# A screenshot of a computer Description automatically generated with medium confidenceAXAR SCHOOL

Web Development Syllabus

# ABOUT

Created by : **Edo Novanto**

# TABLE OF CONTENTS

Contents

[AXAR SCHOOL 1](#_Toc80532679)

[ABOUT 2](#_Toc80532680)

[TABLE OF CONTENTS 3](#_Toc80532681)

[INTRODUCTION 5](#_Toc80532682)

[Description 5](#_Toc80532683)

[Prerequisite 6](#_Toc80532684)

[Requirement Tools 6](#_Toc80532685)

[Learning Outcome 6](#_Toc80532686)

[SYLLABUS 7](#_Toc80532687)

[Level 1 7](#_Toc80532688)

[Prerequisite 7](#_Toc80532689)

[Outcome 7](#_Toc80532690)

[Detail Syllabus 7](#_Toc80532691)

[Level 2 13](#_Toc80532692)

[Prerequisite 13](#_Toc80532693)

[Outcome 13](#_Toc80532694)

[Detail Syllabus 13](#_Toc80532695)

[Level 3 20](#_Toc80532696)

[Prerequisite 20](#_Toc80532697)

[Outcome 20](#_Toc80532698)

[Detail Syllabus 20](#_Toc80532699)

[ASSIGNMENTS 26](#_Toc80532700)

[Assignment I : Web Portofolio Siswa 26](#_Toc80532701)

[Kriteria 26](#_Toc80532702)

[Rekomendasi 27](#_Toc80532703)

[Ketentuan 27](#_Toc80532704)

[Assignment II : Web Notes Management (CRUD) 28](#_Toc80532705)

[Kriteria 28](#_Toc80532706)

[Rekomendasi 28](#_Toc80532707)

[Ketentuan 28](#_Toc80532708)

[Assignment III : Football App (Progressive Web Apps) 29](#_Toc80532709)

[Kriteria 29](#_Toc80532710)

[Rekomendasi 29](#_Toc80532711)

[Ketentuan 29](#_Toc80532712)

# INTRODUCTION

## Description

Web menjadi salah satu teknologi yang masif digunakan saat ini. Implementasinya beragam, mulai dari sarana pendidikan, media hiburan, hingga melakukan transaksi dapat dilakukan dengan mudah menggunakan web. Apabila ditarik ke belakang, perkembangan web dimulai pada tahun 1990 oleh Tim Berners-Lee. Saat itu Tim memperkenalkan 3 komponen fundamental web yang menjadi dasar teknologi web, diantaranya HyperText Markup Language (HTML), HyperText Transfer Protocol (HTTP), dan Uniform Resource Identifier (URI). Dari sinilah cikal bakal web yang kita kenal saat ini. Meskipun web termasuk teknologi yang sudah cukup lama diciptakan, bukan berarti web tidak relevan dan ketinggalan jaman. Tiap tahunnya, banyak perusahaan teknologi seperti Google hingga Huawei yang mengadakan konferensi untuk membahas teknologi-teknologi terbaru untuk diterapkan pada web.

Indonesia pun tidak ingin ketinggalan dalam memanfaatkan teknologi ini. Dilansir dari Indeed.com, gaji untuk seorang pengembang web (Web Developer) berada di angka 5-20 juta/bulan. Hal tersebut menunjukkan betapa dibutuhkannya peran seorang Web Developer dalam menunjang industri saat ini. Sayangnya, permintaan industri yang besar terhadap talenta digital saat ini belum dapat diakomodasi dengan baik. Menteri Komunikasi dan Informatika, Johnny G Plate menyebutkan, Indonesia tengah menghadapi kekurangan talenta digital yang andal di bidang TIK. Oleh sebab itu, diperlukan talenta-talenta digital yang mau belajar serta mengasah kemampuannya dalam bidang TIK khususnya Web Development.

Saat ini sumber pembelajaran, khususnya Web Development sudah banyak dan mudah ditemukan di internet. Namun, banyaknya sumber informasi yang ada terkadang membuat calon Web Developer menjadi kesulitan. Hal tersebut dikarenakan alur pembelajaran yang tidak berjenjang sehingga banyak konsep-konsep fundamental yang terlewatkan. Axar School, sebagai Ed-Tech company menghadirkan Web Development Syllabus yang dapat membantu siswa untuk mempersiapkan karirnya sebagai seorang Web Developer yang andal. Materi dalam silabus disusun berjenjang menjadi Level 1, Level 2, dan Level 3 untuk mempermudah pemahaman siswanya.

## Prerequisite

Silabus didesain berjenjang sehingga dibutuhkan pemahaman yang berbeda pada setiap levelnya. Detail pemahaman yang dibutuhkan dapat dilihat pada bagian [Syllabus](#_heading=h.4d34og8) untuk setiap levelnya.

## Requirement Tools

Agar dapat mengikuti pembelajaran dengan baik, diharapkan siswa memiliki beberapa peralatan sebagai berikut:

1. Spesifikasi minimal komputer atau laptop
   * Sistem operasi : Windows, Mac, Linux
   * Prosesor : Intel Pentium 4/Celeron
   * RAM : 2 GB
2. Perangkat Lunak
   * Teks Editor : VS Code (Rekomendasi), Sublime
   * Web Browser : Google Chrome , Firefox (Versi

mendukung ES6. Lihat [disini](https://www.w3schools.com/js/js_versions.asp) versi browser yang mendukung ES6)

* Uji REST API : Postman
* Version Control : GIT (Gitlab/Github)

3. Koneksi internet yang stabil

## Learning Outcome

Di akhir pembelajaran, siswa akan mampu menguasai kemampuan untuk menunjang karirnya sebagai *front-end*, *back-end*, bahkan *fullstack developer*. Detail kemampuan teknis yang dapat dikuasai dapat dilihat pada bagian [Syllabus](#_heading=h.4d34og8) untuk setiap levelnya.

# SYLLABUS

## Level 1

### Prerequisite

Silabus didesain bagi siswa yang telah memiliki kemampuan dasar dalam mengoperasikan komputer. Apabila terpenuhi, maka siswa dinyatakan dapat mengikuti materi dalam silabus Level 1 ini tanpa harus memahami pemrograman sebelumnya. Mengingat konsep Web Development yang begitu luas dan tidak seluruhnya dapat dirangkum di dalam silabus Level 1, maka siswa didorong untuk memiliki rasa ingin tahu yang tinggi pada setiap materinya. Siswa dapat mempelajari referensi-referensi pembelajaran yang akan diberikan pada setiap materinya untuk memperkaya pengetahuan siswa.

### Outcome

Pada akhir kelas, siswa diharapkan mampu menguasai beberapa hal berikut:

* Memahami teknologi web dan perkembangannya
* Mampu membangun aplikasi web yang interaktif serta memiliki fitur penyimpanan
* Mengerti konsep pemrograman Javascript untuk pengembangan web pada lingkungan browser dan Node

### Detail Syllabus

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Day** | **Topics** | **Notes** |
| **I :**  Web & CSS Introduction | 1 | * Pengenalan Web (sejarah, istilah-istilah penting, anatomi, cara kerja, dan perkembangan) * Pengenalan karir Web Developer (nama posisi, spesialisasi, prediksi, gaji) | * *Siswa diminta untuk melakukan instalasi software yang akan digunakan selama pembelajaran* * *Siswa masuk ke Group WA/Slack untuk mendukung pembelajaran asinkron* * *Siswa diminta untuk mengeksplorasi software yang telah diinstal terlebih dahulu.* |
| 2 | * Pengenalan HTML (pembuatan berkas, elemen pembentuk, atribut, semantic element, melakukan eksekusi) * **QUIZ I** : Pengenalan Web dan HTML | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| 3 | * Pengenalan CSS (fungsi, aturan penulisan, jenis penerapan, pemberian komentar, conception) * Praktik menerapkan CSS pada halaman Web | * *Diberikan source code berisi file HTML, siswa diminta untuk menerapkan CSS ke dalamnya* |
| **II :**  Drilling CSS #1 | 4 | * Pendalaman CSS ( jenis selector, combinators, pseudo, font + text styling) * Praktik menggunakan *external font* (Google Font) |  |
| 5 | * Pengenalan fungsi warna pada web * Implementasi warna dalam CSS (Predefined, numeric value, opacity) * Instalasi Extension yang membantu pemilihan warna (Color Picker) |  |
| 6 | * Box Element (Padding, border, margin) * Box Dimension (Fixed & Responsive) * Styling Box * **EXERCISE I** : Membuat website portofolio sederhana menggunakan HTML + CSS | * *Exercise dikumpulkan dalam bentuk folder berisi source code melalui Google Form* |
| **III :**  Drilling CSS #2 | 7 | * Best practice penerapan gambar (size, position, format) * Membuat card & header sederhana |  |
| 8 | * Pengenalan positioning (Normal-Static Flow, relative, absolute, fixed) * Penerapan Floating |  |
| 9 | * Pengenalan responsive web design (Mengapa penting, cara penerapan) * Pendekatan pembuatan web “Mobile First VS Desktop First” |  |
| **IV :**  Drilling CSS #3 | 10 | * Penerapan media query pada CSS (Mobile, Tablet, dan Desktop) * Fixing error CSS menggunakan automation tools |  |
| 11 | * Pengenalan Flexbox layout (Flex container, grow, direction) * **EXERCISE II** : Membuat layout kalkulator sederhana. | * *Exercise diunggah ke akun Github/Gitlab. Kemudian link unggah dikirimkan ke Google Form.* |
| 12 | * Pengenalan framework CSS * Penerapan framework Bootstrap (CDN, Module) |  |
| **V :**  Javascript Introduction | 13 | * Pengenalan fundamental Javascript (Sejarah, alasan dipelajari, fungsi) * Pengenalan lingkungan pengembangan Javascript (*Dibahas lebih detail pada Week IX*) | * *Siswa diberikan referensi materi pendukung Javascript secara asinkron* |
| 14 | * Penulisan syntax dasar Javascript (tipe variable, tipe data, operator, else/if statement, comments) * Scope variable |  |
| 15 | * Mendalami syntax Javascript (Switch, Loop, Function) * **QUIZ II** : Logika Javascript | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **VI :**  Javascript Data Structure | 16 | * Pengenalan Struktur Data 1 (object, array) * Latihan object & array |  |
| 17 | * Pengenalan Struktur Data 2 (spread operator, destructuring object & array) * Latihan spread operator, destructuring object & array |  |
| 18 | * Pengenalan Struktur Data 3 (map, set, weakMap, weakSet) * Latihan map & set * **QUIZ III** : Data Structure | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **VII :**  Function | 19 | * Pengenalan function (pengertian, fungsi, pembuatan) * Mendeklarasikan fungsi (parameter, argumen, expression function) |  |
| 20 | * Arrow function dan kelebihannya * Variable scope pada function |  |
| 21 | * Penerapan Closure pada function * **QUIZ IV** : Function | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **VIII :**  OOP | 22 | * Pengenalan paradigma Object Oriented Programming (OOP) * Pengenalan Class (instance, properti, method/function) |  |
| 23 | * Konsep-konsep OOP (enkapsulasi, abstraksi, pewarisan, polimorfisasi) * Latihan konsep-konsep OOP |  |
| 24 | * Overriding Method * Object Composition * Built-in Class * **QUIZ V** : OOP | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **IX :**  Javascript Environment | 25 | * Pendalaman Javascript environment (text editor, terminal) * Runtime environment Browser VS Node.js |  |
| 26 | * Pengenalan Node.js * Instalasi Node.js |  |
| 27 | * Membuat & menjalankan proyek JS dengan Node.js * **EXERCISE III** : Membuat proyek Node.js sederhana | * *Exercise diunggah ke akun Github/Gitlab. Kemudian link unggah dikirimkan ke Google Form.* |
| **X :**  DOM | 28 | * Pengenalan DOM (DOM Tree, menggunakan DOM) * Manipulasi DOM |  |
| 29 | * Pengenalan macam-macam event pada web (windows, form, keyboard, clipboard, mouse) * Pendalaman event Form (onSubmit, onInput, onChange, dll) |  |
| 30 | * **EXERCISE IV** : Membuat form sederhana dengan event | * *Exercise diunggah ke akun Github/Gitlab. Kemudian link unggah dikirimkan ke Google Form.* |
| **XI :**  Web Storage | 31 | * Pengenalan Web Storage (pengertian, fungsi, dan macamya) * Pengenalan Cookies pada Web |  |
| 32 | * Tipe Web Storage Web API (Session & Local) * Implementasi Local Storage |  |
| 33 | * Pendalaman local storage dengan studi kasus * **QUIZ VI** : Web Storage | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **XII :**  Summary | 34 | * Penyegaran materi HTML dan CSS * **GAME I** : HTML dan CSS | * *Game dilakukan menggunakan Kahoot* |
| 35 | * Penyegaran materi Javascript (data structure, function, OOP, environment) * **GAME II** : Kahoot Javascript | * *Game dilakukan menggunakan Kahoot* |
| 36 | * Penyegaran materi DOM & Web Storage * Penjelasan materi lanjutan (Level II) * [**ASSIGNMENT I**](#_heading=h.3whwml4) : Web Portofolio Siswa |  |

## Level 2

### Prerequisite

Silabus didesain bagi siswa yang telah memiliki kriteria sebagai berikut:

* Siswa yang sudah mengusasi dasar-dasar teknologi informasi dan mampu mengoperasikan komputer dengan baik
* Siswa sudah familiar dengan Basic Web Development (HTML, CSS, Javascript)

### Outcome

Pada akhir kelas, siswa diharapkan mampu menguasai beberapa hal berikut:

* Membuat aplikasi frontend dengan standar ES6
* Mampu menerapkan web components, webpack, dan memperoleh data dinamis dari API
* Membuat RESTful API sederhana yang mendukung CRUD

### Detail Syllabus

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Day** | **Topics** | **Notes** |
| **I :**  ECMA Scripts 6 (ES6) & Object Oriented Programming (OOP) | 1 | * Pengenalan ECMA Scripts 6 * Deklarasi dan aturan penggunaan Variabel (Var, Let, Const) * Template Literals * Destructuring Object dan Array * Spreading Operator dan Rest Parameter |  |
| 2 | * Pengenalan Function * Arrow Function Expression * Perbedaan Regular Function dan Arrow Function * Parameter & Block Body (Arrow Function) * This keyword (Arrow Function) |  |
| 3 | * Default Parameters * Pengenalan Class (Javascript) * Property & Method * 4 Pilar Object Oriented Programming (OOP) * Constructor * Instance * Property Accesor |  |
| **II** :  OOP II & Concurrency | 4 | * Method * Inheritance * Static Method * Overriding Method * Object Composition * Built in Class |  |
| 5 | * Synchronous VS Asynchronous * Fungsi setTimeout() * Callback Function * Callback Hell |  |
| 6 | * Pengenalan Promise * Membuat Objek Promise * Fungsi onFulfilled dan onRejected * Method catch() |  |
| **III:**  Concurrency II & Module | 7 | * Promise Berantai * Promise All * Sintaks dan Penggunaan Async/Await * Menangani onRejected Menggunakan Async/Await * Promise Berantai dengan Async/Await |  |
| 8 | * Pengenalan Module * Export & Import Module (Node.js) * Export banyak Module (Node.js) |  |
| 9 | * ES6 Module * Export & Import Module (ES6 Module) * **QUIZ** : Module | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **IV:**  Web Components | 10 | * Pengenalan Web Components * Custom Elements * Siklus Hidup Custom Elements * Atribut & Method Custom Elements * Mendapatan Atribut Custom Elements |  |
| 11 | * Menetapkan Styling Custom Elements * Data Kompleks pada Custom Elements * Nested Custom Elements | * *Latihan Menerapkan Custom Elements* |
| 12 | * Pengenalan Shadow DOM * Dasar Pengunaan Shadow DOM * Pengunaan Shadow DOM pada Web * **QUIZ** : ES6 & Web Components | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **V:**  Node Package Manager (NPM) | 13 | * Penganalan NPM * Menginstall Node.js dan NPM * Menggunakan NPM * Project Starter Menggunakan NPM |  |
| 14 | * Memasang (Install) External Package * Pengenalan berkas package.json * Menggunakan (Using) External Package * Menjalankan Runner Scripts |  |
| 15 | * Menghapus (Uninstall) Package * Cakupan Package * **QUIZ** : Node Package Manager | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **VI:**  Webpack | 16 | * Pengenalan Webpack * Webpack Core Concepts * Entry * Output * Loaders |  |
| 17 | * Plugin * Mode * Memasang & Menggunakan Webpack * Membuat Berkas Konfigurasi Webpack |  |
| 18 | * Latihan Menggunakan Laoder dan Plugin * Webpack Dev Server * Konfigurasi Environement Webpack (Production VS Development) * **QUIZ** : NPM & Webpack | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **VII:**  Async Javascript Request | 19 | * Pengenalan Async Javascript Request * HTTP Request * Web APIs * CORS * Menggunakan Postman |  |
| 20 | * Javascript Object Notation (JSON) * AJAX menggunakan XHR * Header dan Body request dengan XHR |  |
| 21 | * Pengenalan AJAX Menggunakan Fetch * Dasar Penggunaan Fetch * Latihan AJAX menggunakan Fect * **QUIZ** : Asynchronous Javascript Request | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| **VIII:**  Pengenalan Backend | 22 | * Apa itu Backend? * Server sebagai perantara Frontend dan Backend * Web Server & Web Service * Komunikasi Client-Server | * *Diisi dengan latihan membuat request/ permintaan HTTP* |
| 23 | * REST Web Service * REST API * Format Request dan Response |  |
| 24 | * HTTTP Verbs * Response Code * URL Design |  |
| **IX:**  Dasar Node.js | 25 | * Dasar Node.js untuk Back-end * Pengenalan detail Node.js * Menjalankan Javascript dengan Node.js | * *Diisi dengan latihan membuat proyek Node.js* |
| 26 | * Node.js Global Object * Process Object * Modularization | * *Diisi dengan latihan Process Object* * *Diisi dengan latihan Modularization* |
| 27 | * Events * Filesystem * Readable Stream * Writeable Stream |  |
| **X:**  Membangun Web Service (Node.js) | 28 | * Pendahuluan membangun Web Service menggunakan Node.js * Membangun Web Server secara native * Membuat HTTP Server | * *Diisi dengan latihan membuat HTTP Server* |
| 29 | * Method/Verb Request * Body Request * Routing Request |  |
| 30 | * Response Status * Response Header * Response Body |  |
| **XI:**  Membangun Web Service (Node.js) 2 | 31 | * Pengenalan Web Framework * Web Framework pada Node.js * Persiapan membangun web server menggunakan Hapi | * *Diisi dengan persiapan project dengan framework Hapi* |
| 32 | * Membuat HTTP Server dengan Hapi * Method/Verb Request dan Routing dengan Hapi * Path Parameter | * *Diisi dengan latihan path parameters* |
| 33 | * Query Parameters * Body/Payload Request * Response Toolkit | * *Diisi dengan latihan query parameters* |
| **XII:**  Summary | 34 | * Penyegaran materi ES6 & OOP * **GAME** : Kahoot ES6 + OOP | * *Game dilakukan menggunakan Kahoot* |
| 35 | * Penyegaran materi Concurrency, Module, dan Web Components * **GAME** : Kahoot Concurrency, Module, dan Web Components | * *Game dilakukan menggunakan Kahoot* |
| 36 | * Penyegaran materi NPM, Webpack, dan Backend * Penjelasan materi lanjutan (Level III) * [**ASSIGNMENT II**](#_Assignment_II_:) **:** Web Notes Management (CRUD) |  |

## Level 3

### Prerequisite

Silabus didesain bagi siswa yang telah memiliki kriteria sebagai berikut:

* Siswa yang sudah mengusasi dasar-dasar teknologi informasi dan mampu mengoperasikan komputer dengan baik
* Siswa sudah familiar dengan Basic Web Development (HTML, CSS, Javascript)
* Siswa sudah familiar dengan konsep Module Bundler (Ex: Webpack)
* Siswa sudah memahami konsep RESTful API dan penggunaannya

### Outcome

Pada akhir kelas, siswa diharapkan mampu menguasai beberapa hal berikut:

* Mampu membuat aplikasi yang memiliki sifat Native
* Membangun aplikasi yang memiliki aksesibilitas yang baik, mudah di-maintenance, dapat diuji, dan memiliki performa yang baik

### Detail Syllabus

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Day** | **Topics** | **Notes** |
| **I :**  Mobile First Approach | 1 | * Mengenal Mobile First Approach * Mengatur Viewport * Relative Size * Target Touch Size | * *Diisi dengan latihan mengatur viewport, penggunaan relative unit, dan ukuran element* |
| 2 | * Responsive Layout (Desktop First Approach VS Mobile First Approach) * Media Query * Breakpoint |  |
| 3 | * Grid Layout Module * Grid Tracks * Fraction Unit * Repeat Notation |  |
| **II :**  Mobile First Approach II & Accessibilty | 4 | * Grid Lines * Gutters * Text Optimization * Off Canvas Menu | * *Diisi dengan latihan menerapkan Responsive Layout* * *Diisi dengan latihan menerapkan Text Optimization dan Off Canvas Menu* |
| 5 | * Pengenalan Accessibility (Pengelihatan, motorik, pendegaran, kognitif) * Aksesibilitas Keyboard (onFocus()) * Focusable Element * Penggunaan Achor VS Button |  |
| 6 | * Focus Order * Tab Index * Styling Focus menggunakan Pesudo-class * Aksesibilitas Konten |  |
| **III :**  Accessibilty II & Javascript Clean Code | 7 | * Affordances dan Semantics * Pengenalan Screen Reader (ChromeVox, Narrator) * Mencoba Screen Reader * **QUIZ** : Mencoba Screen Reader | * *Kuis dilakukan menggunakan Google Form dalam jangka waktu tertentu* |
| 8 | * Skip to Content * Heading dan Landmarks * Label * Alternative Text |  |
| 9 | * Pengenalan Javascript Clean Code * Style Guide dan pentingnya menggunakan Style Guide * Code Convention (AirBnB, Google, Standard Javascript) |  |
| **IV:**  Javascript Clean Code II | 10 | * Pengenalan ESLint * Menggunakan ESLint * Memasang ESLint * Mengintegrasikan ESLint pada Visual Studio Code | * *Latihan Mengaudit Kode menggunakan ESLint* |
| 11 | * Prinsip membuat Variables yang baik * Prinsip membuat Function yang baik * Pendekatan pembuatan Function yang perlu dihindari |  |
| 12 | * Objek & Struktur Data * Classes * Method Chaining |  |
| **V:**  Prinsip SOLID | 13 | * Penganalan Prinsip SOLID (Single Responsibility Principle, Open/Closed Principle) * Single Responsibility Principle (SRP) * Open/Closed Principle (OCP) |  |
| 14 | * Liskov Substitution Principle (LSP) * Interface Segregation Principle (ISP) * Dependencies Inversion Principle (DIP) |  |
| 15 | * Concurrency * Error Handling * Prinsip Penulisan Comments |  |
| **VI:**  Progressive Web Apps (PWA) | 16 | * Pengantar & Pengenalan PWA * Kelebihan & Kekurangan PWA * Komponen-Komponen Pembentuk PWA |  |
| 17 | * Web Architecture * Application Shell | * *Diisi dengan Latihan membuat Application Shell* |
| 18 | * Web App Manifest * Service Worker dan kegunaannya * Registrasi Service Worker * Siklus hidup Service Worker (Installation, Activate, Idle) |  |
| **VIII:**  PWA 2 | 19 | * Pengenalan Event pada Service Worker * Event Fetch * Event Message * Event Sync * Event Push | * *Latihan menerapkan Service Worker* |
| 20 | * Cache API * Dasar Penggunaan Cache API * Menambahkan Resources ke Cache (add & addAll) * Mengambil Resource dari Cache |  |
| 21 | * Mencari Resource dari Cache * Menghapus Resource dari Cache * Caching Strategies * Menerapkan Cache pada service worker | * *Latihan menerapkan Cache API* |
| **IX:**  PWA 3 | 22 | * Pengenalan IndexedDB * Istilah dalam IndexedDB * Dasar Pengunaan IndexedDB * Membuat/Membuka Database dan Object Store |  |
| 23 | * Operasi CRUD pada IndexedDB * Menyiapkan Fungsi Operations IndexedDB * Menampilkan data dari IndexedDB | * *Latihan menerepkan IndexedDB* |
| 24 | * Pengenalan Web Socket * Membuka Koneksi Web Socket * Mengirim Data ke WebSocket Server * Mendapatkan Data dari WebSocket Server * Menutup Koneksi WebSocket | * *Latihan menerapkan Web Socket* |
| **X:**  Automation Testing | 25 | * Pengenalan Automation Testing * TDD Menggunakan Jasmine | * *Latihan menerapkan Automation Testing menggunakan Jasmine* |
| 26 | * Pengenalan Automation Testing menggunakan KarmaJS * Menginstall dan melakukan Automation Testing menggunakan KarmaJS | * *Latihan menerapkan Automation Testing menggunakan KarmaJS* |
| 27 | * Penerapan Prinsip Dependency Inversion pada Automation Testing | * *Latihan menerapkan Automation Testing dengan prinsip dependency inversion* |
| **XI:**  Automation Testing 2 & Web Performance | 28 | * Menambah fungsionalitas menggunakan Test Driven Development (TDD) | * *Latihan menerapkan TDD* |
| 29 | * Pengenalan End-to-End Testing * Melakukan End-to-End Testing | * *Latihan menerapkan End-to-End Testing* |
| 30 | * Pengenalan Web Performance * Pentingnya kecepatan pada Web * Pengenalan Core Web Vitals |  |
| **XII:**  Web Performance 2 | 31 | * LCP (Largest Contentful Paint) * FID (First Input Delay) * CLS (Cumulative Layout Shift) | * *Latiham mengukur LCP, FID, dan CLS* |
| 32 | * Image Optimization * GIF to Video Transformation * Image Responsive * WebP Image Format | * *Latihan menerpakan iamge optimizaion, GIF to Video Transformation, Image Responsive, dan WebP Image Format* |
| 33 | * Menerapkan Lazy Laoding Image * Optimisasi Bundle * Bundle Analyzer * Code Splitting * [**ASSIGNMENT III**](#_Assignment_III_:) : Football Apps (PWA) | * *Latihan melakukan optimasi bundle dan code splitting* |

# ASSIGNMENTS

## Assignment I : Web Portofolio Siswa

### Kriteria

* Menerapkan semantic element <header>, <footer>, <main>, <article>, dan <aside> pada berkas HTML.
* Menerapkan styling external menggunakan CSS
* Mendukung responsive web menggunakan Media Query (Phone, Tablet, dan Desktop)
* Menampilkan biodata diri disertai gambar siswa
* Melakukan deploy web menggunakan Github hosting.

Berikut contoh sketsa Web Portofolio yang dapat dijadikan referensi

A screenshot of a computer

Description automatically generated with medium confidence

Navigation

Header

Main

Footer

### Rekomendasi

* Tampilan desain web portofolio dibuat sesuai kretifitas siswa namun mengikuti kaidah-kaidah *user interface* yang baik. (Pemilihan warna, tata letak, ukuran font, dll)
* Memanfaatkan DOM dan Web Storage pada web.
* Diperkenankan untuk menggunakan framework CSS seperi Bootstrap, Materialize, dll.

### Ketentuan

* Tugas dikirimkan dalam bentuk link website portofolio dan Github/Gitlab
* Github/Gitlab berisi source code HTML, CSS, Javascript, serta aset yang digunakan dalam website

## Assignment II : Web Notes Management (CRUD)

### Kriteria

* Menggunakan standar ES6 sintaks dalam menuliskan kode JavaScript
* Terdapat minimal satu penerapan custom element
* Menggunakan Webpack sebagai module bundler
* Memanfaatkan API dengan menggunakan konsep AJAX dalam menampilkan data yang dinamis pada aplikasi yang dibuat

Berikut contoh Web Notes Management yang dapat dijadikan referensi:

|  |
| --- |
| [**https://youtu.be/rJDxYfQZroQ**](https://youtu.be/rJDxYfQZroQ) |

### Rekomendasi

* Menerapkan tampilan aplikasi yang menarik
* Menggunakan package pihak ke-3 yang di-install melalui package manager (Axios, Lodash, dll)
* Menuliskan kode dengan bersih dan konsisten

### Ketentuan

* Tugas dikirimkan dalam bentuk link website notes management (CRUD) dan Github/Gitlab
* Github/Gitlab berisi source code HTML, CSS, Javascript, serta aset yang digunakan dalam website

## Assignment III : Football App (Progressive Web Apps)

### Kriteria

* Aplikasi dapat diakses dalam keadaan offline tanpa ada aset yang gagal dimuat, termasuk data yang didapatkan dari API.
* Aplikasi harus menampilkan icon Add to Home Screen.
* Menggunakan ESLint sebagai linter ketika menuliskan kode JavaScript.
* Menerapkan responsibilitas tampilan, aksesibilitas pada website, appbar, footer dan sebagainya.
* Menerapkan Integration dan End-to-End Test
* Menerapkan Image Compression
* Memasang bundle analyzer

Berikut contoh Progressive Web Apps yang dapat dijadikan referensi:

|  |
| --- |
| [**https://football-419cd.web.app/**](https://football-419cd.web.app/) |

### Rekomendasi

* Menerapkan Web Component secara native menggunakan Custom Element
* Tambahkan indikator loading ketika data sedang dimuat dari server
* Menampilkan keterangan/halaman bila request yang dilakukan gagal.
* Menuliskan test case secara lengkap, mulai dari kasus positif hingga kasus negatif. Pastikan test case yang dijalankan berhasil.
* Mendeploy hasil web app yang Anda kerjakan ke third-party hosting. Seperti Firebase Hosting, Netlify, Heroku, dll.

### Ketentuan

* Tugas dikirimkan dalam bentuk link website notes management (CRUD) dan Github/Gitlab
* Github/Gitlab berisi source code HTML, CSS, Javascript, serta aset yang digunakan dalam website