

Coding Subuh #19

AJAX

Ahmad Muhardian

Agenda kita..

- Kenalan dengan AJAX
- XMLHttpRequest
- Fungsi fetch()
- Latihan

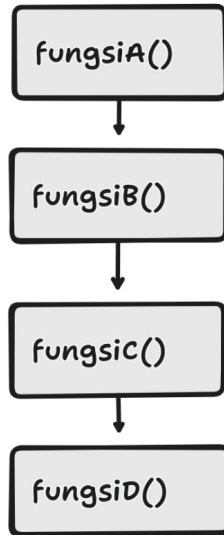
AJAX

Apa itu AJAX?

- AJAX = **A**synchronous **J**avascript **A**nd **X**ML
- Teknik untuk load konten atau data secara *asynchronous* dengan mengirim request ke server secara *background*.
- Sehingga konten dan data dalam web akan bisa diupdate tanpa harus reload.

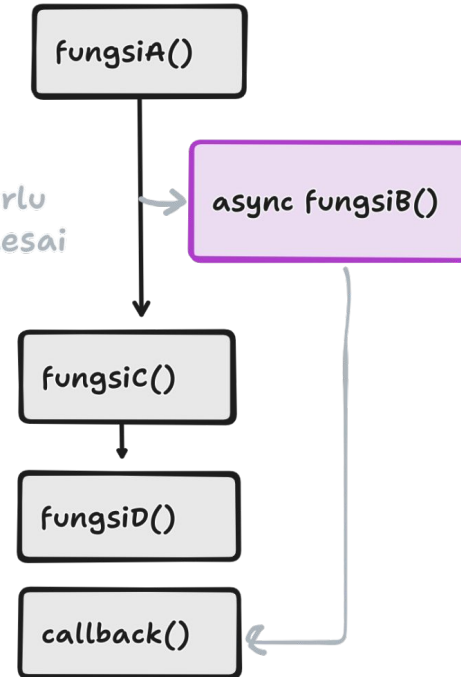
Synchronous

dikerjakan satu persatu
saling menunggu

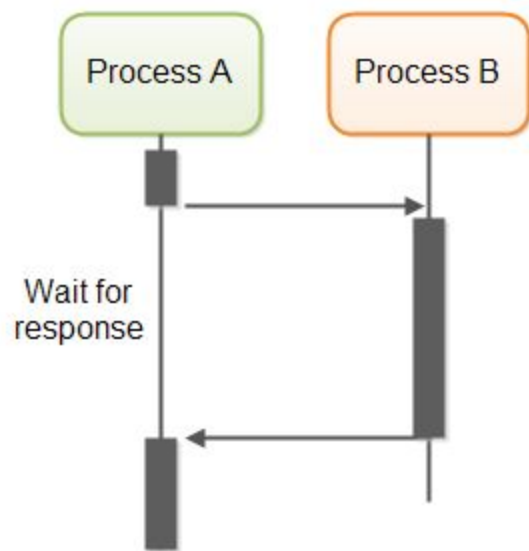


Asynchronous

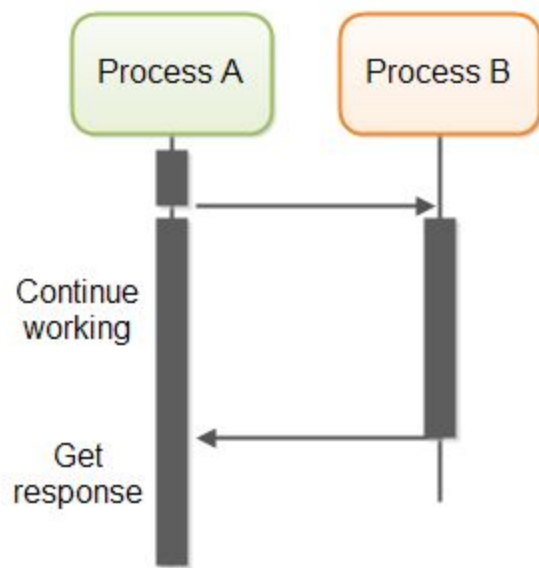
`fungsiC()` nggak perlu
nunggu `fungsiB` selesai
dikerjakan



Synchronous



Asynchronous



Ada 3 cara membuat AJAX request:

1. Menggunakan object XMLHttpRequest
2. Menggunakan fungsi `fetch()`
3. Menggunakan library seperti JQuery dan Axios

Menggunakan XMLHttpRequest

Cara menggunakan XMLHttpRequest

1. Membuat objek dari class XMLHttpRequest
2. Setup fungsi callback untuk tiap state/event dari object XMLHttpRequest, misalnya setelah data di load, saat proses request data, saat terjadi error.
3. Membuka koneksi dengan method `open()`
4. Mengirim data dengan method `send()`

```
// membuat object xhr
const xhr = new XMLHttpRequest();

// fungsi callback saat request selesai
xhr.onload = function () {
  const products = JSON.parse(xhr.responseText);
  console.log("PRODUCT", products);
  showProducts(products);
}

// membuka koneksi dengan metode GET
xhr.open("GET", URL, true);

// mengirim Data
xhr.send();
```

Properti di Object XMLHttpRequest

- `.responseText` -> berisi teks hasil respon dari server
- `.responseType` -> berisi jenis response (json, document, text, blob)
- `.upload` -> berisi object untuk handle upload file
- `.status` -> berisi code status dari request
- `.timeout` -> berisi batas waktu request

Properti lainnya cek di:

<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/>

Event di Object XMLHttpRequest

- **abort** -> event saat request dibatalkan
- **load** -> setelah request berhasil dan selesai
- **error** -> event saat request error
- **timeout** -> event saat waktu request habis
- **progress** -> event saat request berjalan
- **readystatechange** -> event saat terjadi perubahan state

Properti lainnya cek di:

<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/>

Method di Object XMLHttpRequest

- `abord()` -> untuk membatalkan request
- `open()` -> Membuka koneksi/request
- `send()` -> Buat kirim data ke server
- `setRequestHeader()` -> buat set request header

Properti lainnya cek di:

<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/>

Menggunakan Fungsi `fetch()`

Fungsi fetch()

- Fungsi fetch() adalah fungsi untuk melakukan HTTP Request
- Fungsi ini dijalankan secara asynchronous dengan promise
- Secara default fetch() menggunakan method GET
- Buat handle error, fungsi fetch dapat dijalankan dalam **try..catch** atau **then..catch**

```
async function logMovies() {  
  const response = await fetch("http://example.com/movies.json");  
  const movies = await response.json();  
  console.log(movies);  
}
```

Konfigurasi fetch()

HTTP Request akan
dilakukan ke URL ini

`fetch(URL, { ... })`

Object Konfigurasi

Konfigurasi fetch()

- **method** -> untuk menentukan method yang akan dipakai
- **headers** -> untuk menentukan HTTP Headers
- **body** -> untuk menentukan HTTP Body, biasanya berisi data yang akan dikirim
- **redirect** -> untuk handle redirect.
- **cache** -> untuk menentukan cache

Properti konfigurasi lainnya, cek di:

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

Contoh Konfigurasi fetch()

```
const response = await fetch(url, {  
  method: "POST", // *GET, POST, PUT, DELETE, etc.  
  mode: "cors", // no-cors, *cors, same-origin  
  cache: "no-cache", // *default, no-cache, reload, force-cache, only-if-cached  
  credentials: "same-origin", // include, *same-origin, omit  
  headers: {  
    "Content-Type": "application/json",  
    // 'Content-Type': 'application/x-www-form-urlencoded',  
  },  
  redirect: "follow", // manual, *follow, error  
  referrerPolicy: "no-referrer", // no-referrer, *no-referrer-when-downgrade, or  
  body: JSON.stringify(data), // body data type must match "Content-Type" header  
});
```

Menggunakan Library

Melakukan AJAX Request dengan Library



```
$.ajax({
  method: "POST",
  url: "some.php",
  data: { name: "John", location: "Boston" }
}).done(function( msg ) {
  alert( "Data Saved: " + msg );
});
```

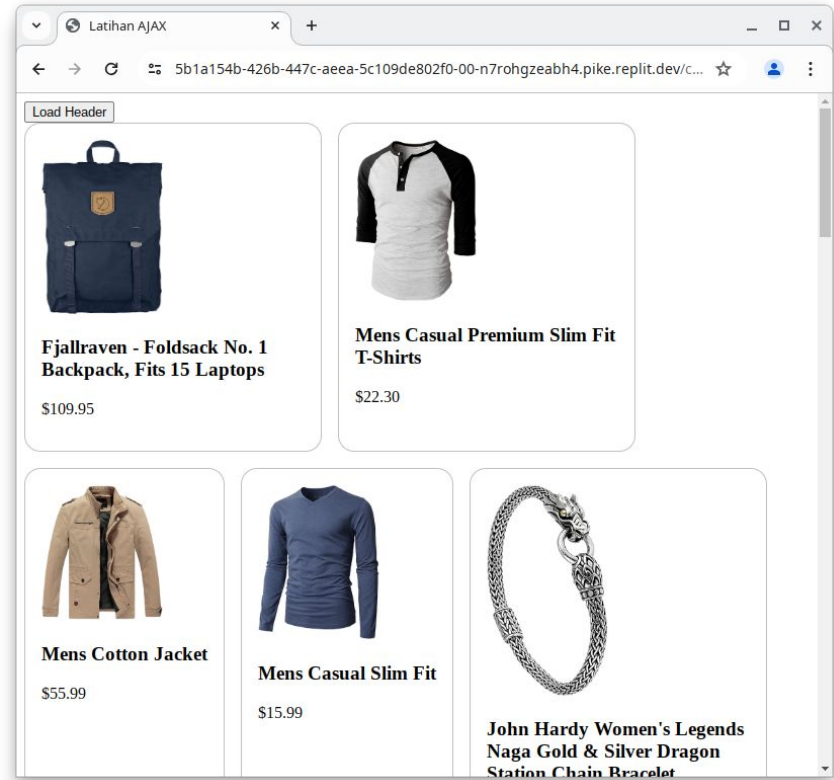


```
axios.get('/user?ID=12345')
  .then(function (response) {
    // handle success
    console.log(response);
  })
  .catch(function (error) {
    // handle error
    console.log(error);
  })
  .finally(function () {
    // always executed
  });
```

Latihan 

Latihan: AJAX dengan XMLHttpRequest

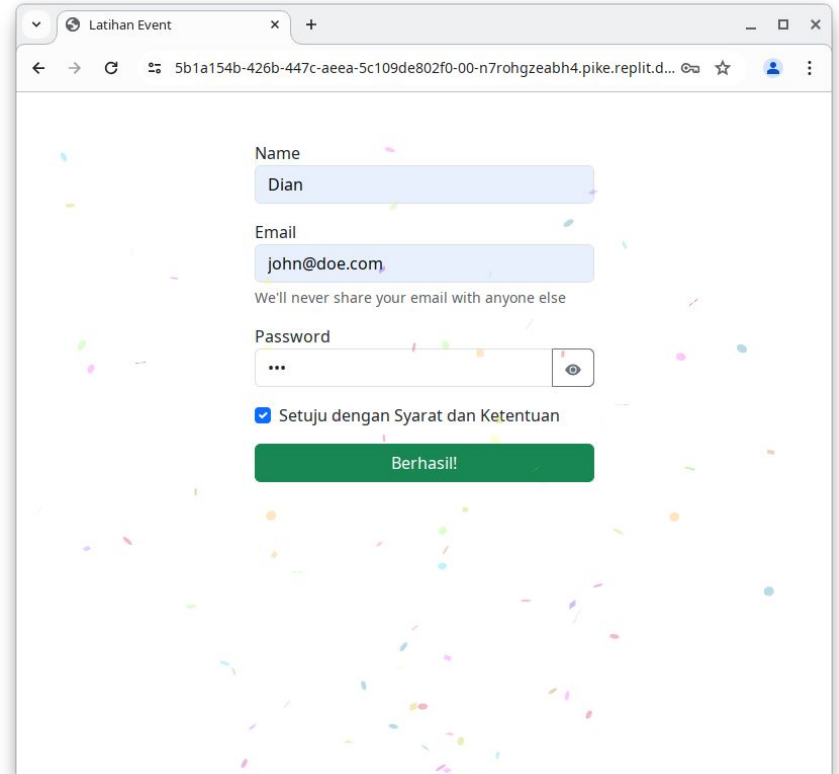
- Ambil data produk dari <https://fakestoreapi.com/products> lalu tampilkan ke HTML
- Load header dengan AJAX saat tombol diklik



Latihan: AJAX dengan fetch()

- Ubah form register dari pertemuan 16 (latihan-event.html)
- Tambahkan AJAX Request dengan fetch() untuk mengirim data ke server:

<https://codesandbox.io/p/sandbox/coding-subuh-ajax-server-33zfhf>



The screenshot shows a web browser window with the title 'Latihan Event'. The address bar displays a URL from a Replit sandbox. The form contains the following elements:

- Name:** A text input field containing the name 'Dian'.
- Email:** A text input field containing the email 'john@doe.com'. Below the field is the text 'We'll never share your email with anyone else'.
- Password:** A password input field with a masked view icon (an eye with a slash) on the right.
- Terms:** A checkbox labeled 'Setuju dengan Syarat dan Ketentuan' which is currently checked.
- Submit:** A green button labeled 'Berhasil!'.

The background of the form is decorated with colorful confetti.

Latihan: AJAX Upload

- Buat form upload file dan kirim filenya dengan AJAX ke server:
<https://codesandbox.io/p/sandbox/coding-subuh-ajax-server-33zfhf>

Upload File dengan AJAX

Choose File

Shantell_Sans_1.008.zip

Upload

 100%



Selamat

Kita sudah selesai hari ini..

Resources & Referensi

- [XMLHttpRequest: responseType property - Web APIs | MDN](#)
- [Using the Fetch API](#)
- <https://www.petanikode.com/javascript-ajax/>
- [Belajar Nodejs #12: Menggunakan Database SQLite pada Nodejs](#)