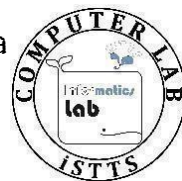




Institut Sains dan Teknologi Terpadu Surabaya
Jl. Ngagel Jaya Tengah 73 - 77, Surabaya 60284
Telp. (031) 5027920 Fax. (031) 5041509



Laboratorium	: L-204	Praktikum	: Service Oriented Architecture
Waktu	: 15.45-17.45	Jurusan	: S1 Informatika
Minggu Ke	: 6	Tanggal	: 13 April 2020
Materi	: Authentication & Authorization	Jenis Soal	: Materi dan Tugas

MATERI (TOTAL: 40)

CATATAN: GUNAKAN MYSQL SEBAGAI MEDIA PENYIMPANAN DATA. GUNAKAN JUGA LIBRARY JSONWEBTOKEN UNTUK MENGERJAKAN MATERI MINGGU INI.

Buatlah sebuah web service yang digunakan untuk sebuah aplikasi download E-Book Open source sederhana dimana terdapat 2 role yakni admin dan juga user. User sendiri dibagi menjadi 2 tipe, yakni user biasa dan juga member. Nantinya ada beberapa perbedaan tentang hal yang dapat diakses oleh user biasa dan juga member. Berikut ini adalah kriteria yang harus dipenuhi pada web service yang dibuat:

1. Buatlah sebuah function POST /api/registerUser yang digunakan untuk mendaftar akun. Data yang dikirim pada fungsi ini adalah {email user, nama user, tipe user (0 untuk admin, 1 untuk user biasa, 2 untuk member), password user}. Berikan pengecekan bahwa field terisi dan username belum digunakan sebelumnya. Response yang dikembalikan oleh request ini adalah message yang berisi apakah registrasi berhasil atau gagal, jika gagal berikan error message.
2. Buatlah sebuah function POST /api/loginUser yang digunakan untuk melakukan login user. Data yang dikirimkan pada fungsi ini adalah {email user, password user}. **Manfaatkan JWT untuk melakukan sign dengan sebuah secret "freebooks" dan waktu expire nya adalah 90 menit (1 jam), data yang dikirim pada fungsi jwt sign adalah {username, tipe user}.** Apabila login gagal, response yang dikembalikan adalah message "Username atau password salah", apabila login berhasil maka response yang dikembalikan adalah token JWT yang dihasilkan.
3. Buatlah sebuah function POST /api/addBook yang digunakan untuk menambah data buku yang ada pada sistem. Data yang dikirim pada fungsi ini adalah {judul buku, penulis, deskripsi buku, jenis buku (novel, komik, biografi, ensiklopedi), tahun terbit, jumlah download (defaultnya adalah 0)}. **Berikan pengecekan bahwa ada token yang disertakan dan juga token tersebut valid dan juga yang dapat mengakses resource ini hanyalah user yang bertipe admin saja.** Apabila add buku berhasil, response yang dikembalikan oleh request ini adalah kode buku yang merupakan autogenerate dengan format "B + 2 digit kode jenis buku (NV untuk novel, CM untuk komik, BG untuk biografi) + 3 digit nomor urut". Apabila add buku gagal, response akan mengembalikan pesan error.
4. Buatlah sebuah function GET /api/getBooks?tahun={keyword} untuk mendapatkan data buku yang ada pada sistem. Parameter search digunakan untuk mencari buku yang judulnya mengandung keyword yang diinputkan, begitu pula untuk parameter tahun. **Berikan pengecekan bahwa parameter tahun bersifat optional dan hanya dapat digunakan oleh user yang bertipe sebagai member saja, sehingga user biasa akan mendapatkan data semua buku tanpa bisa menginputkan parameter tahun. Berikan pengecekan bahwa ada token yang disertakan dan juga token tersebut valid.** Response yang dikembalikan oleh request ini adalah JSON Array yang berisi data buku yang sesuai pencarian.

Contoh response GET /api/getBooks (tanpa parameter)

```
[
  {
    "kode_buku": "BCM001",
    "judul_buku": "Nobita di Negeri Wan Nyan",
    "penulis": "Fujiko Fujio",
    "deskripsi_buku": "Ikuti petualangan Doraemon & Nobita di Negeri Wan Nyan!",
    "type_buku": "Komik",
    "tahun_terbit": "2009",
    "jumlah_download": 5
  },
  {
    "kode_buku": "BBG002",
    "judul_buku": "Biografi Gus Dur",
    "penulis": "Greg Barton",
    "deskripsi_buku": "Buku yang menceritakan tentang perjuangan seorang Gus Dur sehingga menjadi tokoh panutan masyarakat",
    "type_buku": "Biografi",
    "tanggal_terbit": "2011",
    "jumlah_download": 10
  }
]
```

Contoh response GET /api/getBooks?tahun=2009

```
[
  {
    "kode_buku": "BCM001",
    "judul_buku": "Nobita di Negeri Wan Nyan",
    "penulis": "Fujiko Fujio",
    "deskripsi_buku": "Ikuti petualangan Doraemon & Nobita di Negeri Wan Nyan!",
    "type_buku": "Komik",
    "tahun_terbit": "2009",
    "jumlah_download": 5
  },
  {
    "kode_buku": "BNV001",
    "judul_buku": "Negeri 5 Menara",
    "penulis": "Ahmad Faudi",
    "deskripsi_buku": "Novel yang menceritakan lima sahabat yang sedang mondok di sebuah pesantren, kemudian bertemu kembali ketika mereka sudah beranjak dewasa",
    "type_buku": "Novel",
    "tanggal_terbit": "2009",
    "jumlah_download": 10
  }
]
```

5. Buatlah sebuah function POST/api/downloadBook yang dapat digunakan oleh user baik user biasa maupun member untuk mendownload buku yang ada pada sistem. Data yang dikirim pada fungsi ini adalah {email user, kode buku, dan juga tanggal download}. Setiap kali buku berhasil di download, maka tambahkan field “jumlah download” pada buku yang bersangkutan. Berikan pengecekan bahwa kode buku yang didownload ada. **Berikan pengecekan bahwa free user hanya dapat mendownload sebanyak 2 buku saja, sedangkan premium user bebas mendownload sebanyak apapun, berikan pengecekan juga bahwa ada token yang disertakan dan token tersebut valid.** Response yang dikembalikan oleh request ini adalah message yang berisi berhasil tidaknya proses download.
6. Buatlah sebuah function DELETE/api/deleteBooks/:kodebuku yang digunakan untuk menghapus data buku yang ada pada sistem. Berikan pengecekan bahwa buku yang bisa dihapus hanyalah buku yang jumlah downloadnya dibawah 10. **Berikan pengecekan juga bahwa ada token yang disertakan dan token tersebut valid dan juga yang dapat mengakses resource ini hanyalah user yang bertipe admin saja.** Response yang dikembalikan oleh request ini adalah message yang berisi berhasil tidaknya data buku tersebut dihapus.

Berikut ini adalah struktur table yang digunakan:

```
CREATE TABLE `books` (
  `kode_buku` int(11) NOT NULL,
  `judul_buku` varchar(50) NOT NULL,
  `penulis` varchar(30) NOT NULL,
  `deskripsi_buku` int(50) NOT NULL,
  `kode_jenis_buku` varchar(2) NOT NULL,
  `tahun_terbit` varchar(4) NOT NULL,
  `jumlah_download` int(11) NOT NULL
)

CREATE TABLE `user` (
  `id_user` int(11) NOT NULL AUTO_INCREMENT,
  `email_user` varchar(50) NOT NULL,
  `password_user` varchar(30) NOT NULL,
```

```

        `nama_user` varchar(30) NOT NULL,
        `tipe_user` int(1) NOT NULL,
        PRIMARY KEY (`id_user`)
    )

CREATE TABLE `download_books` (
    `kode_buku` int(11) NOT NULL,
    `email_user` varchar(50) NOT NULL
)

```

CATATAN: HIGHLIGHT KRITERIA YANG DIKERJAKAN. JIKA TIDAK MENGUMPULKAN HIGHLIGHT MAKA NILAI DIV 2.

MATERI : 40

SCORE	KRITERIA
0/3/4/5/6	Endpoint register user berjalan dengan benar
0/3/4/5/6	Endpoint login user berjalan dengan benar *3 apabila response tidak mengembalikan jwt token & tidak ada expire token *4 apabila response mengembalikan jwt token namun tidak ada waktu expire
0/3/4/5/6	Endpoint add book user berjalan dengan benar
0/2/4/6/8	Endpoint get book user berjalan dengan benar *2 apabila tidak ada pengecekan token *4 apabila ada pengecekan token dan parameter tahun berfungsi, namun tidak ada pengecekan Member *6 apabila ada pengecekan token dan parameter tahun berfungsi + pengecekan member, namun Format response tidak sama Tidak sama *8 apabila memenuhi semua kriteria
0/2/4/6/8	Endpoint download book user berjalan dengan benar *2 apabila tidak ada pengecekan token & data masuk database *4 apabila ada pengecekan token & data masuk database, namun tidak mengubah field jumlah Download pada buku *6 apabila ada pengecekan token & data masuk database, mengubah field jumlah Download pada buku namun tidak ada pengecekan jumlah download (free user dan premium) Tidak sama *8 apabila memenuhi semua kriteria
0/3/4/5/6	Endpoint delete book user berjalan dengan benar

TUGAS (TOTAL = 30)

Buatlah sebuah web service yang digunakan untuk sebuah aplikasi online classroom sederhana dimana terdapat 2 user yakni student dan juga teacher. Berikut ini adalah kriteria yang harus ada pada web service yang dibuat:

1. Buatlah sebuah function POST /api/registerUser yang digunakan untuk mendaftar akun. Data yang dikirim pada fungsi ini adalah {email user, nama user, tipe user (0 untuk student, 1 untuk teacher), password user}. Berikan pengecekan bahwa field terisi dan username belum digunakan sebelumnya. Response yang dikembalikan oleh request ini adalah message yang berisi apakah registrasi berhasil atau gagal, jika gagal berikan error messagenya.
2. Buatlah sebuah function POST /api/loginUser yang digunakan untuk melakukan login user. Data yang dikirimkan pada fungsi ini adalah {email user, password user}. **Manfaatkan JWT untuk melakukan sign dengan sebuah secret “9 digit NRP anda” dan waktu expire nya adalah 60 menit (1 jam), data yang dikirim pada fungsi jwt sign adalah {username, tipe user}.** Apabila login gagal, response yang dikembalikan adalah message “Username atau password salah”, apabila login berhasil maka response yang dikembalikan adalah token JWT yang dihasilkan.
3. Buatlah sebuah function POST/api/createClass yang digunakan untuk membuat sebuah class yang nantinya dapat diikuti oleh student. Data yang dikirimkan pada fungsi ini adalah {nama kelas, deskripsi kelas, dan juga grade

kelas (SD,SMP,SMA)}. **Berikan pengecekan bahwa ada token yang disertakan dan juga token tersebut valid. Berikan pengecekan juga bahwa yang dapat mengakses resource ini hanyalah user yang bertipe teacher saja.** Apabila create class berhasil, response yang dikembalikan oleh request ini adalah kode kelas yang merupakan autogenerate dengan format sebagai berikut:KE+3digit nomor urut. Apabila request ini gagal, response yang diberikan adalah pesan error.

4. Buatlah sebuah function GET/api/getAllClass/:emailuser yang digunakan untuk mendapatkan semua class yang dimiliki oleh user dengan tipe teacher. **Berikan pengecekan bahwa ada token yang disertakan dan juga token tersebut valid dan juga yang dapat mengakses resource ini hanyalah user yang bertipe teacher saja.**

```
[
  {
    "kode_kelas": "KE002",
    "nama_kelas": "Service Oriented Architecture",
    "murid": [
      {
        "email_user": "yoyo@xyz.com",
        "nama_user": "Yohanes Ivan"
      },
      {
        "email_user": "adrian@xyz.com",
        "nama_user": "Adrian Ignatius"
      },
      {
        "email_user": "robby@xyz.com",
        "nama_user": "Robby Darmawan"
      }
    ],
    "jumlah_murid": 3
  }
]
```

5. Buatlah sebuah function POST/api/joinClass yang digunakan untuk mengikuti sebuah kelas yang sudah ada sebelumnya. Data yang dikirim pada fungsi ini adalah {email user, kode kelas}. **Berikan pengecekan bahwa ada token yang disertakan dan juga token tersebut valid. Berikan pengecekan juga apakah kode kelas yang dimaksud ada.** Response yang dikembalikan oleh request ini adalah message apakah join class berhasil atau tidak.
6. Buatlah sebuah function POST/ api/createAssignment yang digunakan oleh teacher untuk memberikan tugas pada kelas tertentu yang dimiliki olehnya. Data yang dikirimkan oleh fungsi ini adalah {kode_kelas, nama tugas, deskripsi tugas, dan juga deadline tugas}. **Berikan pengecekan bahwa ada token yang disertakan dan juga token tersebut valid dan juga yang dapat mengakses resource ini hanyalah user yang bertipe teacher saja. Berikan juga pengecekan bahwa kode kelas ada dan memang diajar oleh teacher yang bersangkutan.** Apabila create Assignment berhasil, response yang dikembalikan oleh request ini adalah kode tugas yang merupakan autogenerate dengan format sebagai berikut:TU+3digit nomor urut.
7. Buatlah sebuah function GET / api/getAssignment/:emailuser?search={keyword} yang dapat digunakan oleh student untuk mendapatkan semua assignment(tugas) pada kelas yang dia ikuti. Parameter search bersifat optional sehingga jika tidak disertakan, maka tampilkan semua tugas, namun apabila parameter search diberikan maka tampilkan tugas yang diberikan pada kelas yang sesuai dengan nama kelas tersebut . **Berikan pengecekan bahwa ada token yang disertakan dan juga token tersebut valid dan juga yang dapat mengakses resource ini hanyalah user yang bertipe student saja.**

Contoh Response tanpa parameter:

```
[
  {
    "kode_tugas": "TU001",
    "nama_kelas": "IWWW",
    "nama_tugas": "Desain website sederhana menggunakan Materialize",
    "deadline_tugas": "13/04/2020",
    "hari_menuju_deadline": "Sudah melewati batas deadline 2 hari lalu"
  },
  {
    "kode_tugas": "TU004",
    "nama_kelas": "Matematika",
    "nama_tugas": "Soal Latihan Trigonometri Dasar",
    "deadline_tugas": "17/04/2020",
    "hari_menuju_deadline": "2 hari lagi"
  },
  {
    "kode_tugas": "TU006",
    "nama_kelas": "IWWW",
    "nama_tugas": "Membuat blog sederhana menggunakan wordpress",
    "deadline_tugas": "24/04/2020",
    "hari_menuju_deadline": "9 hari lagi"
  }
]
```

Contoh Response dengan parameter search "IWWW":

```
[
  {
    "kode_tugas": "TU001",
    "nama_kelas": "IWWW",
    "nama_tugas": "Desain website sederhana menggunakan Materialize",
    "deadline_tugas": "13/04/2020",
    "hari_menuju_deadline": "Sudah melewati batas deadline 2 hari lalu"
  },
  {
    "kode_tugas": "TU006",
    "nama_kelas": "IWWW",
    "nama_tugas": "Membuat blog sederhana menggunakan wordpress",
    "deadline_tugas": "24/04/2020",
    "hari_menuju_deadline": "9 hari lagi"
  }
]
```

CATATAN: WAJIB MENGUMPULKAN HIGHLIGHT KRITERIA DAN JUGA POSTMAN COLLECTION UNTUK MEMUDAHKAN PROSES PEMERIKSAAN TUGAS.

TUGAS: 30

SCORE	KRITERIA
0/2/3/4	Endpoint register user berjalan dengan benar
0/2/3/4	Endpoint login user berjalan dengan benar *2 apabila response tidak mengembalikan jwt token & tidak ada expire token *3 apabila response mengembalikan jwt token namun tidak ada waktu expire
0/2/3/4	Endpoint create class berjalan dengan benar
0/2/3/4	Endpoint get class berjalan dengan benar
0/2/3/4	Endpoint join class berjalan dengan benar
0/2/3/4	Endpoint create assignment user berjalan dengan benar
0/3/4/5/6	Endpoint get assignment user berjalan dengan benar *3 apabila tidak ada pengecekan token & user student *4 apabila ada pengecekan token & user, namun parameter optional search tidak berfungsi *5 apabila ada pengecekan token & user, parameter search berfungsi, namun format response Tidak sama *6 apabila memenuhi semua kriteria

Menyetujui

Mengetahui

Penyusun Soal

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

(Grace Levina Dewi, M.Kom.)

(Adrian Ignatius)

Koordinator Kuliah

Koordinator Laboratorium

Asisten