



Laboratorium : L-204
Waktu : 08:15 – 10:15
Minggu Ke : 6
Materi : Business Model

Praktikum : Service Oriented Architecture
Jurusan : S1 Informatika
Tanggal : 14 April 2021
Jenis Soal : Materi & Tugas

MATERI (TOTAL: 40)

Buatlah web service dari penyedia informasi wisata. Endpoint yang harus dibuat adalah

1. [POST] /api/users/ yang berfungsi untuk menambahkan user baru kedalam sistem. Data yang harus dikirimkan adalah
 - email: Email dari user yang akan ditambahkan
 - nama_user: nama dari user yang akan ditambahkan
 - tanggal_lahir: tanggal lahir dari user yang akan ditambahkan dengan format dd-mm-yyyy

Berikan pengecekan bahwa email dari user belum pernah terdaftar pada sistem dan berikan juga bahwa umur dari user saat mendaftar adalah 13 tahun. Saat user berhasil dibuat saldo awal dari user dan API hit dari user adalah 0. Response harus dikembalikan adalah API key user yang merupakan string random yang mengandung huruf A-Z, a-z, 0-9 dengan panjang 16 karakter. Pastikan key yang tergenerate tidak kembar dengan user lain. Contoh response yang harus dikembalikan adalah

```
{
  "email": "juli@gmail.com",
  "nama_user": "Julianto",
  "tanggal_lahir": "11/08/1996",
  "saldo": 0,
  "api_hit": 0,
  "api_key": "kXvf01R2VNGa9yUU"
}
```

2. [POST] /api/users/topup yang berfungsi untuk **menambahkan** saldo dari user. Sertakan API key yang didapat saat register pada header dengan key "x-auth-token". Data yang harus disertakan adalah
 - saldo: saldo yang akan ditambahkan

Berikan pengecekan bahwa saldo yang ditambahkan hanya berupa angka saja. Response yang dikembalikan adalah

```
{
  "email": "juli@gmail.com",
  "saldo_awal": 0,
  "saldo_akhir": 500000
}
```

3. [PUT] /api/users/recharge yang berfungsi untuk menambahkan API hit dari user. Sertakan API key yang didapat saat register pada header dengan key "x-auth-token". Berikan pengecekan bahwa API key yang diberikan valid. Jika API key valid maka tambahkan API hit dari user sejumlah 5 dan saldo dari user akan dikurangi sebesar 10000. Berikan pengecekan bahwa saldo dari user mencukupi. Contoh response yang harus dikembalikan adalah

```
{
  "email": "juli@gmail.com",
  "api_hit_awal": 0,
  "api_hit_akhir": 5
}
```

4. [POST] /api/informations yang berfungsi untuk menambahkan informasi tempat wisata. Sertakan API key yang didapat saat register pada header dengan key “x-auth-token”. Data yang harus disertakan adalah
- nama_tempat: Nama dari tempat wisata yang akan ditambahkan
 - kota: kota tempat wisata yang ditambahkan
 - deskripsi: deskripsi singkat dari tempat wisata
 - kategori: kategori tempat wisata (0 untuk ruang terbuka, 1 taman bermain, 2 bangunan bersejarah)

Berikan pengecekan bahwa API key yang diberikan valid dan berikan juga pengecekan bahwa kategori yang diinputkan sesuai. Saat berhasil menambahkan tempat wisata kurangi API hit dari user sebesar 5.

5. Response yang harus dikembalikan adalah

```
{
  "kode": 1,
  "nama_tempat": "Tugu Pahlawan",
  "kota": "Surabaya",
  "deskripsi": "Tempat dimana perjuangan arek-arek Surabaya diabadikan",
  "kategori": "Tempat Bersejarah"
}
```

5. [GET] /api/informations?kota=&kategori= yang berfungsi untuk mendapatkan data tempat wisata sesuai dengan query yang diberikan. Sertakan API key yang didapat saat register pada header dengan key “x-auth-token”. Query yang digunakan bersifat like dan case insensitive. Saat API dipanggil maka kurangi API hit dari user sebesar 1. Berikan pengecekan bahwa API hit dari user tersebut mencukupi. Contoh response yang diberikan adalah
Jika tidak terdapat query

```
[
  {
    "kode": 1,
    "nama_tempat": "Tugu Pahlawan",
    "kota": "Surabaya",
    "kategori": "Tempat Bersejarah"
  },
  {
    "kode": 2,
    "nama_tempat": "Jatim Park 3",
    "kota": "Batu",
    "kategori": "Taman Bermain"
  },
  {
    "kode": 3,
    "nama_tempat": "Wisata Mangrove",
    "kota": "Surabaya",
    "kategori": "Ruang Terbuka"
  },
  {
    "kode": 4,
    "nama_tempat": "Jatim Park 2",
    "kota": "Batu",
    "kategori": "Taman Bermain"
  }
]
```

Jika terdapat query nama=park

```
[
  {
    "kode": 2,
    "nama_tempat": "Jatim Park 3",
    "kota": "Batu",
    "kategori": "Taman Bermain"
  },
  {
    "kode": 4,
    "nama_tempat": "Jatim Park 2",
    "kota": "Batu",
    "kategori": "Taman Bermain"
  }
]
```

Jika terdapat query nama=3 dan kategori=2

```
{
  "kode": 2,
  "nama_tempat": "Jatim Park 3",
  "kota": "Batu",
  "kategori": "Taman Bermain"
}
```

6. [GET] /api/informations/:kode_info yang berfungsi untuk menampilkan detail tempat wisata. Sertakan API key yang didapat saat register pada header dengan key "x-auth-token". Saat API dipanggil maka kurangi API hit dari user sebesar 1. Contoh response yang diberikan adalah

```
{
  "kode": 1,
  "nama_tempat": "Tugu Pahlawan",
  "kota": "Surabaya",
  "deskripsi": "Tempat dimana perjuangan arek-arek Surabaya diabadikan",
  "kategori": "Tempat Bersejarah",
  "added_by": "juli@gmail.com"
}
```

PERHATIKAN KETENTUAN DIBAWAH:

- Wajib menghapus FOLDER "node_modules" sebelum mengumpulkan! Melanggar = Nilai materi = 0
- Nama Endpoint dan nama key untuk request HARUS SAMA dengan ketentuan SOAL
- Dilarang mengganti struktur database yang diberikan. Melanggar = materi 0
- Kembar atau mencontek teman = 0
- Highlight kriteria yang dikerjakan, dan kumpulkan beserta file materi, -10 untuk setiap kriteria yang dihighlight tetapi tidak dikerjakan. Materi tidak diperiksa bila tidak mengumpulkan soal.
- ZIP / RAR File yang dikumpulkan bernama: M6_<NRP> Contoh: M6_217116627.zip

SCORE	KRITERIA
0/3	Dapat menambahkan user baru
0/2	Terdapat pengecekan saat menambahkan user baru
0/3	Dapat melakukan generate API key
0/3	Dapat melakukan topup
0/3	Dapat melakukan recharge API
0/2	Terdapat pengecekan saldo saat melakukan recharge API

SCORE	KRITERIA
0/3	Dapat menambahkan info tempat wisata baru
0/3	Terdapat pengecekan saat menambahkan tempat wisata baru
0/3/6/9	Dapat menampilkan tempat wisata sesuai dengan query
0/3	Format response nomor 5 sesuai
0/3	Dapat menampilkan detail tempat wisata
0/3	Format response nomor 6 sesuai

TUGAS (TOTAL: 30)

Buatlah web service dari website penyedia layanan podcast. Endpoint yang harus dibuat adalah sebagai berikut:

1. [POST] /api/users/ yang berfungsi untuk menambahkan user baru kedalam sistem. Data yang harus dikirimkan adalah
 - email: Email dari user yang akan ditambahkan
 - nama_user: nama dari user yang akan ditambahkan
 - tanggal_lahir: tanggal lahir dari user yang akan ditambahkan dengan format dd-mm-yyyy
 - tipe_user: C untuk creator R untuk reguler

Berikan pengecekan bahwa email dari user belum pernah terdaftar pada sistem dan berikan juga bahwa umur dari user saat mendaftar adalah 13 tahun. Saat user berhasil dibuat saldo awal dari user adalah 0. Response harus dikembalikan adalah API key user yang merupakan string random yang mengandung huruf A-Z, a-z, 0-9 dengan panjang 16 karakter. Pastikan key yang tergenerate tidak kembar dengan user lain. Contoh response yang harus dikembalikan adalah

```
{
  "email": "david@gmail.com",
  "nama_user": "David Cahyadi",
  "tanggal_lahir": "29/12/1996",
  "saldo": 0,
  "api_hit": 0,
  "api_key": "YVcz6mn9XbvUAWXB",
  "tipe_user": "Creator"
}
```

2. [POST] /api/users/topup yang berfungsi untuk **menambahkan** saldo dari user. Sertakan API key yang didapat saat register pada header dengan key "x-auth-token". Data yang harus disertakan adalah
 - saldo: saldo yang akan ditambahkan

Berikan pengecekan bahwa saldo yang ditambahkan hanya berupa angka saja. Response yang harus diberikan adalah

```
{
  "email": "david@gmail.com",
  "saldo_awal": 0,
  "saldo_akhir": 50000
}
```

3. [PUT] /api/users/recharge yang berfungsi untuk menambahkan API hit dari user. Sertakan API key yang didapat saat register pada header dengan key "x-auth-token". Berikan pengecekan bahwa API key yang diberikan valid. Jika API key valid maka tambahkan API hit dari user sejumlah 10 dan saldo dari user akan dikurangi sebesar 10000. Berikan pengecekan bahwa saldo dari user mencukupi. Contoh response yang harus dikembalikan adalah

```
{
  "email": "david@gmail.com",
  "api_hit_awal": 0,
  "api_hit_akhir": 10
}
```

4. [POST] /api/content/ yang berfungsi untuk menambahkan konten baru kedalam sistem. Sertakan API key yang didapat saat register pada header dengan key "x-auth-token".
 - nama_konten: kode dari organisasi yang menyelenggarakan kegiatan
 - status: 0 untuk free dan 1 untuk premium
 - link: link dari konten
 - kategori: kategori dari konten (0 untuk edukasi, 1 entertainment, 2 olahraga)

Berikan pengecekan bahwa user yang menambahkan content tersebut adalah creator dan kategori konten sesuai dengan ketentuan. Response yang dikembalikan adalah kode content dengan ketentuan <P/F>+4 digit nomor urut. Saat berhasil menambahkan konten kurangi API hit dari user sebesar 5. Contoh response yang dikembalikan adalah sebagai berikut

```
{
  "kode_konten": "P0001",
  "nama_konten": "Belajar React 100 jam jadi",
  "status": "Premium",
  "link": "google.com",
  "kategori": "Edukasi"
}

{
  "kode_konten": "F0001",
  "nama_konten": "Basket jago by Charles",
  "status": "Free",
  "link": "youtube.com",
  "kategori": "Olahraga"
}
```

5. [GET] /api/content?kategori=&nama= yang digunakan untuk menampilkan content-content yang sudah terdaftar pada sistem berdasarkan query yang diberikan. Query yang dikirimkan bersifat like dan case insensitive. Sertakan API key yang didapat saat register pada header dengan key “x-auth-token”. Setiap kali API key dipanggil kurangi API hit dari user sebesar 1. Berikan pengecekan bahwa API hit dari user tersebut mencukupi. Response yang dikembalikan adalah Contoh tanpa query

```
[
  {
    "kode_konten": "P0001",
    "nama_konten": "Belajar React 100 jam jadi",
    "kategori": "Edukasi"
  },
  {
    "kode_konten": "F0001",
    "nama_konten": "Basket jago by Charles",
    "kategori": "Olahraga"
  },
  {
    "kode_konten": "P0002",
    "nama_konten": "Belajar musik asik pasti jago",
    "kategori": "Entertainment"
  }
]
```

6. [GET] /api/content/:id_konten yang berfungsi untuk menampilkan detail konten. Sertakan API key yang didapat saat register pada header dengan key “x-auth-token”. Setiap kali API key dipanggil kurangi API hit dari user sebesar 2 untuk konten yang premium dan 1 untuk konten free. Berikan juga pengecekan bahwa API key dari user tersebut mencukupi dan data dari konten tersebut terdapat pada sistem. Contoh response yang diberikan adalah

```
{
  "kode_konten": "F0001",
  "nama_konten": "Basket jago by Charles",
  "status": "Free",
  "link": "youtube.com",
  "kategori": "Olahraga"
}
```

Beberapa hal yang harus diperhatikan:

- **Wajib menghapus FOLDER “Node Modules” sebelum mengumpulkan! Melanggar = Nilai tugas = 0**
- **Export database tugas dengan nama T6_<4_DIGIT_NRP> Contoh: T6_6653. Tidak ada database, tugas tidak diperiksa. Nama database salah, tugas div 2**
- **Highlight kriteria yang dikerjakan, dan kumpulkan beserta file tugas, -10 untuk setiap kriteria yang dihighlight tetapi tidak dikerjakan. Tugas tidak diperiksa bila tidak mengumpulkan soal.**

SCORE	KRITERIA
0/3	Dapat menambahkan user baru beserta pengecekan
0/3	Dapat generate API key
0/2/4	Dapat melakukan topup dan recharge API key
0/3	Dapat menambahkan konten baru
0/3	Terdapat pengecekan saat menambahkan konten
0/2/4/6	Dapat menampilkan konten sesuai dengan query
0/2	Response dari konten sesuai contoh
0/3	Dapat menampilkan detail informasi konten
0/3	Terdapat pengecekan konten premium dan free

Menyetujui

Mengetahui

Penyusun Soal

(Dr. Esther Irawati S.)
Koordinator Kuliah

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

(Matthew Gunawan)
Asisten