

Soal ujian teknis - Backend

- 1 [Ketentuan](#)
- 2 [Penjadwalan kamar operasi](#) STRUKTUR DATA ALGORITMA
- 3 [Penggajian](#) MODULARITY DESIGN PATTERN
- 4 [Mendeteksi alergi obat](#) STRUKTUR DATA SEARCHING
- 5 [Memperbaharui data secara otomatis dan terjadwal](#) BASIC PROGRAMMING

Ketentuan

- Kandidat dapat mengerjakan tugas ini pada *word processing application* (misalnya *microsoft word*) atau *text processing application* (misalnya *notepad++*).
- Kandidat **tidak boleh** menggunakan database dalam mengerjakan tugas ini.
- Jawaban disampaikan dalam 2 format:
 - a. tulisan dalam bentuk code (bebas menggunakan bahasa pemrograman apa saja)
 - b. Video running program sesuai dengan skenario yang dipersyaratkan. Unggah video tersebut ke cloud storage (misalnya youtube, google drive, dll), kemudian share link video tersebut pada lembar jawaban saudara.

Penjadwalan kamar operasi

STRUKTUR DATA ALGORITMA

Buatlah kode **REST API** untuk penjadwalan kamar operasi, dimana jarak antara satu operasi dengan operasi yang lain minimal 2 jam. Jika seorang dokter ingin booking kamar operasi, dokter tersebut harus menyampaikan tanggal dan jam kapan dia mau menggunakan kamar operasi, dan estimasi durasi operasi tersebut. endpoint untuk REST API adalah berikut ini:

```
1 localhost:8080/bookingkamaroperasi/{bookingdate}/{durasi};
```

response dari API tersebut ada dalam body, berisi hanya 2 kemungkinan saja: true atau false.

Skenario

Jalankan service REST API yang sudah dibuat, kemudian panggil bookingkamaroperasi berturut-turut seperti dibawah ini:

1. Ada seorang dokter yang meminta *booking* kamar operasi, untuk tanggal 01/01/2022 jam 10:00 selama 2 jam. Karena belum ada yang *booking*, maka permintaan booking ini akan **berhasil**, hasil pemanggilan method bookingkamaroperasi akan bernilai true.
2. Kemudian ada dokter lain yang meminta *booking* kamar operasi, untuk tanggal 01/01/2022 jam 15:00 selama 2 jam. Booking pertama ada pada pukul 10, dan durasi booking pertama 2 jam, maka operasi berikutnya baru boleh paling cepat pada pukul 14:00. Karena permintaan booking untuk pukul 15:00, maka permintaan booking ini akan **berhasil**, hasil pemanggilan method bookingkamaroperasi akan bernilai true.
3. Kemudian ada dokter lain yang meminta *booking* kamar operasi, untuk tanggal 01/01/2022 jam 18:00 selama 2 jam. Booking pertama ada pada pukul 10, dan durasi booking pertama 2 jam, maka operasi berikutnya baru boleh paling cepat pada pukul 14:00. Kemudian ada booking kedua pada pukul 15:00 selama 2 jam, maka operasi berikutnya paling cepat pada pukul 19:00. Karena permintaan booking ini untuk pukul 15:00, maka permintaan booking ini akan **gagal**, hasil pemanggilan method bookingkamaroperasi akan bernilai false.
4. Kemudian ada dokter lain yang meminta *booking* kamar operasi, untuk tanggal 01/01/2022 jam 07:00 selama 2 jam. Booking pertama ada pada pukul 10, dan karena jarak antar operasi adalah 2 jam, maka operasi sebelumnya harus selesai pada jam 08:00. Karena permintaan booking ini untuk pukul 07:00 dan durasi 2 jam, maka selesainya operasi pada jam 09:00, maka permintaan booking ini akan **gagal**, hasil pemanggilan method bookingkamaroperasi akan bernilai false.

Penggajian

MODULARITY DESIGN PATTERN

Dalam perusahaan global, dimana karyawan ada di berbagai negara, maka besar kemungkinan untuk membayar gaji pegawai di beberapa negara, dimana tiap-tiap negara memiliki regulasi penggajian yang berbeda beda. Katakanlah kita akan melakukan pembayaran gaji untuk karyawan di Indonesia dan di Vietnam. Gaji karyawan di Indonesia dalam Rupiah (IDR) dan karyawan di Vietnam dalam Dong (VND).

Indonesia

Di Indonesia, dalam menghitung pajak penghasilan, digunakan rumus berikut:

$\text{penghasilan netto} = \text{semua penghasilan} - \text{PTKP}$

PTKP = penghasilan tidak kena pajak, sesuai dengan tarif berikut:

Status perkawinan	Simbol	Tarif PTKP (dalam juta IDR)
Belum kawin	TK	25
Kawin dan belum punya anak	K0	50
Kawin dan sudah punya anak	K1	75

Tarif pajak penghasilan tahunan:

Rentang Penghasilan netto (dalam juta IDR)	Tarif pajak
0-50	5%
50-250	10%
>250	15%

Vietnam

Di Vietnam, dalam menghitung pajak penghasilan, digunakan rumus berikut:

$\text{penghasilan netto} = \text{semua penghasilan} - \text{asuransi} - \text{PTKP}$

PTKP = penghasilan tidak kena pajak, sesuai dengan tarif berikut:

Status perkawinan	Tarif PTKP (dalam juta VND)
Belum kawin	15
Sudah kawin	30

Tarif pajak penghasilan tahunan:

Rentang Penghasilan netto (dalam juta VND)	Tarif pajak
0-50	2.5%
>50	7.5%

Tugas

Buatlah **REST API** untuk menghitung gaji, dengan http POST endpoint seperti dibawah ini:

`localhost:8080/hitunggaji`

dengan request body berbentuk json seperti ini:

```
1 {
2   "employee": {},
3   "komponengaji": [
4     {},
5     {}
```

```
6   ]
7   }
```

Pada json diatas, employee masih kosong, silahkan diimplementasikan struktur data yang cocok untuk merepresentasikan object employee.

Pada json diatas, komponengaji adalah json array yang masih kosong. disini dipakai json array karena komponen gaji bisa lebih dari satu. silahkan diimplementasikan struktur data yang cocok untuk merepresentasikan object komponengaji.

Skenario

Jalankan REST API service yang sudah dibuat, kemudian lakukan pemanggilan berikut:

Indonesia

seorang karyawan di **Indonesia** yang sudah menikah dan memiliki anak dengan gaji bersih 30 juta perbulan akan mendapatkan pajak berikut:

penghasilan netto pertahun = (30 juta x 12 bulan) - 75 = 360-75 = 285

pajak penghasilan:

layer	rumus	pajak (dalam juta IDR)
50	50 * 5%	2.5
50-250	(285-50) * 15%	35.25

total pajak setahun = 2.5+32.25 = 37.75 juta

total pajak bulan ini = 37.75 / 12 = 3.146 juta

Tentukan request body yang merepresentasikan skenario ini, kemudian panggil API `hitunggaji` , maka response body adalah 3.146.000.

Vietnam

seorang karyawan di **Vietnam** yang sudah menikah dengan gaji bersih 30 juta perbulan dan tunjangan asuransi 1 juta per bulan, akan mendapatkan pajak berikut:

penghasilan netto pertahun = (30 juta x 12 bulan) - (1 juta x 12 bulan) - 30 = 360-12-30 = 318

pajak penghasilan:

layer	rumus	pajak (dalam juta VND)
50	50 * 2.5%	1.25
>50	(285-50) * 7.5%	20.1

total pajak setahun = 1.25+20.1 =21.35 juta

total pajak bulan ini = 21.35 / 12 = 1.779 juta

Tentukan request body yang merepresentasikan skenario ini, kemudian panggil API `hitunggaji` , maka response body adalah 1.779.000.

Mendeteksi alergi obat STRUKTUR DATA SEARCHING

Beberapa pasien memiliki alergi terhadap obat tertentu, misalnya obat yang mengandung *aspirin*, *ibuprofen*, dan *naproxen*.

Tugas

Buatlah **REST API** yang dapat mengevaluasi potensi alergi obat pada seorang pasien atas sebuah resep.

Code snippet

Berikut endpoint API dari service tersebut:

```
localhost:8080/validasialergiobat
```

dengan request body berbentuk json seperti ini:

```
1 {
2   "pasien": {},
3   "resep": [
4     {},
5     {}
6   ]
7 }
```

Pada json diatas, pasien masih kosong, silahkan diimplementasikan struktur data yang cocok untuk merepresentasikan object pasien.

Pada json diatas, resep adalah json array yang masih kosong. disini dipakai json array karena resep bisa lebih dari satu. silahkan diimplementasikan struktur data yang cocok untuk merepresentasikan object resep.

Yang mendapat perhatian lebih

Satu obat, dapat memiliki beberapa kandungan sekaligus. Satu obat boleh jadi memiliki 1 kandungan utama. Contohnya **Paratusin** yang memiliki kandungan utama paracetamol 125 mg, disamping kandungan lainnya terdiri dari pseudoepedrid 7.5 mg, noscapine 10 mg, ctm 0.5 mg, guaafenisin 25 mg, succus liquiritae 125 ethanol 10 %.

Yang tidak boleh dilakukan

Mendelegasikan **search problem** pada **database**. Milsalnya, untuk mencari obat dengan yang memiliki kandungan ibuprofen, maka dicari dengan sql berikut:

```
1 select obat from obat where lower(kandungan) like '%ibuprofen%'
```

Search diatas **dilarang** dilakukan, sebab kami ingin mengetahui kemampuan logika yang dimiliki kandidat. Untuk hal itu, kandidat bisa menggunakan *array* atau *collection* untuk menyimpan data obat dan kandungannya, kemudian menggunakan trik pemrograman untuk mendapatkan data yang dimaksud.

Yang harus diperhatikan

1. Kebersihan kode. buatlah kode yang sebersih mungkin dengan menerapkan *SOLID principle*, dan *design pattern* yang mungkin bermanfaat.
2. Performa. Perhatikan jumlah loop/iterasi yang digunakan. Bayangkan jika katalog obat sudah mencapai 2 juta jenis obat, dan 500 ribu merek obat.

Skenario

Misalnya pasien anak atas nama Bejo, mengidap alergi **ibuprofen**. Setelah konsultasi rawat jalan, dokter memberi resep berupa: Proris sirup 60ml dan Paratusin sirup 60ml. Maka ketika `validasiAlergiObat` dipanggil, dia akan me-return collection `Obat` dengan panjang 1, berisi `Proris`. Hal ini dikarenakan **Proris** mengandung **ibuprofen** dimana Bejo alergi terhadapnya.

Tentukan request body yang merepresentasikan skenario ini, kemudian panggil API `validasialergiobat`, maka response body adalah

```
1 {
2   "resep": [
3     {"obat": "Proris"}
4   ]
5 }
```

```
4   ]
5 }
```

Memperbaharui data secara otomatis dan terjadwal BASIC PROGRAMMING

Dalam dunia medis, umur merupakan data yang penting. Tidak seperti orang awam yang menyebut umur dalam *magnitude* tahun, dalam dunia medis, umur sangat detail yakni terdiri dari Tahun Bulan dan Hari.

Misalnya seorang pasien yang lahir tanggal 28 Oktober 1983, maka pada tanggal 3 Oktober 2022 umurnya 38 tahun 11 bulan dan 6 hari.

Tugas

Buatlah REST API menghitung **umur** pada data pasien.

Endpoint

```
localhost:8080/calculateage/{dateofbirth}
```

yang memiliki response body

```
1 {
2   "umur": {
3     "year": 1,
4     "month": 1,
5     "day": 1
6   }
7 }
```

Skenario

Tanggal Lahir	REST API dipanggil pada tanggal	response body jika diubah dalam string berformat yy/mm/dd
28-10-1983	03-10-2022	38/11/06
	04-10-2022	38/11/07
13-11-1982	03-10-2022	39/10/20
	04-10-2022	39/10/21