalah sebagai berikut:

1. **[POST] /api/users**
   Endpoint ini berguna untuk mendaftarkan akun baru pada layanan ini.
   - Data yang diperlukan adalah username, nama, dan date of birth. Semua field ini harus ada.
   - Minimal panjang username adalah 5.
   - Username harus alphanumeric dan string
   - Minimal panjang username adalah 5 dan maksimalnya adalah 20
   - Field date of birth harus memiliki format (DD/MM/YYYY).
   - Beri pengecekan username harus unik.

Contoh Request	Contoh Response

body:
- username: "joarelio"
- nama: "Jonathan Arelio"
- dob: "02/23/1992"
	status: 201
	body: {
	username: "joarelio",
	nama: "Jonathan Arelio",
	dob: "02/10/1992",
	created_at: "03/03/2023"
	}

body:
- username: "joa"
- nama: "Jonathan Arelio"
- dob: "02/10/1992"
	status: 400
	body: {
	message: "Panjang username harus minimal 5 huruf"
	}

body:
- username: "joarelio"
- nama: "Jonathan Arelio"
- dob: "02/23/1992"
	status: 400
	body: {
	message: "Date of Birth tidak sesuai format"
	}

body:
- username: "joarelio"
- nama: "Jonathan Arelio"
- dob: "02/23/1992"
	status: 400
	body: {
	message: "username harus unik!"
	}

2. **[POST] /api/gigs**
   Endpoint ini berguna untuk membuat gigs atau layanan yang ingin ditawarkan oleh freelancer.
   - Data yang diperlukan adalah nama, deskripsi, harga gig, username, dan durasi pengerjaan.

- Beri pengecekan harga gig harus berupa angka dan di atas 0.
- Beri pengecekan username harus sudah terdaftar
- Durasi yang diberikan adalah dalam format jam. Apabila diberikan 50 jam, maka response akan berupa ‘2 hari 2 jam’. Beri pengecekan harus berupa angka dan di atas 0.
- Berikan status gig sedang ‘Available’
- Generatekanlah gig id dengan format “GG” + 5 digit nomor urut serta buatkan field created_at

Contoh Request	Contoh Response

body:
- nama: "I will try to love you"
- deskripsi: "Saya akan mencoba mencintai anda"
- harga: "50000"
- username: "lawrence"
- durasi: "50"
	status: 201
	body: {
	id: "GG00001",
	nama: "I will try to love you",
	deskripsi: "Saya akan mencoba mencintai anda",
	harga: "Rp 50000",
	username: "lawrence",
	durasi: "2 hari 2 jam",
	status: "Available",
	created_at: "04/03/2023"
	}

body:
- nama: "I will try to love you"
- deskripsi: "Saya akan mencoba mencintai anda"
- harga: "50 ribu mas"
- username: "lawrence"
- durasi: "50"
	status: 400
	body: {
	message: "Harga harus berupa angka!"
	}

body:
- nama: "I will try to love you"
- deskripsi: "Saya akan mencoba mencintai anda"
- harga: "50 ribu mas"
- username: "aaa"
- durasi: "50"
	status: 404
	body: {
	message: "User tidak terdaftar!"
	}

body:
- nama: "I will try to love you"
- deskripsi: "Saya akan mencoba mencintai anda"
- harga: "50000"
- username: "lawrence"
- durasi: "berjam-jam"
	status: 404
	body: {
	message: "Durasi harus berupa angka!"
	}

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

Endpoint ini berguna untuk melakukan update gigs yang sudah pernah ada.

- Field yang harus ada adalah id gig dan username user. Beri pengecekan id gig ada dan username ada.
- Field yang dapat diupdate hanyalah nama, deskripsi, harga, durasi, dan status (“Available” atau “Unavailable”).
- Yang dapat mengupdate hanyalah user yang memiliki gig tersebut.
- Beri pengecekan id gig dan username terdaftar
- Berikan pengecekan yang sama seperti nomor 2 untuk field yang dapat diupdate

Contoh Request	Contoh Response

body:
- id_gig: "GG00016"
- username: "budi"
- nama: "Saya akan terjun payung"
- deskripsi: "Terjun payung sambil tidur"
- harga: "150000"
- durasi: "10"
- status: "Unavailable"
	status: 201
	body: {
	id: "GG00016",
	nama: "Saya akan terjun payung",
	deskripsi: "Terjun payung sambil tidur",
	harga: "Rp 150000",
	username: "budi",
	durasi: "0 hari 10 jam",
	status: "Unavailable",
	created_at: "26/02/2023"
	}

body:
- id_gig: "GG00016"
- username: "lawrence"
- nama: "Saya akan terjun payung"
- deskripsi: "Terjun payung sambil tidur"
- harga: "150000"
- durasi: "10"
- status: "Unavailable"
	status: 400
	body: {
	message: "Anda bukan pemilik gig ini!"
	}

4. [GET] /api/gigs

Endpoint ini berguna untuk melihat list gigs yang ada.

- Data yang dikembalikan adalah sebagai berikut

Contoh Request	Contoh Response
body: -	status: 200 body: [{ id: "GG00001", nama: "I will try to love you", deskripsi: "Saya akan mencoba mencintai anda", harga: "Rp 25000", username: "lawrence", durasi: "2 hari 2 jam", status: "Available", created_at: "04/03/2023" }, { id: "GG00002", nama: "Gig Kedua", deskripsi: "Deskripsi gig kedua", harga: "Rp 10000", username: "christianchen", durasi: "0 hari 2 jam", status: "Unavailable", created_at: "03/02/2023" }, ...]

5. [POST] /api/gigs/:id/order

Endpoint ini berguna untuk membuat orderan berdasarkan gig yang dipilih oleh user lain

- Field yang dibutuhkan adalah id gig, username, dan nominal bayar
- Beri pengecekan id gig dan username terdaftar, serta gig berstatus Available
- Beri pengecekan pemilik gig tidak dapat order gig milik sendiri
- Cek juga apakah nominal bayar sesuai dan pas dengan harga gig
- Generatekanlah order id dengan format “OR” + 5 digit nomor urut.

Contoh Request	Contoh Response

param:
- id: "GG00001"
 	
body:
- username: "budi"
- nominal: "50000"
	status: 201
	body: {
	message: "lawrence berhasil melakukan order gig (OR00014)!"
	}

param:
- id: "GG00001"
 	
body:
- username: "lawrence"
- nominal: "50000"
	status: 400
	body: {
	message: "Anda tidak bisa beli gig milik sendiri"
	}

param:
- id: "GG00122"
	status: 400
	body: {
	message: "Gig berstatus Unavailable"
	}

6. **[POST] /api/gigs/:id_gig/action/:id_order**

Endpoint ini berguna untuk menerima orderan yang dipesan user lain.

- Beri pengecekan id gig, id order, dan username terdaftar
- Beri pengecekan order yang di accept/reject merujuk ke gig yang sama
- Field yang diperlukan adalah username freelancer dan actionnya (“accept” atau “reject”)

Contoh Request	Contoh Response

param:
- id_gig: "GG00001"
- id_order: "OR00014"
 	
body:
- username: "lawrence"
- action: "accept"
	status: 201
	body: {
	message: "Orderan berhasil diterima dan dikerjakan!"
	}

param:
- id_gig: "GG00001"
- id_order: "OR00014"
 	
body:
- username: "lawrence"
- action: "reject"
	status: 201
	body: {
	message: "Orderan berhasil ditolak!"
	}

param:
- id_gig: "GG00001"
- id_order: "OR00004"
 	
body:
- username: "lawrence"
- action: "reject"
	status: 400
	body: {
	message: "ID Order tidak ditemukan di gig"
	}

Struktur Database Minggu 4 (PRAK_M4_SOA):

Users	Keterangan
username	varchar, PK
nama	varchar, NN
dob	date, NN
created_at	date, NN

Orders	Keterangan
id	varchar, PK
user_pembeli	varchar, NN, FK Users(username)
gig_id	varchar, NN, FK Gigs(id)
status	varchar

Gigs	Keterangan
id	varchar, PK
nama	varchar, NN
deskripsi	varchar, NN
harga	int, NN
user_penjual	varchar, NN, FK Users(username)
durasi	date, NN
status	varchar, NN
created_at	date, 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 kententuan SOAL**
- **Dilarang mengganti struktur DB yang diberikan. Melanggar = Materi 0**

- **TIDAK WAJIB MENGGUNAKAN DATABASE! Nomor yang menggunakan database akan mendapat poin +2 per nomornya.**
- **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**
- **WAJIB MENGGUNKAN PENGECKAN JOI!**

MATERI : 40

SCORE	KRITERIA
0/2/4/6/8/10	**[POST] /api/users** - Pengecekan field yang dibutuhkan sesuai - Pengecekan username sesuai - Pengecekan nama sesuai - Format date of birth sesuai - Pengecekan username unik - Data masuk ke database
0/2/4/6/8	**[POST] /api/gigs** - Pengecekan field yang dibutuhkan sesuai - Pengecekan harga gig sesuai - Pengecekan durasi sesuai - Format response durasi sesuai - Format id sesuai - Data masuk ke database
0/2/4/6	**[PUT] /api/gigs/:id** - Field yang boleh diupdate sesuai - Pengecekan gig dan user terdaftar - Pengecekan user yang mengupdate adalah pemilik gig sesuai - Pengecekan seperti nomor 2 - Data terupdate di database
0/1/3	**[GET] /api/gigs** - Data diminta sesuai
0/2/4/6/8	**[POST] /api/gigs/:id/order** - Pengecekan field yang dibutuhkan sesuai - Pengecekan gig dan user terdaftar - Pengecekan gig berstatus Available - Pengecekan pemilik gig sesuai - Pengecekan nominal bayar sesuai - Data masuk ke database - Format id sesuai
0/1/3/5	**[POST] /api/gigs/:id_gig/action/:id_order** - Pengecekan field yang dibutuhkan sesuai - Pengecekan gig dan user terdaftar - Pengecekan action sesuai - Berhasil melakukan action gig (terupdate di db)
@2 (Max: 12)	Menggunakan Database untuk setiap nomornya
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.

Buatlah sebuah web service Pembelian Tiket Konser. Web Service ini akan menyediakan layanan untuk mengadakan event konser, lalu user bisa membeli tiketnya. User nantinya juga dapat membuat akun serta topup saldo untuk membeli tiket konser. **Database silahkan dirancang sendiri sesuai kebutuhan**. Endpoint yang harus dibuat adalah sebagai berikut:

1. **[POST] /api/users**
   Endpoint ini berguna untuk mendaftarkan user baru.
   - Field yang dibutuhkan adalah email, NIK, nama lengkap, nomor telpon, dan tanggal lahir
   - Berikan beberapa pengecekan yaitu (berikan pesan error yang sesuai):
     - Email harus unik dan berupa email valid
     - NIK harus angka dan tepat 16 digit
     - Nomor telepon harus angka dan minimal 10 digit
     - Tanggal lahir harus tanggal dengan format DD/MM/YYYY
   - Buatkan id user dengan format 2 digit terakhir YY tanggal join + MM tanggal join+ 3 digit nomor urut. Contoh: 12/02/2023, maka 2302001

Contoh Request	Contoh Response

body:
- email: "nicoletta@gmail.com"
- NIK: "3578070601020006"
- nama_lengkap: "Nicoletta Valencia"
- no_telpon: "08212345678"
- tanggal_lahir: "02/01/2001"
	status: 201
	body: {
	id: "2303001",
	email: "nicoletta@gmail.com",
	NIK: "3578070601020006",
	nama_lengkap: "Nicoletta Valencia",
	no_telpon: "08212345678",
	tanggal_lahir: "02/01/2001",
	saldo: "Rp 0"
	}

2. **[POST] /api/topup**
   Endpoint ini berguna untuk menambahkan saldo milik user.
   - Field yang diperlukan adalah user id, NIK, dan nominal saldo
   - Berikan beberapa pengecekan yaitu (berikan pesan error yang sesuai):
     - User yang dicari terdaftar
     - NIK user harus sesuai dengan yang ada di database
     - Nominal saldo merupakan angka dan nilainya harus di atas 10000
   - Apabila berhasil, tambahkan saldo user sebanyak nominal.

Contoh Request	Contoh Response

body:
- user_id: "2302067"
- NIK: "3579801234567890"
- nominal: "30000"
	status: 201
	body: {
	id: "2302067",
	email: "lawrence@gmail.com",
	NIK: "3579801234567890",
	nama_lengkap: "Lawrence Patrick",
	saldo_baru: "Rp 50000"
	}

body:
- user_id: "2302067"
- NIK: "3578070601020006"
- nominal: "30000"
	status: 400
	body: {
	message: "NIK tidak sesuai dengan yang terdaftar!"
	}

3. **[POST] /api/konser**

Endpoint ini berguna untuk menambahkan sebuah konser yang akan datang.

- Field yang harus diikutsertakan adalah nama konser, nama artis, tempat konser, tanggal konser, dan harga tiket.
- Berikan beberapa pengecekan yaitu (berikan pesan error yang sesuai):
  - Nama konser harus string dan di atas 3 huruf
  - Tanggal konser harus minimal 1 bulan setelah tanggal sekarang (bukan 13 Maret 2023).
  - Tanggal konser harus memiliki format DD/MM/YYYY HH:mm
  - Harga tiket harus berupa angka dan di atas 50000 dan dalam rupiah
- Generatekan id konser dengan format "KR" + 3 digit nomor urut
- Apabila berhasil, masukkan data ke database dan beri field penonton 0

Contoh Request	Contoh Response

body:
- nama_konser: "So Happy It Hurts 2023"
- nama_artis: "Bryan Adams"
- tempat: "The Star Theatre"
- tanggal: "13/03/2023 20:00"
- harga: "1770000"
	status: 201
	body: {
	id: "KR001",
	nama_konser: "So Happy It Hurts 2023",
	nama_artis: "Bryan Adams",
	tempat: "The Star Theatre",
	tanggal: "13 Maret 2023 20:00",
	harga: "Rp 1770000",
	penonton: "0"
	}

4. **[GET] /api/konser /:id**

Endpoint ini berguna untuk melihat data konser yang ada sesuai parameter id yang diberikan

- Tampilkan data yang dibutuhkan seperti contoh di bawah.
- Apabila id konser tidak diberikan, maka tampilkan data semua user dan urutkan id secara ascending

Contoh Request	Contoh Response

param:
- id: "KR002"
	status: 200
	body: {
	id: "KR002",
	nama_konser: "Head in The Clouds Jakarta 2023",
	nama_artis: "88 Rising",
	tempat: "Community Park PIK2",
	tanggal: "03 December 2022 11:00",
	harga: "1150000",
	penonton: "32"
	}

param:
-
	status: 200
	body: [
	{
	id: "KR001",
	...
	},
	{
	id: "KR002",
	...
	}
	]

5. **[POST] /api/tickets**

Endpoint ini berguna untuk membeli tiket untuk konser yang dipilih.

- Field yang harus diikutsertakan adalah id konser, id user, jumlah tiket, dan email user.
- Email user disertakan gunanya untuk verifikasi dengan id user saja. Beri pengecekan email harus sesuai dengan id user yang diberikan.
- Pastikan saldo milik user cukup untuk membeli tiket sesuai jumlah belinya. Apabila cukup, maka saldo user akan terpotong sebesar total beli.
- Tambahkan juga jumlah penonton di konser yang dipilih sebanyak jumlah tiket.
- Generatekan nomor tiket sebanyak jumlah beli dengan format "TK" + 5 digit nomor urut

Contoh Request	Contoh Response

body:
- id_konser: "KR002"
- id_user: "2303003"
- jumlah: "3"
- email: "ingnatiusodi@gmail.com"
	status: 201
	body: {
	konser: "Head in THe Clouds Jakarta 2023",
	nama_user: "Ignatius Odi",
	jumlah: "3",
	total_harga: "3450000",
	tiket: [
	"TK00001",
	"TK00002",
	"TK00003",
	]
	}

body:
- id_konser: "KR002"
- id_user: "2303003"
- jumlah: "3"
- email: "lawrencepatrick@gmail.com"
	status: 400
	body: {
	message: "Verifikasi akun user gagal!"
	}

6. **[POST] /api/tickets/:id/refund**

Endpoint ini berguna untuk melakukan refund tiket yang ingin dibatalkan.

- Field yang harus diikutsertakan adalah id konser, id user, jumlah refund, dan email user.
- Email user disertakan gunanya untuk verifikasi dengan id user saja. Beri pengecekan email harus sesuai dengan id user yang diberikan.
- Tambahkan saldo user dengan rumus (harga tiket x jumlah refund) dikurangi fee refund 10%.
- Kurangi jumlah penonton di konser yang dipilih sebanyak jumlah tiket refund.

Contoh Request	Contoh Response

body:
- id_konser: "KR002"
- id_user: "2303003"
- jumlah: "1"
- email: "ignatiusodi@gmail.com"
	status: 200
	body: {
	message: "Tiket berhasil direfund dan saldo bertambah Rp 1035000"
	}

7. **[GET] /api/users/:id**

   Endpoint ini berguna untuk melihat data milik user beserta dengan tiket yang pernah dibelinya sesuai parameter id user yang diberikan

   - Tampilkan data yang dibutuhkan seperti contoh di bawah.
   - Apabila id user tidak diberikan, maka tampilkan data semua user dan urutkan id secara Ascending

Contoh Request	Contoh Response

param:
- id: "2303003"
	status: 200
	body: {
	id: "2303003",
	email: "ignatiusodi@gmail.com",
	nama_lengkap: "Ignatius Odi",
	tanggal_lahir: "21/05/2002",
	saldo: "Rp 12000",
	tickets: [
	{
	konser: "Head in THe Clouds Jakarta 2023",
	jumlah: 2
	},
	{
	konser: "So Happy It Hurts 2023",
	jumlah: 1
	}
	]
	}

param: -	status: 200 body: [{ id: "2303001", ... }, { id: "2303003", ... }, ...]

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! TIDAK MENGUMPULKAN DATABASE, TUGAS = 0!**
- **Nama Endpoint dan Nama Key untuk request HARUS SAMA dengan kententuan 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: T4_SOA_9DigitNRP**

TUGAS : 30

SCORE	KRITERIA
0/1/3/5	**[POST] /api/users** - Pengecekan email unik - Pengecekan NIK sesuai - Pengecekan nomor telepon sesuai - Pengecekan format DOB sesuai - Format ID sesuai - Data masuk ke database
0/2/4	**[POST] /api/topup** - Pengecekan user terdaftar - Pengecekan NIK sesuai - Pengecekan nominal sesuai - Saldo user bertambah
0/1/3/5	**[POST] /api/konser** - Pengecekan nama sesuai - Pengecekan tanggal sesuai - Pengecekan harga sesuai - Format ID sesuai - Data masuk ke database
0/1/3	**[GET] /api/konser/:id** - Data diminta sesuai - Diurutkan secara ascending
0/2/4/6	**[POST] /api/tickets** - Verifikasi email user sesuai - Pengecekan saldo sesuai - Saldo terpotong dan jumlah penonton bertambah - Format ID sesuai - Data masuk ke database
0/2/4	**[POST] /api/tickets/:id/refund** - Verifikasi email user sesuai - Perhitungan saldo refund benar - Saldo bertambah dan jumlah penonton berkurang
0/1/3	**[GET] /api/users/:id** - Data diminta sesuai - Diurutkan secara ascending
TUGAS: 30	

Menyetujui	Mengetahui	Penyusun Soal
(Dr. Ir. Esther Irawati Setiawan, S.Kom., M.Kom.)	(Grace Levina Dewi, M.Kom.)	(Lawrence Patrick)
Koordinator Kuliah	Koordinator Laboratorium	Asisten