

---

# **Enterprise Front End Developer**

Materi Pembelajaran

Dr. Bambang Purnomosidi D. P.



# Daftar Isi

<b>Tentang Buku Ini</b>	<b>1</b>
<b>Minggu 01</b>	<b>2</b>
Hari 1: Ekosistem dan Pemrograman Dasar JavaScript . . . . .	2
Tujuan . . . . .	2
Pembahasan . . . . .	2
Pembelajaran . . . . .	3
Hari 2: Konstruksi Dasar JavaScript . . . . .	4
Tujuan . . . . .	4
Pembahasan . . . . .	4
Pembelajaran . . . . .	4
Hari 3: Function / Fungsi di JavaScript . . . . .	5
Tujuan . . . . .	5
Pembahasan . . . . .	5
Pembelajaran . . . . .	5
Hari 4: String dan Regular Expression . . . . .	6
Tujuan . . . . .	6
Pembahasan . . . . .	6
Pembelajaran . . . . .	6
Hari 5: Struktur Data di JavaScript . . . . .	7
Tujuan . . . . .	7
Pembahasan . . . . .	7
Pembelajaran . . . . .	7
<b>Minggu 02</b>	<b>9</b>
Hari 1: Paradigma Pemrograman di JavaScript: OOP (1)* . . . . .	9
Tujuan . . . . .	9
Pembahasan . . . . .	9
Pembelajaran . . . . .	9
Hari 2: Paradigma Pemrograman di JavaScript: OOP (2) . . . . .	10
Tujuan . . . . .	10

Pembahasan . . . . .	10
Pembelajaran . . . . .	10
Hari 3: Paradigma Pemrograman di JavaScript: Functional Programming . . . . .	11
Tujuan . . . . .	11
Pembahasan . . . . .	11
Pembelajaran . . . . .	12
Hari 4: Asynchronous Programming di JavaScript . . . . .	12
Tujuan . . . . .	12
Pembahasan . . . . .	13
Pembelajaran . . . . .	13
Hari 5: JavaScript Modules . . . . .	14
Tujuan . . . . .	14
Pembahasan . . . . .	14
Pembelajaran . . . . .	14
<b>Minggu 03</b>	<b>16</b>
Hari 1: <i>Error Handling</i> . . . . .	16
Tujuan . . . . .	16
Pembahasan . . . . .	16
Pembelajaran . . . . .	16
Hari 2: Web API . . . . .	17
Tujuan . . . . .	17
Pembahasan . . . . .	17
Pembelajaran . . . . .	17
Hari 3: Manipulasi DOM . . . . .	18
Tujuan . . . . .	18
Pembahasan . . . . .	18
Pembelajaran . . . . .	19
Hari 4: DOM Lanjut . . . . .	19
Tujuan . . . . .	19
Pembahasan . . . . .	19
Pembelajaran . . . . .	20
Hari 5: Pengenalan Vue.js . . . . .	20
Tujuan . . . . .	20
Pembahasan . . . . .	20
Pembelajaran . . . . .	20

<b>Minggu 04</b>	<b>22</b>
Hari 1: Lebih Lanjut Dengan Vue.js (1)	22
Tujuan	22
Pembahasan	22
Pembelajaran	22
Hari 2: Lebih Lanjut Dengan Vue.js (2)	23
Tujuan	23
Pembahasan	23
Hari 3: Just Enough React	24
Tujuan	24
Pembahasan	24
Hari 4: single-spa untuk Micro Frontends	24
Tujuan	24
Pembahasan	24
Pembelajaran	25
Hari 5: JavaScript Front End Tooling	25

# Tentang Buku Ini



## Versi

1 1.0.0-RC--17-Jul-2019--05:10:12

Buku ini merupakan buku pegangan program “**Enterprise Front End Developer**” untuk para mentor, siswa, serta pihak-pihak lain yang berkepentingan dengan proses pendidikan di Praxis Academy. Dengan panduan buku ini, diharapkan proses belajar menjadi lebih efektif dan efisien serta mempunyai pedoman yang jelas. Lisensi buku ini adalah Creative Commons Attributiona ShareAlike 4.0 International License - CC-BY-SA 4.0. Secara umum, penggunaan lisensi ini mempunyai implikasi bahwa pengguna materi:

1. Harus memberikan atribusi ke penulis dan sponsor untuk penulisan materi ini (Praxis Academy).
2. Boleh menggunakan produk yang ada disini untuk keperluan apapun jika point 1 di atas terpenuhi.
3. Boleh membuat produk derivatif dari produk yang ada disini sepanjang perubahan-perubahan yang dilakukan diberitahukan ke kami dan di-share dengan menggunakan lisensi yang sama.

Untuk penggunaan selain ketentuan tersebut, silahkan menghubungi:

1 Praxis Academy  
2 Jl.Garuda no 67, Manukan  
3 Condong Catur  
4 Sleman  
5 Yogyakarta 55283  
6 Indonesia  
7 Email: [hello@praxisacademy.id](mailto:hello@praxisacademy.id)  
8 Web: <https://praxisacademy.id>

# Minggu 01

## Hari 1: Ekosistem dan Pemrograman Dasar JavaScript

### Tujuan

1. Siswa bisa memahami keterkaitan antara bahasa pemrograman dengan *compiler/interpreter*.
2. Siswa memahami komponen dari peranti pengembangan (*development tools*) dan bisa mencari komponen-komponen untuk suatu bahasa pemrograman tertentu.
3. Siswa memahami keterkaitan antara ECMAScript, JavaScript, dan Node.js,
4. Siswa memahami keterkaitan antara Browser, HTML, CSS, JavaScript.
5. Siswa mampu menginstall Visual Studio Code serta plugin untuk peranti pengembangan JavaScript. Siswa juga dibebaskan menggunakan editor teks maupun IDE lainnya.
6. Siswa mampu menginstall Node.js
7. Siswa memahami, mampu membuat, serta menjalankan *source code* JavaScript, baik untuk sisi client (menggunakan *browser*) maupun di luar *browser* (Node.js).
8. Siswa memahami dan mampu menggunakan [npm](#) untuk mengelola paket pustaka Node.js
9. Siswa memahami struktur dasar *source code* dalam bahasa pemrograman JavaScript baik untuk sisi *client (browser)* maupun di luar *browser*.
10. Siswa memahami dan mampu menggunakan konstruksi dasar JavaScript untuk mengimplementasikan algoritma sederhana.

### Pembahasan

1. Development tools dan komponen-komponennya.
2. Ekosistem JavaScript: ES, JS, Node.js, Browser, HTML, CSS.
3. Dasar-dasar JavaScript:
  - JavaScript untuk browser dan di luar browser
  - Tools dan utilities
  - Konstruksi dasar bahasa pemrograman JavaScript: sintaksis dasar, komentar, variabel, tipe data, ekspresi, operator, pengenalan obyek, literal.

## Pembelajaran

### 1 Materi dan Penjelasan

1. Keterkaitan antara bahasa pemrograman, compiler, dan interpreter.
2. How Can We Define a Programming Language menjelaskan keterkaitan antara spesifikasi bahasa pemrograman dengan *reference implementation* dalam bentuk *compiler* maupun *interpreter*. Halaman Wikipedia untuk spesifikasi bahasa pemrograman menjelaskan tentang gambaran umum spesifikasi berbagai bahasa pemrograman.
3. Komponen dari peranti pengembangan (*development tools*).
4. Spesifikasi ECMAScript / JavaScript - lihat di ECMAScript. Lihat juga penjelasan di Mozilla Developer Network tentang ECMAScript dan JavaScript
5. Keterkaitan antara HTML, CSS, dan JavaScript dijelaskan pada blog ini dan jawaban di Stack-Overflow ini. Tambahan untuk pengetahuan: How Browsers Work: Behind the scenes of modern web browsers.
6. Instalasi Visual Studio Code serta plugin untuk peranti pengembangan JavaScript.
7. Ekosistem dari JavaScript / Node.js dan awesome JavaScript memberikan gambaran tentang ekosistem JavaScript.
8. JavaScript di luar browser: Wikipedia dan URL Node.js.
9. Instalasi Node.js
10. Introduction to [npm](#) and basic [npm](#) commands menjelaskan tentang dasar-dasar penggunaan npm untuk mengelola paket.
11. JavaScript basics memberikan gambaran singkat tentang konstruksi dasar dari JavaScript untuk sisi front end.
12. Eloquent JavaScript - Introduction, bab 1 (Values, Types, and Operators), dan bab 2 (Program Structure) memberikan penjelasan yang memadai tentang konstruksi dasar dari JavaScript.
13. Materi di Mozilla Developer Network sebagai referensi.

### 1 Latihan

1. Install Node.js.
2. Install Visual Studio Code dan konfigurasikan untuk JavaScript.
3. Kerjakan no 11 dan 12.

### 1 Kasus

Kerjakan [Exercises](#) pada buku Eloquent JavaScript - bab 2 (Program Structures).

## Hari 2: Konstruksi Dasar JavaScript

### Tujuan

1. Siswa memahami dan mampu menggunakan berbagai statement untuk:
  - pengendali alur
  - iterasi dan looping

### Pembahasan

1. Pengendali alur:
  - if .. else
  - switch
  - try ... catch ... throw
2. Iterasi dan *looping*
  - for
  - while
  - do ... while
  - break/continue
  - for ... in
  - for ... of

### Pembelajaran

#### 1 Materi dan Penjelasan

1. Materi di MDN tentang pengendali alur.
2. Materi di MDN tentang iterasi dan *looping*.

#### 1 Latihan

Kerjakan contoh-contoh yang dibahas pada materi dan penjelasan di atas.

#### 1 Kasus

Kerjakan kasus yang ada di w3resource.com.



## Hari 3: Function / Fungsi di JavaScript

### Tujuan

1. Siswa cara mendeklarasikan *function* di JS serta cara menggunakannya
2. Siswa memahami dan mampu menggunakan berbagai jenis *function* yang ada di JS.

### Pembahasan

1. Pengertian fungsi
2. Definisi dan pemanggilan fungsi
3. Ruang lingkup variabel
4. *Nested functions* dan *closure*
5. Fungsi *generator*
6. Fungsi *arrow*
7. Fungsi *hoisted*

### Pembelajaran

#### 1 Materi dan Penjelasan

1. Materi sangat mendasar tentang fungsi.
2. Materi tentang Functions di Eloquent JavaScript serta materi tentang *function* di MDN memberikan gambaran yang memadai tentang *function* di JS.
3. Baca juga tentang variable hoisting serta *function hoisting*.
4. Self executing function.
5. Nested functions dan closure di javascript.info.
6. Generator function).
7. Pemahaman arrow function lebih dalam akan bisa diperoleh saat sudah mempelajari paradigma pemrograman dan struktur data, tetapi untuk saat ini silahkan dipelajari artikel pengantar ini terlebih dahulu dan artikel ini.
8. Artikel tentang function hoisting.

#### 1 Latihan

Kerjakan contoh-contoh yang ada pada materi dan penjelasan di atas, minimal satu kode sumber untuk satu point materi dan penjelasan di atas dan harus yang menunjukkan pembahasan utama.

Contoh, saat membahas tentang closure di materi nomor 5, kerjakan minimal satu contoh yang menampilkan penuh tentang closure.

#### 1 Kasus

Kerjakan kasus yang ada pada latihan ini minimal 2. Lengkapi sampai ke file HTML-nya sehingga untuk menjalankan cukup memanggil file HTML dari browser melalui protokol `file:///lokasi/file/html`

## Hari 4: String dan Regular Expression

### Tujuan

1. Siswa memahami dan mampu memanipulasi string menggunakan JavaScript
2. Siswa memahami pengertian *regular expression* serta mampu menggunakannya pada kode sumber yang dibuat untuk mengenali pola dan melakukan sesuatu terkait pola yang ditemukan tersebut.

### Pembahasan

1. Obyek String di JavaScript
2. Gambaran umum RegExp
3. Pola dan RegExp
4. Karakter-karakter RegExp dan penggunaannya

### Pembelajaran

#### 1 Materi dan Penjelasan

1. Dokumentasi di MDN tentang pengolahan teks / string.
2. Dokumentasi di MDN tentang RegExp.

#### 1 Latihan

Kerjakan contoh-contoh yang ada pada materi dan penjelasan

#### 1 Kasus

1. Buat input HTML yang meminta input berupa teks, setelah itu deteksi menggunakan *regular expression* apakah teks tersebut adalah plat nomor kendaraan di Indonesia atau bukan. Gunakan alert untuk menampilkan hasilnya.
2. Buat input HTML yang meminta input berupa alamat email, setelah itu deteksi menggunakan *regular expression* apakah teks tersebut adalah email atau bukan. Jika, email, tampilkan pada alert:

Halo, username anda adalah *username*, domain anda adalah *domain.tld*.

Contoh, jika e-mail adalah namaku@praxisacademy.id, maka alert yang akan muncul adalah:

```
1 Halo, username anda adalah namaku, domain anda adalah praxisacademy.id
```

## Hari 5: Struktur Data di JavaScript

### Tujuan

1. Siswa memahami dan mampu menggunakan konstruksi Object di JS.
2. Siswa memahami dan mampu menggunakan Array di JS.
3. Siswa memahami dan mampu menggunakan Iterables di JS.
4. Siswa memahami dan mampu menggunakan Map dan Set di JS.
5. Siswa memahami dan mampu menggunakan WeakMap dan WeakSet.
6. Siswa memahami dan mampu menggunakan data Date dan Time.

### Pembahasan

1. Obyek
2. Array
3. Iterables
4. Map dan Set
5. WeakMap dan WeakSet
6. Date dan Time

### Pembelajaran

```
1 Materi dan Penjelasan
```

1. Objects and Arrays, lihat juga Object.keys, values, entries.

2. Iterables
3. Map dan Set
4. WeakMap dan WeakSet
5. Date dan Time.

#### 1 Latihan

Selesaikan semua *tasks* dari **javascript.info** untuk masing-masing pembahasan (no 1 - 5) di atas.

#### 1 Kasus

1. Buat program untuk menjumlahkan 2 buah matriks
2. Buat program kamus Inggris-Indonesia menggunakan Map. Fasilitas program tersebut adalah:
  - mencari kata dan menampilkan artinya
  - menambahkan kata dan artinya
  - menghapus kata serta artinya.

# Minggu 02

## Hari 1: Paradigma Pemrograman di JavaScript: OOP (1)\*

### Tujuan

1. Siswa mengetahui berbagai paradigma pemrograman yang didukung oleh JavaScript.
2. Siswa memahami dan mampu menggunakan fitur OOP menggunakan *prototype-based*

### Pembahasan

1. Paradigma pemrograman
2. Memahami OOP
3. OOP di JavaScript: Prototype-based
4. Prototype-based OOP di JavaScript

### Pembelajaran

#### 1 Materi dan Penjelasan

1. Halaman Wikipedia untuk paradigma pemrograman harus dipahami supaya mengetahui berbagai paradigma pemrograman yang ada sampai saat ini serta berbagai fitur utamanya.
2. Halaman Wikipedia untuk OOP merupakan ringkasan tentang OOP, harus dipahami sebagai gambaran umum sebelum masuk ke *prototype-based* dan *class-based* OOP.
3. Halaman Wikipedia untuk prototpe-based OOP harus dipahami sebelum masuk ke teknis penggunaan *prototype* di JavaScript.
4. Object.prototype di MDN
5. Beberapa artikel:
  - Prototype-based Inheritance and Prototype chain in JavaScript (ES5) menjelaskan tentang implementasi *prototype-based* OOP di JavaScript.
  - Protoype in JavaScript.

- Introduction.
- JavaScript Prototype Object.
- Master JavaScript Prototypes and Inheritance.
- Artikel di DigitalOcean.
- Prototype by examples.
- Prototypal Inheritance.
- Prototype in JS.

#### 1 Latihan

1. Kerjakan contoh-contoh seperti yang dijelaskan pada point materi dan penjelasan nomor 4 di atas, minimal dari 2 artikel.
2. Kerjakan latihan dari buku ini.

#### 1 Kasus

Kerjakan kasus di Hackerrank.

## Hari 2: Paradigma Pemrograman di JavaScript: OOP (2)

### Tujuan

1. Siswa memahami dan mampu menggunakan class serta obyek di aplikasi yang dibuat.
2. Siswa memahami dan mampu mendefinisikan class
3. Siswa memahami dan mampu membuat konstruktor
4. Siswa memahami dan mampu membuat properties dan methods pada class.
5. Siswa memahami dan mampu meng-*inherit* suatu class untuk membuat class baru.

### Pembahasan

1. Class-based OOP di JavaScript (ES6)
2. Definisi class
3. Konstruktor
4. Properties dan methods
5. Inheritance

### Pembelajaran

### 1 Materi dan Penjelasan

1. Materi tentang *class* di MDN.
2. Beberapa artikel:
  - JavaScript Classes - Under the Hood.
  - Materi class di javascript.info.
  - [Understanding Classes in JavaScript]((https://www.digitalocean.com/community/tutorials/understanding-classes-in-javascript)).
  - Contoh-contoh.
  - Classes in JavaScript.

### 1 Latihan

Kerjakan beberapa contoh yang diberikan di berbagai artikel di materi dan penjelasan nomor 2, minimal buatlah 1 program pada tiap-tiap artikel.

### 1 Kasus

Kerjakan salah satu kasus OOP di GitHub ini.

## Hari 3: Paradigma Pemrograman di JavaScript: Functional Programming

### Tujuan

1. Siswa memahami paradigma *functional programming*.
2. Siswa memahami dan mampu menggunakan fitur FP di JavaScript:
  - *Side effects*
  - *immutability*
  - HoF (\*Higher-order Function)
  - Rekursi
  - Currying

### Pembahasan

1. Gambaran umum *functional programming*
2. *Side effects*

3. *Immutability*
4. *Higher-order functions*
5. Rekursi dan *functional programming*
6. *Currying*

## Pembelajaran

### 1 Materi dan Penjelasan

1. Artikel Wikipedia tentang Functional Programming memberikan ringkasan tentang *functional programming*. Pelajari dan pahami untuk mendapatkan gambaran umum dari FP.
2. Beberapa artikel:
  - Artikel di CodeBurst.io memberikan gambaran umum tentang FP di JavaScript.
  - Pelajari juga artikel dari freecodecamp.org.
  - Master the JavaScript Interview: What is Functional Programming?
  - Awesome FP - JS.
  - An Introduction to FP with JS.
  - Currying in JS.

### 1 Latihan

1. Kerjakan contoh-contoh seperti yang dijelaskan pada point materi dan penjelasan nomor 2 di atas.

### 1 Kasus

Artikel pada materi dan penjelasan nomor 2 (CodeBurst.io) menyebutkan ada 3 pustaka untuk *functional programming* di JavaScript: Ramda, Underscore, dan lodash. Cari kesamaan antara 3 pustaka tersebut sebanyak 3 hal saja, buat contoh implementasi menggunakan masing-masing pustaka tersebut.

## Hari 4: Asynchronous Programming di JavaScript

### Tujuan

1. Siswa memahami pengertian *asynchronous programming*.
2. Siswa memahami berbagai teknik pemrograman di JavaScript untuk *asynchronous programming*.



3. Siswa memahami dan mampu menggunakan *callback* untuk *asynchronous programming*.
4. Siswa memahami dan mampu menggunakan *promise* untuk *asynchronous programming*.
5. Siswa memahami dan mampu menggunakan *async/await* untuk *asynchronous programming*.

## Pembahasan

1. Pengertian *asynchronous programming* / non-blocking I/O
2. Berbagai teknik pemrograman di JavaScript untuk *asynchronous programming*
3. *Callback* di JavaScript
4. *Promise* di JavaScript
5. *async/await* di JavaScript

## Pembelajaran

### 1 Materi dan Penjelasan

Referensi: libuv - pustaka yang digunakan oleh node.js untuk implementasi non-blocking I/O.

1. Artikel di Wikipedia tentang Asynchronous I/O memberikan ringkasan komprehensif tentang pengertian *asynchronous I/O*. Baca dan pahami juga penjelasan di StackOverflow. Artikel byte archer ini juga memberikan gambaran yang cukup umum tentang *blocking (synchronous)* dan *non-blocking (asynchronous) I/O*. Artikel lain.
2. Artikel node.dev tentang event-loop memberikan penjelasan tentang *event-loop* di Node.js yang menjadi latar belakang munculnya *asynchronous I/O*.
3. Non-blocking I/O model di node.js menjelaskan tentang mekanisme *non-blocking I/O* di node.js.
4. Perbedaan blocking dan non-blocking di nodejs.org menjelaskan secara lebih spesifik perbedaan *blocking* dan *non-blocking I/O di node.js*
5. Gambaran umum implementasi non-blocking I/O di JavaScript.
6. Callback pattern di node.js. Lebih rinci ada di Event Loop dan Callback.
7. Tutorialspoint - callback.
8. Spesifikasi Promises/A+, hanya jika berminat ke spesifikasi dan ingin mengimplementasikan. Biasanya dipelajari oleh vendor / implementor bahasa pemrograman.
9. Promise - MDN memberikan latar belakang serta contoh-contoh terkait Promise. Using Promise memberikan petunjuk penggunaan *promise*.
10. Promises dan *async/await* di javascript.info serta *async/await* di javascript.info.
11. Generators function menjelaskan tentang generators. Teknik pemrograman lebih lanjut biasanya mengakomodasi *async* di generators.

## 1 Latihan

Kerjakan latihan-latihan yang ada pada materi dan penjelasan di atas, setidaknya ada latihan untuk `callback`, `promises`, `async/await`.

## 1 Kasus

Dengan menggunakan `node.js`, buatlah program untuk mengakses / mengambil konten pada suatu *endpoint* / URL di Web menggunakan *non-blocking I/O* di JavaScript. Saran: gunakan artikel ini untuk mengetahui berbagai pustaka *http client* yang bisa digunakan.

# Hari 5:JavaScript Modules

## Tujuan

1. Siswa memahami pengertian module di JavaScript dan bagaimana module itu akan digunakan.
2. Siswa memahami cara JavaScript mengimplementasikan modules
3. Siswa mampu menggunakan module yang sudah ada
4. Siswa mampu membuat module

## Pembahasan

1. Pengertian modules
2. Gambaran umum implementasi *modules* di JavaScript
3. Membuat dan menggunakan modules

## Pembelajaran

## 1 Materi dan Penjelasan

1. Artikel JavaScript modules di MDN. Sampai saat ini ada beberapa format modules.
2. Exploring JavaScript.
3. Membuat modules.
4. JavaScript modules.
5. Understanding ES6 modules.
6. JavaScript modules.

## 7. Modules - javascript.info.

### 1 Latihan

Kerjakan contoh-contoh yang ada pada materi dan penjelasan. Silahkan memilih materi minimal dari 4 artikel materi dan penjelasan.

### 1 Kasus

Buatlah module menggunakan class. Buat juga contoh pemakaiannya.

# Minggu 03

## Hari 1: *Error Handling*

### Tujuan

1. Siswa memahami berbagai kategori error
2. Siswa mengetahui secara umum, teknik penanganan error pada JavaScript
3. Siswa memahami dan bisa menggunakan *try...catch...finally* di JavaScript untuk mengelola error.
4. Siswa memahami dan bisa menggunakan *Throw* di JavaScript untuk mengelola error.
5. Siswa memahami dan bisa menggunakan *onerror* di JavaScript untuk mengelola error.
6. Siswa memahami dan mampu membuat sendiri *custom* error dengan cara meng-*extend* Error.

### Pembahasan

1. Kategori Error
2. Gambaran umum teknik penanganan error dan exception di JavaScript
3. Berbagai konstruksi bahasa pemrograman di JavaScript untuk menangani error:
  - *try .. catch .. finally*
  - *Throw*
  - *onerror*
4. *Custom error*.

### Pembelajaran

#### 1 Materi dan Penjelasan

1. Materi di [scotch.io](https://scotch.io) ini menjelaskan secara umum tentang error dan penanganannya di JavaScript.

2. Materi di buku Eloquent JavaScript menjelaskan secara lebih gamblang tentang penanganan error dan praktik-praktik yang baik.
3. Materi di JavaScript.info menjelaskan secara detail penggunaan *try ... catch ... finally*.
4. Error bisa dikustomisasi. Materi di JavaScript.info ini menjelaskan tentang hal tersebut. Lihat juga referensi obyek Error di MDN.
5. Materi tentang control flow and error handling di MDN menjelaskan tentang penanganan error lebih lanjut terkait *control flow*.
6. Materi di sitepoint.com.

#### 1 Latihan

1. Kerjakan latihan di JavaScript.info untuk point 3 dan 4 di atas. Jelaskan di file README.md

#### 1 Kasus

1. Kerjakan latihan tentang error dan bugs di Eloquent JavaScript (point 2) di atas.

## Hari 2: Web API

### Tujuan

1. Siswa memahami pengertian Web API dan kaitannya dengan *front end development*
2. Siswa memahami berbagai Web API yang tersedia di sisi klien
3. Siswa mampu mencari informasi serta dokumentasi tentang Web API dan mampu menggunakannya dalam *front end development*.

### Pembahasan

1. Gambaran umum dan pengenalan (*client-side*) Web API
2. Kategori Web API
3. Berbagai Web API yang tersedia
4. Penggunaan berbagai Web API

### Pembelajaran

#### 1 Materi dan Penjelasan

Istilah Web API sebenarnya merupakan istilah yang generik dan bisa digunakan pada berbagai konteks (misalnya bisa mengacu pada *web services* yang disediakan oleh server), tetapi dalam pembahasan ini Web API mengacu pada API yang disediakan di platform browser dan pada dasarnya digunakan oleh JavaScript (meski mungkin juga di HTML - CSS).

1. Materi Client-side Web APIs - pengenalan di MDN memberikan ringkasan secara umum tentang *client-side Web API*.
2. Introduction to Web APIs merupakan materi lanjutan untuk membahas lebih lanjut tentang Web API.
3. Artikel tentang keterkaitan browser dengan Web API.
4. Kategori dan daftar Web API di browser.
5. Contoh pembahasan: Fetch API. Konsep dan penggunaan.

#### 1 Latihan

Kerjakan contoh-contoh di Fetch API pada materi dan penjelasan nomor 5.

#### 1 Kasus

Berkreasilah menggunakan salah satu dari Web API yang ada pada daftar Web APIs.

## Hari 3: Manipulasi DOM

### Tujuan

1. Siswa memahami pengertian DOM
2. Siswa memahami keterkaitan DOM dengan JavaScript
3. Siswa bisa mengakses DOM menggunakan JavaScript
4. Siswa memahami dan bisa menggunakan tipe data fundamental dalam DOM
5. Siswa memahami dan bisa menggunakan dasar-dasar DOM *interfaces - objects*.

### Pembahasan

1. Pengertian dan gambaran umum DOM
2. DOM dan JavaScript
3. Mengakses DOM menggunakan JavaScript
4. Tipe data fundamental dalam DOM
5. *Interfaces - objects*

## Pembelajaran

### 1 Materi dan Penjelasan

1. Materi di Eloquent JavaScript tentang JavaScript dan browser harus dipahami terlebih dahulu sebelum masuk ke DOM.
2. Introduction to the DOM dari MDN merupakan ringkasan dasar tentang DOM. Lihat juga materi Manipulating documents di MDN.
3. Materi di Eloquent JavaScript menjelaskan lebih detail tentang pengenalan DOM dan JavaScript.

### 1 Latihan

Kerjakan latihan tentang DOM di buku Eloquent JavaScript di point 3.

### 1 Kasus

Lihat kasus di akhir materi *Manipulating documents* pada point 6 di atas. Kerjakan kasus tersebut. Gunakan Bulma untuk framework CSS tampilan tersebut.

## Hari 4: DOM Lanjut

### Tujuan

1. Siswa memahami dan mampu memanfaatkan node
2. Siswa memahami dan mampu memanfaatkan CSS
3. Siswa memahami dan mampu memanfaatkan event
4. Siswa memahami dan bisa memanipulasi komponent-komponen DOM (node, CSS, event) menggunakan JavaScript.

### Pembahasan

1. Node
2. CSS
3. Event DOM
4. Mengakses DOM menggunakan JavaScript

## Pembelajaran

### 1 Materi dan Penjelasan

0. Buku DOM Enlightenment memberikan referensi yang lengkap tentang DOM. Silahkan melihat ke buku ini untuk referensi berbagai hal terkait DOM.
1. DOM Node di MDN menjelaskan secara rinci tentang DOM node.
2. Secara umum, CSSOM dijelaskan di CSSOM Introduction. Setelah itu pelajari Referensi CSSOM. CSSOM menjelaskan CSS Object Model dan kaitannya dengan manipulasi *style* di CSS.
3. Materi Event di MDN menjelaskan pengertian Event serta referensi berbagai event yang tersedia.
4. Pelajari cara membuat dan men-trigger event.
5. Obyek Event.

### 1 Latihan

Kerjakan contoh-contoh di 15 JavaScript Methods For DOM Manipulation for Web Developers. Jelaskan masing-masing contoh tersebut di README.md

### 1 Kasus

Kerjakan DOM manipulation exercises

## Hari 5: Pengenalan Vue.js

### Tujuan

Siswa memahami dasar-dasar penggunaan Vue.js: 1. Instalasi 2. Aktivasi di browser 3. Gambaran umum penggunaan Vue.js

### Pembahasan

1. Dasar-dasar Vue.js
2. Instalasi Vue.js
3. Pengenalan Vue.js
4. Contoh program menggunakan Vue.js

## Pembelajaran



## 1 Materi dan Penjelasan

1. Dsaar-dasar Vue.js
2. Pelajari Vue.js Guide pada bagian Installation dan Introduction.
3. Lihat berbagai contoh penggunaan Vue.js.

## 1 Latihan

Kerjakan berbagai contoh di point 7 (Vue.js). Perhatikan dan pelajari *source code* contoh-contoh tersebut.

## 1 Kasus

Pilih 3 contoh di point 3 (materi dan penjelasan Vue.js), buat *source code*-nya dengan menggunakan template Bulma. Jelaskan 3 contoh tersebut dalam README.md.

# Minggu 04

## Hari 1: Lebih Lanjut Dengan Vue.js (1)

### Tujuan

1. Siswa memahami dan mampu menggunakan berbagai komponen Vue.js
2. Siswa memahami dan mampu menggunakan *transition wrapper component*.
3. Siswa memahami dan mampu menggunakan fitur animasi pada Vue.js

### Pembahasan

1. Komponen
2. Transisi
3. Animasi

### Pembelajaran

#### 1 Materi dan Penjelasan

1. Pelajari Vue.js Guide untuk materi-materi di atas

#### 1 Latihan

Kerjakan latihan-latihan yang ada pada Vue.js Guide - selain Installation dan Introduction, utamakan untuk latihan-latihan yang terkait dengan komponen, *transition wrapper component*, dan animasi.

#### 1 Kasus

Buat komponen *Tree View* yang berisi berbagai repo dari username di GitHub. Saat memilih (klik) pada suatu repo, ditampilkan README.dari repo tersebut.

## Hari 2: Lebih Lanjut Dengan Vue.js (2)

### Tujuan

1. Siswa memahami dan mampu menggunakan fitur mixins untuk *reusable components* dari Vue.js
2. Siswa memahami dan mampu menggunakan *custom directives* dari Vue.js
3. Siswa memahami dan mampu menggunakan *render* dan JSX dari Vue.js
4. Siswa memahami dan mampu menggunakan *plugins* serta *filters* dari Vue.js
5. Siswa memahami dan mampu menggunakan berbagai *tooling* di Vue.js
6. Siswa memahami dan mampu melakukan *scaling up* saat menggunakan Vue.js untuk pengembangan aplikasi
7. Siswa memahami ekosistem dari Vue.js dan bisa memanfaatkan ekosistem tersebut untuk menyelesaikan pekerjaan *front end development*.

### Pembahasan

1. Reusability and Compositions
2. Tooling
3. Scaling Up
4. Ekosistem Vue.js

#### 1 Materi dan Penjelasan

1. Pelajari Vue.js Guide untuk materi-materi di atas
2. Pelajari materi yang ada pada menu *Ecosystem* di Vue.js, khususnya *Tooling* dan *Core libraries*.

#### 1 Latihan

Kerjakan latihan-latihan yang ada pada Vue.js Guide - selain Installation dan Introduction, utamakan untuk latihan-latihan yang terkait dengan materi-materi pada pembahasan di atas.

#### 1 Kasus

Pelajari contoh Vue.js untuk *pagination*. Ubah menjadi daftar *commit* dari suatu repo di GitHub.

## Hari 3: Just Enough React

### Tujuan

1. Siswa memahami dasar-dasar pemrograman UI di Web menggunakan React.

### Pembahasan

1. Setup untuk React
2. Dasar-dasar React: *components, props, state*
3. Teknik pembuatan aplikasi di React

#### 1 Materi dan Penjelasan

Pelajari React Tutorial dengan baik.

#### 1 Latihan

Kerjakan React Tutorial

#### 1 Kasus

Kerjakan minimal 2 dari *ideas for improvements* di bagian akhir React Tutorial.

## Hari 4: single-spa untuk Micro Frontends

### Tujuan

1. Siswa memahami pengertian Micro Frontends dan mampu merancang solusi pengembangan aplikasi front end menggunakan arsitektur Micro Frontends.
2. Siswa memahami dan mampu menggunakan *single-spa* untuk membangun aplikasi berbasis arsitektur Micro Frontends.
3. Siswa memahami cara mengintegrasikan berbagai *framework* JavaScript di sisi front end untuk aplikasi.

### Pembahasan

1. Arsitektur Micro Frontends
2. Memahami single-spa

### 3. Implementasi Arsitektur Micro Frontends Menggunakan React, Vue.js, dan single-spa

## Pembelajaran

### 1 Materi dan Penjelasan

1. Web resmi arsitektur Micro Frontends.
2. ThoughtWorks Technology Radar sudah menetapkan Micro Frontends dalam kondisi *Adopt*.
3. single-spa adalah framework yang digunakan untuk mengimplementasikan Micro Frontends. Artikel langkah-perlangkah menggunakan single-spa.
4. Contoh Micro Frontends menggunakan single-spa menampilkan berbagai contoh implementasi single-spa untuk Micro Frontends.
5. Artikel tentang Micro Frontends menggunakan single-spa.
6. Contoh menggunakan React, Vue, dan single-spa

### 1 Latihan

1. Kerjakan point nomor 6 di atas.
2. Perhatikan dan pelajari contoh-contoh di point nomor 4 di atas. Kerjakan semaksimal mungkin yang bisa anda kerjakan untuk contoh-contoh tersebut.

### 1 Kasus

Meneruskan hari pertama di minggu ini, setelah selesai membuat Tree View, tambahkan satu routing yang akan mengakses data pribadi public user di GitHub (nama, username, dan seterusnya). Tampilkan dengan menggunakan React. Buat menjadi 2 routing:

1. /user => menampilkan informasi publik dari user (menggunakan React)
2. /repos => menampilkan seperti kasus sebelumnya.

Gunakan single-spa untuk keperluan tersebut.

## Hari 5: JavaScript Front End Tooling

Di akhir pelatihan, sebagai *front end developer*, siswa harus memahami bahwa JavaScript serta front end development mempunyai ekosistem yang luar biasa banyak dan berkembang dengan cepat. Kemauan untuk mengikuti serta kemampuan beradaptasi diperlukan bagi siswa. Untuk mengikuti perkembangan, silahkan mengikuti dari [frontendtools.com](https://frontendtools.com). Silahkan melihat berbagai *tools* yang

belum pernah dipelajari atau digunakan. Coba beberapa *tools* tersebut dan manfaatkan nantinya untuk siklus pengembangan aplikasi.