

## TUTORIAL 10 : Melakukan request ke Web Service pada React Js App

### Jawaban Soal :

1. Variable cors berisi url cors anywhere. Yang memungkinkan pengiriman request dari satu host ke host lainnya (heroku). Dimana penggunaan cors anywhere adalah sebagai pihak ketiga yang menampilkan data kepada user. Jadi dalam kasus ini, user meminta data kepada SI-Appointment, dan yang memberikan hasil data kepada user.  
Jika prefix `{cors}` dihilangkan, proses loading pada data menjadi sangat lama. Tidak diketahui apakah data akan bisa diperlihatkan atau tidak karena status loading selalu false. Dan jika dilihat pada console, terdapat error tidak bisa load karena tidak memenuhi CORS policy.
2. `Name.pop()` berarti melakukan transisi ulang dan melepaskan nilai name yang dipegang saat ini. `Name.reduce()` melakukan pengurangan (mengurangi) array menjadi single value. Dengan syntax **`array.reduce(function(total, currentValue, currentIndex, arr), initialValue)`**.  
Apabila code tersebut diganti dengan `dataJson[key] = val` maka fungsi update tidak dapat dilakukan karena `response.status` tidak sama dengan 200. Dengan kata lain fungsi tidak jalan.

### Cara implementasi soal latihan :

1. Menampilkan seluruh staf farmasi
  - Melakukan hal yang sama dengan pada `getAllPasien`, dengan pemanggilan api SI-Appointment <http://si-appointment.herokuapp.com/api/1/getAllStaffFarmasi>
  - File baru yang ditambah adalah :
    - `DaftarStaffFarmasiRow` (pada folder components)
      - Digunakan untuk mengisi data tables yang ada dengan data dari staf farmasi. Akan mengembalikan data berbentuk map yang akan mengembalikan list staf farmasi
    - `DaftarStaffFarmasi` (pada folder screens)
      - Digunakan untuk memanggil table container yang telah dibuat yang akan menampilkan daftar staf farmasi. Dengan return

tableContainer, header Nama dan jenis, serta state yang berisi listStaf.

- Method yang ditambah
  - Pada Appointment di folder utils => method getAllStafFarmasi
    - Perbedaan dengan getAllPasien adalah url yang di fetch. Untuk response dll, sama karena data yang di ambil berbentuk sama.
- 2. Melakukan tambah hasil lab pada pasien
  - Melakukan hal yang relatif sama dengan updatePasien dengan pemanggilan api SI-Appointment <http://si-appointment.herokuapp.com/api/1/addLabResult>
  - File baru yang ditambah adalah :
    - FormAddLabResult (pada folder containers)
      - Yang berisi form yang digunakan untuk menambahkan hasil lab pada masing-masing pasien. Pada form ini akan ada input id pasien yang bertipe hidden. Dan input lain yakni jenis, hasil, dan tanggalPengajuan.
    - AddLabResult (pada folder screens)
      - Berisi relatif sama dengan UpdatePasien.js yang berbeda hanya form yang di import, dan return yang diberikan adalah form add Lab result. Akan berfungsi untuk mengambil data yang telah diisikan pada form dan melakukan method handleFormSubmit yang akan dijalankan ketika user menekan tombol submit pada form (di trigger pada return).
  - Method yang ditambah
    - Pada Appointment di folder utils => method addLabResult
      - Method dengan parameter requestBody yang fetch url addlabresult dan melakukan stringify body, dan inisiasi response.
- 3. Secara keseluruhan yang dilakukan pada semua tahap latihan adalah penambahan route path pada app.js untuk fitur-fitur baru. Dan menambah kan navbar pada pelaksanaan latihan menampilkan daftar staf farmasi.

*-- Terima kasih bg jik, semoga hari bg jik menyenangkan 😊 --*