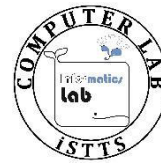




# Institut Sains dan Teknologi Terpadu Surabaya

Jl. Ngagel Jaya Tengah 73 - 77, Surabaya 60284

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



Laboratorium : L-304

Waktu : Senin / 08.15-10.15

Minggu Ke : 1

Materi : Node JS & REST Service Basic

Praktikum : Service Oriented Architecture

Jurusan : S1 – Informatika

Tanggal : 20 Februari 2023

Jenis Soal : Materi dan Tugas

## MATERI (TOTAL = 40)

Buatlah sebuah web service untuk website toko alat musik. Website ini memiliki 6 buah endpoint yang harus dibuat. Tujuan utama pada website ini adalah untuk melakukan CRUD alat musik. Tabel dibawah ini merupakan tabel bantuan yang berisi id tipe dan nama tipe alat musik. Isinya paten hanya 3 data ini saja.

type_id	type_name
TY001	Guitar
TY002	Piano
TY003	Drum

Endpoint yang harus dibuat adalah sebagai berikut

### 1. [POST] /api/instruments/:type\_id

Endpoint ini digunakan untuk menambah alat musik baru dengan memberikan data alat musik berserta tipe alat musik tersebut melalui path variabel. Instrument\_id akan tergenerate secara otomatis dengan format "IN" + 3 digit nomor urut.

Data instrument yang harus disertakan adalah

Key	Value type
instrument_name	string
instrument_specification	string
instrument_price	number
instrument_brand	string
instrument_stock	number

Berikan pengecekan:

- Type id sesuai dengan tabel
- Data tidak boleh ada yang kosong
- Data price > 0 dan stock > 0

Contoh Request	Contoh Response
<pre>param • type_id: "TY001" body: • instrument_name: "Godin Multiact Encore" • instrument_specification: "Mahogany neck, rosewood fingerboard" • instrument_price: 150000</pre>	<pre>status: 201, body: {   instrument_id: "IN001",   instrument_brand: "Godin",   instrument_name: "Godin Multiact Encore",   instrument_specification: "Mahogany neck, rosewood fingerboard",   instrument_price: 150000,   instrument_stock: 8,</pre>

<ul style="list-style-type: none"> <li>• instrument_brand: "Godin"</li> <li>• instrument_stock: 8</li> </ul>	<pre>instrument_type_id: "TY001", instrument_type_name: "Guitar", instrument_created_at: "29/01/2023" }</pre>
param <ul style="list-style-type: none"> <li>• type_id: "TY010"</li> </ul> body: <ul style="list-style-type: none"> <li>• instrument_name: "DGX670"</li> <li>• instrument_specification: "88 Keys"</li> <li>• instrument_price: 10000</li> <li>• instrument_brand: "Yamaha"</li> <li>• instrument_stock: 1</li> </ul>	<pre>status: 404, body: {   message: "Type tidak ditemukan!" }</pre>
param body:	<pre>status: 400, body: {   message: "Param atau field body tidak sesuai ketentuan!" }</pre>
param <ul style="list-style-type: none"> <li>• type_id: ""</li> </ul> body: <ul style="list-style-type: none"> <li>• instrument_name: ""</li> <li>• instrument_specification: ""</li> <li>• instrument_price: 0</li> <li>• instrument_brand: ""</li> <li>• instrument_stock: 0</li> </ul>	<pre>status: 400, body: {   message: "Param atau field body tidak sesuai ketentuan!" }</pre>

## 2. [PUT] /api/instruments/:instrument\_id

Endpoint ini digunakan untuk mengedit alat musik yang sudah ada dengan memberikan data yang baru beserta id alat musik tersebut melalui path variabel. Field yang dapat di edit hanya instrument\_price, instrument\_stock dan instrument\_specification.

Berikan pengecekan:

- Instrument terdaftar dalam sistem
- Data tidak boleh ada yang kosong
- Data price > 0 dan stock > 0

Contoh Request	Contoh Response
param <ul style="list-style-type: none"> <li>• instrument_id: "IN001"</li> </ul> body: <ul style="list-style-type: none"> <li>• instrument_specification: "ORI, BNI B"</li> <li>• instrument_price: 20000</li> <li>• instrument_stock: 3</li> </ul>	<pre>status: 200, body: {   instrument_id: "IN001",   instrument_specification: "ORI, BNI B",   instrument_price: 20000,   instrument_stock: 3,   instrument_updated_at: "30/01/2023" }</pre>
param <ul style="list-style-type: none"> <li>• instrument_id: "IN010"</li> </ul> body: <ul style="list-style-type: none"> <li>• instrument_specification: "88 Keys"</li> <li>• instrument_price: 10000</li> <li>• instrument_stock: 1</li> </ul>	<pre>status: 404, body: {   message: "Instrument tidak ditemukan!" }</pre>
param <ul style="list-style-type: none"> <li>• instrument_id: "IN001"</li> </ul> body: <ul style="list-style-type: none"> <li>• instrument_specification: ""</li> <li>• instrument_price: 0</li> <li>• instrument_stock: 0</li> </ul>	<pre>status: 400, body: {   message: "Param atau field body tidak sesuai ketentuan!" }</pre>

### 3. [DELETE] /api/instruments/:instrument\_id

Endpoint ini digunakan untuk menghapus data instrument yang terdaftar pada sistem. Berikan pengecekan bahwa instrument harus terdaftar pada sistem.

Contoh Request	Contoh Response
<pre>param   • instrument_id: "IN001"</pre>	<pre>status: 200, body: {   instrument_id: "IN001",   message: "Instrument telah dihapus!" }</pre>
<pre>param instrument_id: "IN111"</pre>	<pre>status: 404, body: {   message: "Instrument tidak ditemukan!" }</pre>
<pre>param</pre>	<pre>status: 400, body: {   message: "Param atau field body tidak sesuai ketentuan!" }</pre>

### 4. [GET] /api/instruments?

Endpoint ini digunakan untuk menampilkan instrument yang terdaftar pada sistem sesuai dengan format. Request ini memiliki 2 jenis query:

- Tanpa query (hanya search\_name tapi tanpa data) => tampilkan semua instrument yang terdaftar
- Dengan query search\_name => tampilkan semua instrument yang name nya mengandung search\_name dan terdaftar

Contoh Request	Contoh Response
<pre>query:   • search_name: "eNCo"</pre>	<pre>status: 200, body: [   {     instrument_id: "IN001",     instrument_brand: "Godin",     instrument_name: "Godin Multiact Encore",     instrument_stock: 8,     instrument_type_id: "TY001",     instrument_type_name: "Guitar"   },   ... ]</pre>
<pre>query:   • search_name: ""</pre>	<pre>status: 200, body: [   {     instrument_id: "IN001",     instrument_brand: "Godin",     instrument_name: "Godin Multiact Encore",     instrument_stock: 8,     instrument_type_id: "TY001",     instrument_type_name: "Guitar"   },   {     instrument_id: "IN002",     instrument_brand: "Yamaha",     instrument_name: "Yamaha P-125",     instrument_stock: 15,     instrument_type_id: "TY002",     instrument_type_name: "Piano"   } ]</pre>

	<pre>       },       {         instrument_id: "IN003",         instrument_brand: "Yamaha",         instrument_name: "Yamaha C315",         instrument_stock: 15,         instrument_type_id: "TY001",         instrument_type_name: "Guitar"       }     ]   } } </pre>
<b>query:</b> <ul style="list-style-type: none"> <li>search_name: "wkwk"</li> </ul>	<pre> status: 200, body: {   message: "Tidak ada instrument yang sesuai!" } </pre>

### 5. [GET] /api/instruments/:instrument\_id

Endpoint ini digunakan untuk mendapatkan data dari instrument tertentu yang sudah terdaftar pada sistem. Berikan pengecekan bahwa instrument harus terdaftar.

Contoh Request	Contoh Response
<b>param</b> <ul style="list-style-type: none"> <li>instrument_id: "IN001"</li> </ul>	<pre> status: 200, body: {   instrument_id: "IN001",   instrument_brand: "Godin",   instrument_name: "Godin Multiact Encore",   instrument_specification: "ORI, BNI B",   instrument_price: 20000,   instrument_stock: 3,   instrument_type_id: "TY001",   instrument_type_name: "Guitar",   instrument_created_at: "29/01/2023",   instrument_updated_at: "30/01/2023" } </pre>
<b>param</b> <ul style="list-style-type: none"> <li>instrument_id: "IN111"</li> </ul>	<pre> status: 404, body: {   message: "Instrument tidak ditemukan!" } </pre>

### 6. [GET] /api/instruments/:type\_id

Endpoint ini digunakan untuk mendapatkan data dari instrument tertentu yang sudah terdaftar pada sistem. Berikan pengecekan bahwa type id harus ada dalam tabel.

Contoh Request	Contoh Response
<b>param</b> <ul style="list-style-type: none"> <li>type_id: "TY001"</li> </ul>	<pre> status: 200, body: {   type_name: "Guitar",   instrument_list: [     {       instrument_id: "IN001",       instrument_brand: "Godin",       instrument_name: "Godin Multiact Encore",       instrument_stock: 8     },     {       instrument_id: "IN003",       instrument_brand: "Yamaha",       instrument_name: "Yamaha C315",       instrument_stock: 15     }   ] } </pre>

	<pre>     },     ...   ] } </pre>
param <ul style="list-style-type: none"> <li>type_id: "TY111"</li> </ul>	<pre> status: 404, body: {   message: "Type tidak ditemukan!" } </pre>

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

**PERHATIKAN KETENTUAN DI BAWAH:**

- 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**
- Nama Endpoint, Field dan Response harus sama dengan ketentuan setiap soal. Melanggar = nilai kriteria tersebut div 2

**MATERI : 40**

SCORE	KRITERIA
<b>No 1 (8)</b>	
0/2/4	data baru berhasil di input dengan benar, response status dan body benar (status 201)
0/1	response status dan body benar (status 404)
0/1/3	response status dan body benar (status 400)
<b>No 2 (6)</b>	
0/2/4	data berhasil di update dengan benar, response status dan body benar (status 200)
0/1	response status dan body benar (status 404)
0/1	response status dan body benar (status 400)
<b>No 3 (6)</b>	
0/2/4	data berhasil di delete dengan benar, response status dan body benar (status 200)
0/1	response status dan body benar (status 404)
0/1	response status dan body benar (status 400)
<b>No 4 (7)</b>	
0/1/3	response status dan body benar (status 200)
0/1/3	response status dan body benar (status 200)
0/1	response status dan body benar (status 200)
<b>No 5 (6)</b>	
0/1/3	response status dan body benar (status 200)
0/1/3	response status dan body benar (status 404)
<b>No 6 (7)</b>	
0/2/4	response status dan body benar (status 200)
0/1/3	response status dan body benar (status 404)
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 Mixue Online. Endpoint menyediakan layanan registrasi user, penambahan produk baru, pemesanan produk, hingga pengambilan data user maupun . Endpoint yang harus dibuat adalah sebagai berikut:

### 1. [POST] /api/pelanggans/

Endpoint ini berguna untuk mendaftarkan pelanggan baru di Mixue. Terdapat field nama, jenis kelamin, nomor HP, dan email. Pelanggan nantinya akan mendapatkan ID Mixue yang akan degenerate secara otomatis dengan format “MXE”+ 3 digit nomor urut (Contoh: MXE002, MXE003). Berilah pengecekan jika isi field tidak sesuai, tidak terisi, atau format jenis kelamin tidak sesuai (“laki” atau “perempuan”).

Contoh Request	Contoh Response
<pre>body: {   nama: "Cherilyn Eugenia"   jenis_kelamin: "perempuan"   nomor_HP: "66778899"   email: "cherilyn_e20@mhs.istts.ac.id"</pre>	<pre>status: 201 body: {   id: "MXE002",   nama: "Cherilyn Eugenia",   jenis_kelamin: "perempuan",   nomor_HP: "66778899",   email: "cherilyn_e20@mhs.istts.ac.id", }</pre>
<pre>body: {   nama: ""   jenis_kelamin: "macho"   nomor_HP: "testes"   email: ""</pre>	<pre>status: 400 body: {   message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

### 2. [PUT] /api/pelanggans/:id

Endpoint ini berguna untuk mendapatkan data pelanggan yang dimaksud dari parameter id yang dimilikinya. Respons yang harus diberikan adalah id, nama, jenis kelamin, nomor hp, email, dan list pembeliannya. Beri pengecekan apabila user tidak ditemukan.

Contoh Request	Contoh Response
<pre>param:   id: "MXE001"</pre>	<pre>status: 200 body: {   id: "MXE001",   nama: "Alexander Kevin",   jenis_kelamin: "laki",   nomor_HP: "11223344",   email: "alexander_k20@mhs.istts.ac.id",   pembelian: [     {       trans_id: "T003",       nama: "Lucky Strawberry Sundae",       jumlah: "1"     }   ] }</pre>
<pre>param:   id: "😭😭"</pre>	<pre>status: 404 body: {   message: "Pelanggan Mixue tidak ditemukan!" }</pre>

### 3. [PUT] /api/pelanggans/:id

Endpoint ini berguna untuk melakukan update data pelanggan. Data yang dapat diupdate hanyalah nama, jenis kelamin, dan nomor HP. Berikan pengecekan yang sama seperti nomor 1. Apabila berhasil, tampilkan data baru user yang diupdate.

Contoh Request	Contoh Response
<pre>param: - id: "MXE002"  body: - nama: "Lawrence Patrick" - jenis_kelamin: "laki" - nomor_HP: "22222"</pre>	<pre>status: 201 body: {   id: "MXE002",   nama: "Lawrence Patrick",   jenis_kelamin: "laki",   nomor_HP: "22222",   email: "cherilyn_e20@mhs.istts.ac.id" }</pre>

### 4. [POST] /api/menu

Endpoint ini berguna untuk membuat sebuah minuman Mixue baru. Data yang diperlukan untuk membuat sebuah minuman adalah nama, harga, dan jenis minuman. Minuman yang berhasil dimasukkan akan membuat kode minuman dengan format "M" + 3 digit minuman dan tanggal ditambahkan akan mengambil tanggal menu ditambahkan.. Beri pengecekan harga harus diisi, berupa angka, dan di atas 0. Beri pengecekan field juga harus diisi.

Contoh Request	Contoh Response
<pre>body: - nama: "DUREN Fruit Tea" - harga: "20000" - jenis_minuman: "Fruit Tea"</pre>	<pre>status: 201 body: {   id: "M006",   nama: "DUREN Fruit Tea",   harga: "20000",   jenis_minuman: "Fruit Tea",   tanggal_ditambahkan: "24/02/2023" }</pre>
<pre>body: - nama: "Choco Sundae" - harga: "abc" - jenis_minuman: ""</pre>	<pre>status: 400 body: {   message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

### 5. [GET] /api/menu?nama=

Endpoint ini berguna untuk mengambil sebuah minuman yang memiliki nama sesuai dengan parameter nama yang diberikan. Berikan data id, nama, harga, dan banyak dibeli. Pencarian nama minuman adalah bersifat case insensitive. Beri pengecekan apabila minuman tidak ditemukan.

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

<pre>param: - nama: "mI"</pre>	<pre>status: 200 body: [   {     id: "M004",     nama: "Strawberry Mi-Shake",     harga: "16000",     banyak_dibeli: "121"   },   {     id: "M010",     nama: "Pearl Milk Tea",     harga: "19000",     banyak_dibeli: "34"   } ]</pre>
<pre>param: - id: "oogabooga"</pre>	<pre>status: 404 body: {   message: "Minuman Mixue tidak ditemukan!" }</pre>

## 6. [POST] /api/pesanans/

Endpoint ini berguna untuk membeli minuman yang tersedia di Mixue. Field yang diperlukan adalah id pelanggan, id minuman, dan jumlah. Respons akan mengembalikan sebuah id transaksi dan total harga minuman yang dibeli. Beri pengecekan apabila pelanggan atau minuman tidak ditemukan. Beri pengecekan juga jumlah minuman harus di atas 0.

Contoh Request	Contoh Response
<pre>body: - id_pelanggan: "MXE002" - id_minuman: "M004" - jumlah: "3"</pre>	<pre>status: 201 body: {   id_transaksi: "T003",   nama_pelanggan: "Cherilyn Eugenia",   nama_minuman: "Strawberry Mi-Shake",   jumlah: "3",   harga_total: "48000" }</pre>
<pre>body: - id_pelanggan: "MXE999" - id_minuman: "M999" - jumlah: "-1"</pre>	<pre>status: 400 body: {   message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

## 7. [DELETE] /api/pesanans/:id

Endpoint ini berguna untuk menghapus data pesanan user yang telah masuk. Berikan data id transaksi dalam parameter. Beri pengecekan apabila pesanan tidak ditemukan.

Contoh Request	Contoh Response
<pre>param: - id: "T009"</pre>	<pre>status: 200 body: {   message: "Pesanan berhasil dibatalkan!" }</pre>
<pre>param: - id: "TT"</pre>	<pre>status: 200 body: {   message: "Pesanan Mixue tidak ditemukan!" }</pre>

## 8. [POST] /api/isi-ruko-kosong

Endpoint ini berguna untuk membeli (memperlebar) franchise Mixue untuk mengisi ruko kosong milik anda. Field yang dibutuhkan adalah nama pemilik, no KTP, NPWP, dan jumlah franchise. Untuk setiap



franchise akan dikenakan harga 700 juta. Buatlah id ruko dengan format “R” + 3 digit nomor urut ruko. Beri pengecekan semua field harus diisi dan jumlah franchise minimal harus 1.

Contoh Request	Contoh Response
<pre>body: - nama_pemilik: "Lawrence Patrick" - no_KTP: "3578012345678" - no_NPWP: "123141521" - jumlah_franchise: "2"</pre>	<pre>status: 201 body: [   {     id_ruko: "R002",     nama_pemilik: "Lawrence Patrick",     no_KTP: "3578012345678",     total_harga: "700 juta"   },   {     id_ruko: "R003",     nama_pemilik: "Lawrence Patrick",     no_KTP: "3578012345678",     total_harga: "700 juta"   } ]</pre>
<pre>body: - nama_pemilik: "" - no_KTP: "3578012345678" - no_NPWP: "abde" - jumlah_franchise: "-6"</pre>	<pre>status: 400 body: {   message: "Terdapat field yang kosong/format salah/tidak ada!" }</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**
- **DATA WAJIB TERSIMPAN MENGGUNAKAN ARRAY OF JSON OBJECT YANG DITAMPUNG DALAM VARIABLE SEHINGGA SETIAP KALI APLIKASI DIRESTART MAKA DATA AKAN TERHAPUS.**
- **DILARANG MENGINSTALL NPM APAPUN SELAIN EXPRESS DAN BODY-PARSER. APABILA DATA TIDAK TERSIMPAN MAKA TUGAS TIDAK AKAN DIPERIKSA.**

**TUGAS : 30**

SCORE	KRITERIA
0/2/4	<b>[POST] /api/pelanggans/</b> Berjalan dengan sesuai, data masuk, format id benar, pengecekan sesuai
0/1/3	<b>[GET] /api/pelanggans/:id</b>

	Berjalan sesuai ketentuan, data diminta sesuai, pengecekan sesuai
0/1/3	<b>[PUT] /api/pelanggans/:id</b> Berjalan sesuai ketentuan, data terupdate s
0/2/5	<b>[POST] /api/menu</b> Berjalan sesuai ketentuan, data masuk, format kode benar, tanggal ditambahkan sesuai, pengecekan sesuai
0/2/4	<b>[GET] /api/menu?nama=</b> Berjalan sesuai ketentuan, pencarian sesuai, data diminta sesuai, pengecekan sesuai
0/2/4	<b>[POST] /api/pesanans/</b> Berjalan sesuai ketentuan, data masuk, format kode benar, pengecekan sesuai
0/1/3	<b>[DELETE] /api/pesanans/:id</b> Berjalan sesuai ketentuan, data berhasil dihapus, pengecekan sesuai
0/2/4	<b>[POST] /api/isi-ruko-kosong</b> Berjalan sesuai ketentuan, data masuk, format ID benar, pengecekan sesuai
TUGAS: 30	

Menyetujui

Mengetahui

Penyusun Soal

(Dr. Ir. Esther Irawati  
Setiawan, S.Kom., M.Kom.)  
Koordinator Kuliah

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

(Lawrence Patrick)  
Asisten