



Institut Sains dan Teknologi Terpadu Surabaya

Jl. Ngagel Jaya Tengah 73 - 77, Surabaya 60284

Telp. (031) 5027920 Fax. (031) 504150



Laboratorium : L-204

Waktu : 13.15-15.15

Minggu Ke : 2

Materi : REST Service Basic

Praktikum : Service Oriented Architecture

Jurusan : S1 Informatika

Tanggal : 4 Maret 2020

Jenis Soal : Materi dan Tugas

MATERI (TOTAL: 40)

Buatlah sebuah web service yang digunakan untuk menghandle transaksi data untuk sistem informasi mahasiswa sebuah universitas. **Semua data wajib tersimpan namun dilarang menggunakan MySQL tetapi menggunakan JSON Object yang disimpan dalam variable.** Berikut ini adalah endpoint yang wajib dibuat:

1. Buatlah sebuah function POST /addNewMahasiswa yang digunakan untuk melakukan menambah mahasiswa baru ke dalam sistem. Data yang disertakan dalam fungsi ini adalah nama mahasiswa, kode jurusan (10 untuk informatika, 11 untuk sistem informasi, dan 17 untuk desain komunikasi visual), tahun masuk dan juga nomor telepon. Response yang dikembalikan dari request ini adalah NRP yang di generate oleh sistem dengan format sebagai berikut: 2 + 2 digit belakang tahun masuk + 2 digit kode jurusan + 3 nomor urut (autoincrement).
2. Buatlah sebuah function PUT /updateMahasiswa/:NRP yang digunakan untuk mengubah data mahasiswa sesuai dengan NRP yang disertakan. Data yang dapat diubah hanyalah nomor telepon saja sehingga data yang disertakan pada fungsi ini adalah NRP dan juga nomor telepon baru. Response yang dikembalikan dari request ini adalah JSON Array yang berisi status dan juga message. Apabila NRP tidak ditemukan, berikan message "NRP Tidak ditemukan". Apabila NRP ditemukan maka ubahlah data nomor telepon mahasiswa yang bersangkutan dan berilah message "Data mahasiswa berhasil diubah".
3. Buatlah sebuah function GET /getMahasiswaByJurusan/:kodejurusan untuk mendapatkan data mahasiswa yang memiliki jurusan sesuai dengan kode jurusan yang disertakan pada parameter. Response yang dikembalikan berupa JSON Array dengan data semua mahasiswa yang memiliki jurusan sesuai dengan kode jurusan yang dicari.

```
[
  {
    "nrp": "21611001",
    "nama": "adrian",
    "kode_jurusan": 11,
    "no_telepon": "08123045812522"
  }, {
    "nrp": "21611002",
    "nama": "yohanes",
    "kode_jurusan": 11,
    "no_telepon": "08395729185291"
  }
]
```

4. Buatlah sebuah function POST /addMataKuliah untuk menambah data mata kuliah baru ke dalam sistem. Data yang disertakan pada dalam fungsi ini adalah nama mata kuliah dan juga jumlah sks. Response yang dikembalikan dari request ini adalah kode mata kuliah yang merupakan autogenerate dari sistem dengan format "MK + 3 digit nomor urut (autoincrement)".
5. Buatlah sebuah function POST /tambahKelas/:nrp/:kodematakuliah untuk mendaftarkan mahasiswa

dengan nrp sesuai parameter kedalam kelas matakuliah dengan kode matakuliah sesuai parameter. Data yang dikirimkan pada response ini adalah nrp dan juga kode mata kuliah. Response yang dikembalikan request ini adalah JSON Array berisi status dan juga message. Apabila NRP atau Kode mata kuliah tidak ditemukan, berikan pesan “NRP / Mata kuliah tidak ditemukan”.

6. Buatlah sebuah function GET /getMatakuliah/:nrp untuk menampilkan mata kuliah apa saja yang diambil oleh mahasiswa sesuai dengan NRP pada parameter. Response yang dikembalikan berupa JSON Array dengan data semua mata kuliah yang diambil oleh mahasiswa yang bersangkutan.

```
[
  {
    "kode_mata_kuliah": "MK001",
    "nama_mata_kuliah": "SOA",
    "jumlah_SKS": 3
  }, {
    "kode_mata_kuliah": "MK004",
    "nama_mata_kuliah": "INTRO TO PROGRAMMING",
    "jumlah_SKS": 3
  }
]
```

7. Buatlah sebuah function DELETE /deleteMataKuliah/:kode_matakuliah untuk menghapus data matakuliah yang ada pada sistem. Data yang dikirimkan pada request ini adalah kode mata kuliah yang hendak di delete. Response yang dikembalikan dari request ini adalah JSON Array berisi status dan message. Apabila kode mata kuliah tidak ditemukan berikan message “Mata kuliah tidak ditemukan”, apabila berhasil berikan message “Mata kuliah berhasil dihapus”.

MATERI : 40

SCORE	KRITERIA
8/4/0	Endpoint POST /addNewMahasiswa berjalan dengan benar
5/2/0	Endpoint PUT /updateMahasiswa berjalan dengan benar
6/3/0	Endpoint GET /getMahasiswaByJurusan berjalan dengan benar
5/2/0	Endpoint POST /addMataKuliah berjalan dengan benar
6/3/0	Endpoint POST /getMataKuliah berjalan dengan benar
5/2/0	Endpoint POST /tambahKelas berjalan dengan benar
5/2/0	Endpoint DELETE /deleteMataKuliah berjalan dengan benar

TUGAS (TOTAL = 30)

Buatlah sebuah web service untuk web repository music online “SPOTYKU” (layaknya SPOTIFY API). Pengguna dapat melakukan registrasi user baru baik sebagai artis maupun user biasa. Pengguna juga dapat membuat playlist yang dapat berisikan lebih dari 1 lagu. Berikut ini adalah ketentuan endpoint yang harus dibuat:

1. Buatlah sebuah function POST /registerUser untuk registrasi user baru. Pengguna dapat melakukan registrasi dirinya sebagai artis maupun pengguna biasa. Data yang dikirimkan pada fungsi ini antara lain adalah idpengguna (autoincrement), email, nama, password dan juga tipe pengguna (0 untuk pengguna biasa, 1 untuk artis). Response dari request ini adalah JSON Array yang berisikan status dan message. Berikan pengecekan bahwa email belum terpakai sebelumnya, jika email sudah terpakai sebelumnya maka response akan mengirimkan status fail dan juga message “Registrasi gagal! Email sudah terpakai sebelumnya”, sebaliknya jika berhasil maka status success dan message “Registrasi user {email} berhasil!”.
2. Buatlah sebuah function POST/uploadSong untuk mengupload lagu baru yang dapat dilakukan oleh artis. Data yang dikirim adalah id lagu(autoincrement), judul lagu, genre (pop, country, jazz), durasi(dalam detik) dan juga id pengguna. Response dari request ini adalah JSON Array yang berisikan status dan message. Berikan pengecekan bahwa id pengguna ada dan juga bertipe artis dan judul lagu

belum pernah ada sebelumnya. Jika id pengguna tidak ada berikan message “Pengguna tidak ditemukan”, jika id pengguna bukan bertipe artis berikan message “Pengguna tidak mempunyai hak untuk mengupload lagu” jika judul lagu sudah ada sebelumnya berikan message “Judul lagu sudah ada sebelumnya”.

3. Buatlah sebuah function POST /createPlaylist untuk membuat playlist baru yang dapat dilakukan oleh pengguna biasa. Data yang dikirim adalah id playlist (autoincrement), email pengguna, nama playlist, dan id lagu yang ingin diinputkan pada playlist (dalam bentuk array). Berikan pengecekan bahwa semua judul lagu yang diinputkan ada dan email pengguna valid.
4. Buatlah sebuah function GET /searchSongByKeyword/:keyword untuk mendapatkan semua lagu yang mengandung keyword yang diinputkan. Keyword ini bersifat LIKE. Response dari request ini adalah JSON Array yang berisikan data semua lagu yang mengandung keyword yang diinputkan.
5. Buatlah sebuah function DELETE /deleteSong/:judul untuk menghapus lagu dengan judul yang sesuai dengan parameter. Response dari request ini adalah JSON Array berisi status dan juga message. Apabila judul tidak ditemukan, berikan message “Judul lagu tidak ditemukan”.
6. Buatlah sebuah function GET /getPlaylist/:idpengguna untuk mendapatkan semua data playlist (termasuk lagu apa saja yang ada dalam playlist itu) yang dimiliki oleh email pengguna yang diinputkan pada parameter. Kembalikan response dalam bentuk JSON Array.

```
[
  {
    "id_playlist": 2,
    "judul_playlist": "Chill On",
    "collection_lagu": [1,3,4]
  }, {
    "id_playlist": 4,
    "judul_playlist": "Let It Go",
    "collection_lagu": [2,3,4,7]
  }
]
```

7. Buatlah sebuah function PUT /updatePlaylist/:idplaylist yang berguna untuk mengubah nama playlist yang memiliki id sesuai dengan parameter yang diinputkan. Response dari request ini adalah JSON Array yang berisi status dan juga message berhasil / gagal. Jika id playlist tidak ditemukan, berikan message “Id Playlist tidak ditemukan”.

CATATAN: DILARANG MENGGUNAKAN DATABASE (MYSQL). 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
6/3/0	Endpoint POST /registerUser berjalan dengan benar
3/0	Endpoint POST /uploadSong berjalan dengan benar
6/3/0	Endpoint POST /createPlaylist berjalan dengan benar
3/0	Endpoint GET /searchSongByKeyword berjalan dengan benar
3/0	Endpoint DELETE /deleteSong/:judul berjalan dengan benar
6/3/0	Endpoint GET /getPlaylist/:emailpengguna berjalan dengan benar
3/0	Endpoint PUT /updatePlaylist/:idplaylist berjalan dengan benar

Menyetujui

Mengetahui

Penyusun Soal

(Esther Irawati S., S.Kom, M.Kom)

(Grace Levina Dewi, M.Kom.)

(Adrian Ignatius)

Koordinator Kuliah

Koordinator Laboratorium

Asisten

