2022

PBW

M08 – AJAX

Silakan mengikuti petunjuk yang ada.

Raymond Chandra Putra, S.T., M.T.
IF UNPAR
6/2/2022

PETUNJUK PENGERJAAN

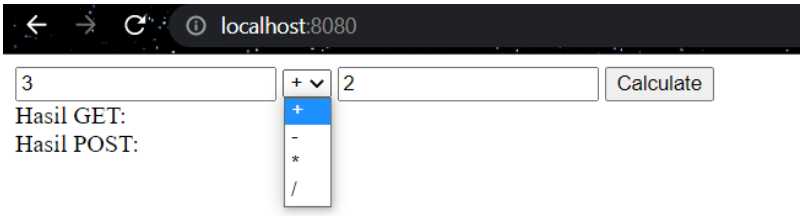
1. Kumpulkan hasil pengerjaan dengan nama M08xxyyy.zip.
2. Modul ini menggunakan template yang disediakan.
3. Jawablah pertanyaan jika text berwarna oranye pada sebuah txt.

Intro

AJAX (Asynchronous JavaScript And XML) merupakan suatu teknik yang memungkinkan sisi klien mampu mengakses sisi server tanpa mengganti halaman. Penggunaan AJAX ini dapat menambah kenyamanan pengguna dan mempermudah beberapa hal, contohnya jika kita membutuhkan data yang tidak disediakan ketika halaman tersebut dibuka. Untuk mengerjakan modul ini, bukalah template yang sudah disediakan untuk mengerjakan modul ini. Pada modul ini teknik AJAX yang digunakan adalah fetchAPI.

Bagian 1

Pada template yang disediakan sudah terdapat halaman HTML yang jika dibuka, maka akan menampilkan tampilan sebagai berikut:



Gambar 1 – Tampilan dari modul ini.

Silahkan mengikuti petunjuk pengerjaan berikut:

1. Pelajari terlebih dahulu kode yang sudah disediakan pada template untuk mengetahui alur kerja program. Beberapa kode program dibuat sebagai komentar agar tidak menyebabkan error untuk fungsi yang belum ada.

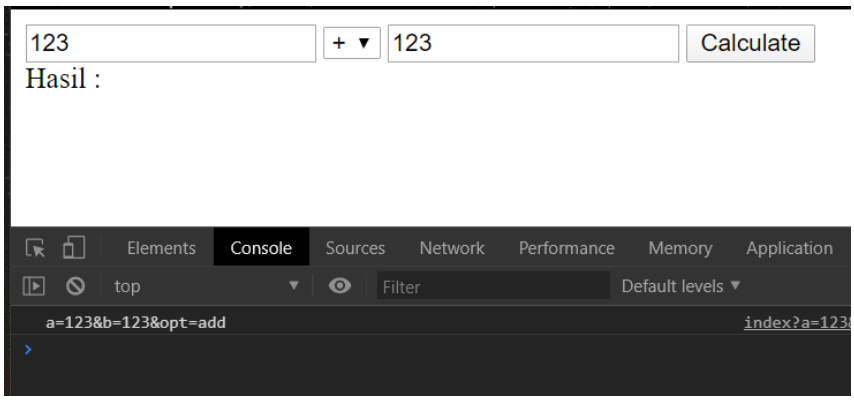
2. Bukalah file app.js pada folder public.

   a. Apa yang terjadi jika tombol submit ditekan?

   b. Apa yang terjadi jika kode event.preventDefault() pada fungsi onSubmit dihilangkan? Jelaskan.

   c. Apa isi dari variable formElements? (cek menggunakan console)

3. Pada method onSubmit, buatlah object input yang berisi semua input dari form. Setelah itu, object tersebut diubah menjadi string dengan format url (lihat gambar). Hint: manfaatkan method encodeURL yang sudah disediakan. Jika variable tersebut ditampilkan pada console, maka hasilnya akan menampilkan sebagai berikut:



Gambar 2 – Isi dari variable input.

4. Pada folder model, sudah tersedia kelas Calculator yang dapat digunakan langsung.

5. Pada folder controller, terdapat file controller.js yang berfungsi memanfaatkan kelas Calculator di nomor 4 dan mempunyai 2 buah fungsi (sebelum challenge) yaitu calculate_get dan calculate _post. Fungsi utama dari kedua fungsi ini sama yaitu menerima input dari user lalu melakukan perhitungan dengan memanggil method yang berada di kelas Calculator. Pada bagian ini, implementasikan dahulu method GET.

   a. Apa parameter dan return value dari kedua fungsi ini?

6. Challenge (opsional): Implementasikan perintah pada nomor 5 tanpa menggunakan syntax if / switch.

7. Kembali ke javascript, tambahkanlah pemanggilan AJAX pada fungsi onSubmit dengan kode berikut:

```
//1
const promise = fetch('calculate?'+input);
promise.then(function(response){
    console.log(response.status);
    console.log(response.text());
});
```

Gambar 3

Fungsi ini akan memanggil fungsi calculate_get di controller yang sudah dibuat. Lalu jalankan fungsi tersebut. Lakukan segala penyesuaian agar kode tersebut dapat berjalan.

   a. Apa tipe data dari variable promise?

   b. Apa hasil keluaran kedua baris kode tersebut pada console? Jelaskan.

8. Untuk melihat hasil teksnya, maka kita perlu mengubah kode tersebut menjadi seperti ini:

```
//2
const promise2 = fetch('calculate?'+input);

const request = promise2.then(function(response){
    //console.log(response.status);
    return response.text();
});

request.then(function(text){
    console.log(text);
});
```

Gambar 4.

a. Dengan perubahan ini, cek kembali console log yang dihasilkan. Jelaskan perbandingan hasil ini dengan 7b.

9. Ubahlah fungsi diatas dengan menggunakan promise chaining.

```
//4
fetch('calculate?'+input).then(onSuccess).then(showResult);
```

10. Implementasikan method onSuccess dan showResult. Pada method showResult ubahlah agar kode tersebut mengubah text pada element HTML (span result).

Bagian 2 POST + JSON

Pada bagian kedua ini akan dicoba pemanggilan AJAX dengan method POST menggunakan fetchAPI. Selain itu kita juga akan digunakan format JSON. Secara default, method fetch yang digunakan adalah method GET, sehingga untuk mengubah menjadi method POST, pada fetchAPI harus ditambahkan konfigurasi init.

11. Kode pemanggilan fetchAPI untuk method POST adalah sebagai berikut:

```
//5
let init = {
    method: 'post',
    headers: {
        "Content-Type": "application/x-www-form-urlencoded"
    },
    body: input
};
fetch('calculate', init).then(onSuccess).then(showResult);
```

12. Sesuaikan kode pada PHP agar method ini dapat berjalan sesuai fungsi seharusnya. Gunakan method calculate_post pada controller.

a. Apa yang terjadi jika variable headers dihilangkan dari init?

13. Untuk mengubah format pengiriman menjadi JSON kita harus melakukan perubahan baik dari sisi klien, maupun server. Pada sisi klien kita perlu mengubah tipe headers dan data yang dikirim pun harus berupa format JSON.

a. Pada bagian Content Type menjadi format json.

b. Pada input (bagian body) diubah menjadi json dengan menggunakan JSON.stringify.

c. Pada sisi server menggunakan middleware express.json (Node versi 4.16).

d. Implementasi fungsi calculate_get.

e. Mengembalikan hasil dengan method res.json();

f. Hasil ditampilkan pada span result_json.

14. Jadi sekarang dengan perubahan ini, setiap kali tombol submit dipanggil maka akan dipanggil 2 buah AJAX untuk method GET dan POST. Lalu akan diupdate tampilannya pada bagian yang sudah disediakan.

15. Challenge (opsional): Tambahkan middleware untuk melakukan validasi pada sisi server. Jika input salah maka pada kedua span akan ditampilkan “Invalid Input”. Validasi yang diperiksa adalah:

    a. Memeriksa kedua input angka apakah berbentuk angka atau tidak (dapat diisi teks atau kosong).

    b. Memeriksa apakah opt termasuk dari 4 kategori tersebut (pengguna dapat mengubah dengan inspect element).

Seluruh pekerjaan disatukan pada sebuah zip dan dikumpulkan dengan nama M08xxyyy.zip.

= SELAMAT MENGERJAKAN =