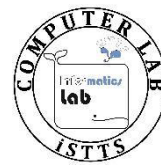




# Institut Sains dan Teknologi Terpadu Surabaya

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 : 3

Materi : Object Relational Mapping

Praktikum : Service Oriented Architecture

Jurusan : S1 – Informatika

Tanggal : 06 Maret 2023

Jenis Soal : Materi dan Tugas

## MATERI (TOTAL = 40)

Buatlah sebuah web service untuk program spotify sederhana. Pada program ini pengguna dapat melakukan CRUD lagu dan album, menambahkan lagu-lagu ke dalam album dan membuat queue untuk urutan lagu yang akan diputar. Materi ini wajib dikerjakan dengan menggunakan database dengan konsep ORM Sequelize. Endpoint yang harus dibuat yaitu:

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

Endpoint ini digunakan untuk menambahkan lagu baru ke dalam sistem. Lagu memiliki id dengan format 2 huruf pertama uppercase + 3 digit no urut. Body duration dalam detik, ditampilkan dalam menit dan detik. Berikan pengecekan duration tidak boleh kurang dari 30 detik.

Contoh Request	Contoh Response
body: <ul style="list-style-type: none"><li>- song_title:"Damage"</li><li>- song_genre:"pop"</li><li>- song_duration:153</li></ul>	status: 201, body: { song_id: "DA001", song_title: "Damage", song_genre: "pop", song_duration: "2 menit 33 detik" }
body: <ul style="list-style-type: none"><li>- song_title:""</li><li>- song_genre:""</li><li>- song_duration:20</li></ul>	status: 400, body: { message: "Body tidak sesuai!" }
body:	status: 400, body: { message: "Body tidak sesuai!" }

### 2. [POST] /api/albums

Endpoint ini digunakan untuk menambahkan album baru ke dalam sistem. Apabila statusnya public tidak perlu memberikan password atau bisa diisi "-". Format id album adalah ALB + 3 digit no urut.

Contoh Request	Contoh Response
body: <ul style="list-style-type: none"><li>- album_title:"Pop Radio"</li><li>- album_status:"private"</li><li>- album_password:"123"</li><li>- album_max_songs:3</li></ul>	status: 201, body: { album_id: "ALB001", album_title: "Pop Radio", album_max_songs: 3, album_total_songs: 0, album_total_duration: "-", }

	<pre>album_status: "private", album_password: "123" }</pre>
<pre>body: - album_title:"" - album_max_songs:0 - album_status:"" - album_password:""</pre>	<pre>status: 400, body: {   message: "Body tidak sesuai!" }</pre>
<pre>body:</pre>	<pre>status: 400, body: {   message: "Body tidak sesuai!" }</pre>

### 3. [POST] /api/songs/add-to-album/:id

Endpoint ini digunakan untuk menambahkan lagu yang sudah ada ke dalam sebuah album. Album public dan private dapat ditambahkan lagu, namun album yang sifatnya private harus memberikan password agar bisa menambahkan lagu. Berikan pengecekan id lagu dan album harus ada, lagu belum pernah ditambahkan ke album (apabila sudah ditambahkan keluaran pesan "Lagu sudah ada di album <album\_title>!") dan apabila albumnya private maka passwordnya harus benar.

Contoh Request	Contoh Response
<pre>param: - id:"DA001" body: - album_id:"ALB001" - password:"123"</pre>	<pre>status: 200, body: {   album_id: "ALB001",   album_title: "Pop Radio",   album_total_songs: 1,   album_total_duration: "2 menit 33 detik",   album_songs: [     {       Song_id: "DA001",       song_title: "Damage",       song_duration: "2 menit 33 detik",     },     ...   ] }</pre>
<pre>param: - id:"DA001" body: - album_id:"ALB001" - password:"123"</pre>	<pre>status: 400, body: {   message: "Lagu sudah ada di album Pop Radio!" }</pre>
<pre>param: - id:"DA111" body: - album_id:"ALB001" - password:"123"</pre>	<pre>status: 404, body: {   message: "Lagu tidak ada!" }</pre>
<pre>param: - id:"DA002" body: - album_id:"ALB001" - password:"qwe"</pre>	<pre>status: 400, body: {   message: "Password salah!" }</pre>

### 4. [PUT] /api/albums/:id

Endpoint ini digunakan untuk mengedit sebuah album. Field yang dapat diedit adalah status dan max\_songs. Berikan pengecekan status hanya berisi public/private dan max\_songs lebih dari 0. Apabila max\_songs yang baru lebih kecil dari yang sebelumnya maka hapuslah lagu mulai dari yang terakhir sampai jumlahnya sesuai dengan max\_songs yang baru. Apabila albumnya berstatus private, berikan juga passwordnya. Tampilkan data lama dan barunya.

Contoh Request	Contoh Response
param: - id:" ALB001" body: - status:"public" - max_songs:5 - password:"123"	status: 200, body: { "old": { album_id: "ALB001", album_title: "Pop Radio", album_max_songs: 3, album_total_songs: 0, album_total_duration: "-", album_status: "private", album_password: "123" }, "new": { album_id: "ALB001", album_title: "Pop Radio", album_max_songs: 5, album_total_songs: 0, album_total_duration: "-", album_status: "public", album_password: "-" } }
param: - id:" ALB002" body: - status:"public" - max_songs:5 - password:"qwe"	status: 404, body: { message: "Album tidak ada!" }
param: - id:" ALB001" body: - status:"public" - max_songs:5 - password:"qwe"	status: 400, body: { message: "Password salah!" }

#### 5. [POST] /api/queues/:id

Endpoint ini digunakan untuk melakukan menambahkan lagu yang ada di dalam album ke dalam queue, sebelum menambahkan bersihkan queue terlebih dahulu. Setelah lagu ditambahkan, terdapat opsi shuffle. Shuffle lagu berarti merubah urutan lagu menjadi random.

Contoh Request	Contoh Response
param: - id:"ALB001" body: - shuffle:"false"	status: 200, body: { queue_total_songs: 1, queue_total_duration: "2 menit 33 detik", queue_songs: [ { Song_id: "DA001", }

	<pre>         song_title: "Damage",         song_duration: "2 menit 33 detik",     },     ... ] } </pre>
<pre> param:   - id:"ALB001" body:   - shuffle:"true" </pre>	<pre> status: 200, body: {   queue_total_songs: 1,   queue_total_duration: "2 menit 33 detik",   queue_songs: [     ...,     {       Song_id: "DA001",       song_title: "Damage",       song_duration: "2 menit 33 detik",     }   ] } </pre>

### Struktur Database Minggu 3 (PRAK\_M3\_SOA):

Songs	Keterangan
song_id	varchar, PK
song_title	varchar, NN
song_genre	varchar, NN
song_duration	varchar, NN

Albums	Keterangan
album_id	varchar, PK
album_title	varchar, NN
album_total_songs	int, NN
album_total_duration	int, NN
album_max_songs	Int, NN

Playlists	Keterangan
song_id	varchar, PK
album_id	varchar, PK

Queues	Keterangan
song_id	varchar, NN, FK Songs(song_id)

**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
- Apabila materi tidak menggunakan konsep ORM maka kriteria endpoint tersebut akan dikurangi 1 poin

**MATERI : 40**

SCORE		KRITERIA
<b>No 1 (5)</b>		
0/3		data baru berhasil di input dengan benar, response status dan body benar (status 201)
0/2		response status dan body benar (status 400)
<b>No 2 (5)</b>		
0/3		data baru berhasil di input dengan benar, response status dan body benar (status 201)
0/2		response status dan body benar (status 400)
<b>No 3 (12)</b>		
0/3/6		response status dan body benar (status 200)
0/2		response status dan body benar (status 404)
0/2/4		response status dan body benar (status 400)
<b>No 4 (9)</b>		
0/2/5		response status dan body benar (status 200)
0/2		response status dan body benar (status 404)
0/2		response status dan body benar (status 400)
<b>No 5 (6)</b>		
0/3/6		response status dan body benar (status 200)
<b>Lain-Lain (3)</b>		
@1 (3)		terdapat init untuk semua model menggunakan ORM Sequelize (Songs, Albums, Playlists, Queues)
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 Layanan Pembelian Investasi Online. Endpoint nantinya dapat mendaftarkan nasabah baru, menambah instrumen investasi baru serta mengganti perubahan harga, melakukan pembelian investasi, dan mendapatkan data nasabah dan investasi. **Database silahkan dirancang sendiri sesuai kebutuhan.** Endpoint yang harus dibuat adalah sebagai berikut:

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

Endpoint ini berguna untuk mendaftarkan akun baru.

- Data yang diperlukan adalah email, nama lengkap, nomor KTP, nomor HP, 6 digit pin aplikasi.
- Pin digit nantinya akan dienkripsi menggunakan Bcrypt dengan 10 rounds.
- Generatekan id unik user dengan format "BI" + 3 digit nomor urut.
- Saldo RDN (Rekening Dana Nasabah) dan total portofolio awal mula nasabah adalah 0.
- Beri pengecekan semua field terisi, ada, dan sesuai format, serta email harus unik.

Contoh Request	Contoh Response
<pre>body: - email: "davidcahyadi@gmail.com" - nama_lengkap: "David Cahyadi" - nomor_KTP: "357908812240002" - nomor_hp: "082123455231" - pin: "654321"</pre>	<pre>status: 201 body: {   id: "BI002",   email: "davidcahyadi@gmail.com",   nama_lengkap: "David Cahyadi",   nomor_KTP: "357908812240002",   nomor_hp: "082123455231",   pin: "\$2a\$10\$L2.9LHBt2ERf1vhHIIa4g09nZ92lCcKCnuUxlb7kw0a/RQIpWmeIC",   saldo_RDN: "Rp 0",   total_portofolio: "Rp 0" }</pre>
<pre>body: - email: "" - nama_lengkap: "Christian Chen" - nomor_KTP: "bbbbbb" - pin: "1123"</pre>	<pre>status: 400 body: {   message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

## 2. [GET] /api/nasabah/:id

Endpoint ini berguna untuk mendapatkan user berdasarkan parameter id yang diberikan.

- Apabila parameter tidak diberikan maka tampilkan semua user terdaftar.
- Data yang harus ditampilkan dapat dilihat di contoh di bawah (tampilkan portofolionya)
- Jika pencarian tidak ada maka berikan pesan.

Contoh Request	Contoh Response
<pre>param: - - id: "BI001"</pre>	<pre>status: 200 body: {   id: "BI001",   email: "lawrencepatrick@gmail.com",   nama_lengkap: "Lawrence Patrick",   nomor_hp: "081231513457",   pin: "\$2as105S66rXnA7iG30Yzd4KvY@seM7wa.D15oVvZx4GQyCrktb0px9gekU5y",   saldo_RDN: "Rp 999999",   total_portofolio: "Rp 610000",   portofolio: [     {       kode: "GGRM",       total: "Rp 550000",       jumlah: "10"     },     {       kode: "SPTP",       jumlah: "Rp 60000",       jumlah: "4"     }   ] }</pre>
<pre>param: - -</pre>	<pre>status: 200 body: [   {     id: "BI001",     ...   },   {     id: "BI002",     ...   },   ... ]</pre>
<pre>param: - - id: "B00002"</pre>	<pre>status: 404 body: {   message: "User tidak ditemukan!" }</pre>

## 3. [POST] /api/topup

Endpoint ini berguna untuk melakukan topup saldo RDN milik user.

- Field yang dibutuhkan adalah id user dan nominal topup.
- Berikan pengecekan id user sudah terdaftar dan minimal topup adalah Rp 10.000
- Apabila berhasil maka tambahkan RDN ke saldo RDN user.

Contoh Request	Contoh Response
<pre>body: - id_user: "BI002" - nominal: "800000"</pre>	<pre>status: 201 body: {   message: "Berhasil menambah saldo David Cahyadi sebanyak Rp 800000!" }</pre>

<pre>body: - id_user: "BI002" - nominal: "5000"</pre>	<pre>status: 400 body: {   message: "Nominal minimal adalah Rp 10.000!" }</pre>
---	---

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

Endpoint ini berguna untuk mendaftarkan perusahaan terbuka yang ingin menjual instrument investasi.

- Data yang diperlukan adalah nama perusahaan, nama instrumen investasi, kode singkatan instrumen, kategori instrument, dan harga jual
- Berikan pengecekan harga jual harus di atas 0, nama perusahaan harus diakhiri dengan kata “tbk” (terbuka) dengan case insensitive, dan singkatan instrumen harus 4 huruf.
- Kode singkatan instrument ini yang menjadi **primary key**.
- Kategori instrumen harus salah satu dari list berikut
  - “SA” = Saham
  - “RU” = Reksadana Pasar Uang
  - “RT” = Reksadana Pendapatan Tetap
  - “OB” = Obligasi
- Nantinya nasabah dapat melakukan pembelian investasi yang sudah didaftarkan

Contoh Request	Contoh Response
<pre>body: - nama_perusahaan: "PT Gudang Garam Tbk" - instrumen: "Gudang Garam Saham" - singkatan: "GGRM" - kategori: "SA" - harga: "55000"</pre>	<pre>status: 201 body: {   nama_perusahaan: "PT Gudang Garam Tbk",   instrumen: "Gudang Garam Saham",   singkatan: "GGRM",   kategori: "SA",   harga: "55000" }</pre>
<pre>body: - nama_perusahaan: "PT Gudang Garam Tbk" - instrumen: "Gudang Garam Saham" - singkatan: "GGRMA" - kategori: "ZZ" - harga: "0"</pre>	<pre>status: 400 body: {   message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

#### 5. [GET] /api/investasi/:kode

Endpoint ini berguna untuk mendapatkan instrument investasi berdasarkan parameter kode singkatan yang diberikan.

- Apabila parameter tidak diberikan maka tampilkan semua investasi terdaftar
- Urutkan secara Ascending dari kode singkatan.
- Data yang harus ditampilkan dapat dilihat di contoh di bawah.
- Jika pencarian tidak ada maka berikan pesan.

Contoh Request	Contoh Response
<pre>param: - kode: "SPTP"</pre>	<pre>status: 200 body: {   nama_perusahaan: "PT Syailendra Capital Tbk",   instrumen: "Syailendra Pendapatan Tetap Premium",   singkatan: "SPTP",   kategori: "RT",   harga: "15000" }</pre>



<pre>param: - "</pre>	<pre>status: 200 body: [   {     nama_perusahaan: "PT Gudang Garam Tbk",     instrumen: "Gudang Garam Saham",     singkatan: "GGRM",     kategori: "SA",     harga: "55000"   },   {     nama_perusahaan: "PT Syailendra Capital Tbk",     instrumen: "Syailendra Pendapatan Tetap Premium",     singkatan: "SPTP",     kategori: "RT",     harga: "15000"   },   ... ]</pre>
<pre>param: - kode: "BBBB"</pre>	<pre>status: 404 body: {   message: "Instrumen Investasi tidak ditemukan!" }</pre>

## 6. [PUT] /api/investasi/harga

Endpoint ini berguna untuk merubah harga jual dari sebuah instrument investasi.

- Field yang dibutuhkan adalah kode investasi dan harga beli yang baru.
- Beri pengecekan kode investasi harus ada dan harga jual baru harus di atas 0.
- Apabila berhasil updatelah harga investasi.
- Berikan pesan berhasil seperti contoh di bawah. Apabila harga naik, maka tulisannya **naik dari**, apabila harga turun, maka tulisannya **turun dari**.

Contoh Request	Contoh Response
<pre>body: - kode_investasi: "GGRM" - harga_jual: "60000"</pre>	<pre>status: 201 body: {   message: "Harga jual GGRM naik dari Rp 55000 ke Rp 60000!" }</pre>
<pre>body: - kode_investasi: "GGRM" - harga_jual: "0"</pre>	<pre>status: 400 body: {   message: "Terdapat field yang kosong/format salah/tidak ada!" }</pre>

## 7. [POST] /api/investasi/beli

Endpoint ini berguna untuk nasabah dapat melakukan pembelian instrument investasi.

- Field yang dibutuhkan adalah kode singkatan investasi, id nasabah, dan nominal beli.
- Beri pengecekan kode investasi dan id nasabah terdaftar.
- Beri pengecekan saldo RDN user harus mencukupi.
- Beri pengecekan juga bahwa nominal beli harus kelipatan dari harga beli investasi. Contoh: harga jual GGRM = Rp 55.000, nominal beli = Rp 130.000. Maka contoh tadi tidak valid.
- Generatekan kode pembelian dengan format "BL" + 5 digit nomor urut
- Jumlah beli adalah kelipatan belinya.
- Apabila berhasil membeli, kurangi saldo RDN milik nasabah

Contoh Request	Contoh Response
<pre>body: - kode_investasi: "BBCA" - id_user: "BI0002" - nominal: "850000"</pre>	<pre>status: 201 body: {   kode_beli: "BL00002",   kode_investasi: "BBCA",   nama_user: "David Cahyadi",   nominal: "Rp 850000",   jumlah_beli: "1",   saldo_RDN: "Rp 100000" }</pre>
<pre>body: - kode_investasi: "BBCA" - id_user: "BI001" - nominal: "1000000"</pre>	<pre>status: 400 body: {   message: "Nominal tidak sesuai kelipatan harga beli" }</pre>

## 8. [POST] /api/investasi/jual

Endpoint ini berguna untuk nasabah dapat melakukan penjualan instrumen investasi.

- Field yang dibutuhkan adalah kode singkatan investasi, id nasabah, dan jumlah nominal.
- Beri pengecekan kode investasi dan id nasabah terdaftar.
- Beri pengecekan apakah nasabah memiliki nominal yang ingin dijual di Instrumen yang dipilih.

Contoh:

- Mau jual GGRM Rp 150.000, tapi hanya punya GGRM Rp 15.000. (MAKA TIDAK BISA)
- Generatekan kode penjualan dengan format "JL" + 5 digit nomor urut
- Apabila berhasil membeli, tambahkan saldo RDN milik nasabah

Contoh Request	Contoh Response
<pre>body: - kode_investasi: "GGRM" - id_user: "BI001" - harga_jual: "400000"</pre>	<pre>status: 201 body: {   kode_beli: "JL00016",   kode_investasi: "GGRM",   nama_user: "Lawrence Patrick",   nominal: "Rp 400000",   saldo_RDN: "Rp 1399999" }</pre>
<pre>body: - kode_investasi: "GGRM" - id_user: "BI001" - harga_jual: "10000000"</pre>	<pre>status: 400 body: {   message: "Anda tidak memiliki investasi sebanyak itu!" }</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! TIDAK MENGUMPULKAN DATABASE, 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\_SOA\_9DigitNRP**
- **Apabila materi tidak menggunakan konsep ORM maka kriteria endpoint tersebut akan dikurangi 1 poin**
- **Wajib menggunakan Association dalam menghubungkan antar model**

## TUGAS : 30

SCORE	KRITERIA
0/2/4	<b>[POST] /api/nasabah</b> <ul style="list-style-type: none"> <li>- Berjalan dengan sesuai</li> <li>- Pin terenkripsi</li> <li>- Format id benar</li> <li>- Pengecekan sesuai</li> </ul>
0/2/4	<b>[GET] /api/nasabah/:id</b> <ul style="list-style-type: none"> <li>- Berjalan dengan sesuai</li> <li>- Data diminta sesuai (beserta portofolio)</li> <li>- pengecekan sesuai</li> </ul>
0/1/3	<b>[POST] /api/topup</b> <ul style="list-style-type: none"> <li>- Berjalan dengan sesuai</li> <li>- Saldo RDN nasabah bertambah</li> <li>- pengecekan sesuai</li> </ul>
0/2/4	<b>[POST] /api/investasi</b> <ul style="list-style-type: none"> <li>- Berjalan dengan sesuai</li> <li>- Pengecekan harga jual sesuai</li> <li>- Pengecekan nama perusahaan sesuai</li> <li>- Pengecekan singkatan sesuai</li> </ul>
0/1/2	<b>[GET] /api/investasi/:kode</b> <ul style="list-style-type: none"> <li>- Berjalan dengan sesuai</li> <li>- Data diminta sesuai</li> <li>- Diurutkan Ascending</li> <li>- Pengecekan sesuai</li> </ul>
0/1/3	<b>[PUT] /api/investasi/harga</b> <ul style="list-style-type: none"> <li>- Harga terupdate dengan benar</li> <li>- Pesan berhasil sesuai</li> <li>- Pengecekan sesuai</li> </ul>
0/3/5	<b>[POST] /api/investasi/beli</b> <ul style="list-style-type: none"> <li>- Berjalan dengan sesuai</li> <li>- Format kode beli benar</li> <li>- Saldo RDN berkurang</li> <li>- Pengecekan sesuai</li> </ul>
0/3/5	<b>[POST] /api/investasi/jual</b> <ul style="list-style-type: none"> <li>- Berjalan dengan sesuai</li> <li>- Format kode jual benar</li> <li>- Saldo RDN bertambah</li> <li>- Pengecekan sesuai</li> </ul>
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