**LAPORAN PEMROGRAMAN FRAMEWORK**

**“SESSION 2 Modern JavaScript”**



**Oleh:**

**Nama : Bagus Satria Putra**

**Kelas : 3F**

**Absen/NIM : 8 / 1841720146**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**16 FEBRUARI 2021**

|  |  |
| --- | --- |
| **No.** | **Keterangan** |
| **a.** | Membuat variabel menggunakan const |
| * 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.   const.html  **Jawaban :**  const.js  **Jawaban :**     * 1. Selanjutnya buka file const.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.   **Jawaban :**    **Penjelasan :**  Variable **const** akan mengijikan untuk melakukan perubahan data sesuai dengan method yang telah disediakan oleh array maupun object dan ini disebut dengan mutable. |
| **b.** | Membuat variabel menggunakan let  1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.   let.html  **Jawaban :**  let.js Jawaban : 2. Selanjutnya buka file let.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.  **Jawaban :**    **Penjelasan :**  fitur baru yang diperkenalkan oleh ES6 yang pada dasarnya pengembangan dari var.  Perbedaannya ada pada scope, apabila var cangkupannya adalah **function scope**, bila berada diluar dari fungsi akan dinyatakan sebagai global objek. Berbeda halnya dengan let yang cangkupannya adalah **block scope.**  **Block scope** dengan menggunakan let yaitu cangkupanya terbatas pada block tempat variable let didefenisikan. Sehingga dengan konsep block scope yang dianut oleh let dapat menjadi solusi atas permasalahan yang dialami oleh var. |
| **C.** | Membuat Template Strings  1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.   template.html  **Jawaban :**    template.js  **Jawaban :**  2.Selanjutnya buka file template.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.  **Jawaban :**    **Penjelasan :**  menggabungkan teks dan variable dengan tampilan yang lebih rapi. Prinsip penggunaan template literal adalah   1. Dimulai dan diakhir dengan ` 2. Penggunaan ${} untuk menuliskan suatu variable atau javascript statement lainnya |
| **d.** | Membuat default parameters  1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.   defaultParameters.html  **Jawaban :**    defaultParameters.js    2. Selanjutnya buka file defaultParameters.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.   Penjelasan :Default parameter merupakan fitur yang bisa dimanfaatkan untuk membuat fungsi kita lebih fleksibel, karena cara pemanggilan fungsi dapat dirancang dengan lebih sederhana. Untuk merancang fungsi yang lebih rumit, kita bisa membuat jumlah argumen yang tidak terbatas. |
| **e.** | Membuat Arrow Function 1  * 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.   arrow.html    arrow.js    2. Selanjutnya buka file arrow.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.    **Penjelasan :**  sintak penulisan fungsi yang bisa dibilang lebih singkat, menggunakan token baru yaitu “=>”, fungsi yang dideklarasikan menggunakan panah ini bersifat anonim. Mirip lambda arrow operator di bahasa lain. Dengan arrow function kita bisa menulis fungsi tanpa menggunakan kata kunci “function”, “return”, dan kurung kurawal. Oke mari kita langsung saja membahas sedikit. |
| **F.** | Membuat Arrow Function 2  * 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.   arrowf.html    arrowf.js     * 1. Selanjutnya buka file arrowf.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.     **Penjelasan :**  jika parameter yang dibutuhkan hanya 1 maka kita bisa meringkasnya kembali dengan menghilangkan tanda kurung pada baris kode tersebut.  namun jika kita tidak memiliki parameter yang dibutuhkan maka kita teap harus menggunakan tanda kurung agar block kodenya dapat berjalan dengan benar. |
| **G.** | Membuat Destructuring Object  * 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.   destructuring.html    destructuring.js     * 1. Selanjutnya buka file destructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.     **Penjelasan :**  'membongkar' isi array dan objek untuk disimpan pada beberapa variabel. Dengan menggunakan fitur ini, kita dapat menghemat kode secara signifikan. Dalam object destructuring, nama key pada sisi kiri pernyataan harus match dengan nama key pada objek sumber |
| **H.** | Membuat Destructuring an Array  * 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.   array.html    array.js    2. Selanjutnya buka file array.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.    3. Kemudian ganti sedikit code pada array.js seperti dibawah ini  4. Bukalah kembali, maka hasilnya akan berbeda seperti gambar dibawah ini    **Penjelasan :**  dalam Javascript akan mengembalikan undefined ketika kita mencoba mengakses nilai di luar batasnya (index out of bounds). Konsekuensinya, jika panjang array yang kita deklarasikan melebihi panjang array sumbernya, kita dapat temukan variabel yang kita assign memiliki nilai undefined. Untuk menanggulanginya, kita dapat memberikan nilai default bagi variabel yang akan di-assign melalui destructuring.  Seperti array destructuring, kita juga dapat melakukan destructuring terhadap objek tanpa melakukan deklarasi pada baris yang sama. Perbedaannya adalah **kita harus membungkusnya dengan sepasang tanda kurung** agar pernyataan pada sisi kiri ({ …, …}) tidak dianggap sebagai blok; |
| **i.** | Membuat Restructuring 1.Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.  restructuring.html  restructuring.js    2.Selanjutnya buka file restructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.    **Penjelasan :**  fungsi yang memanggil dirinya sendiri. Fungsi **restructuring** sering saya bayangkan seperti perulangan. Karena tingkah lakunya yang mengulang-ulang setiap pemanggilan dirinya |
| **J.** | Membuat Spread and Rest operator 1.Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.  spread.html    spread.js    2.Selanjutnya buka file spread.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.    3.Untuk melakukan praktek rest operation, buatlah dua file .html dan .js seperti dibawah ini  restO.html  restO.js    4.Selanjutnya buka file restO.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.    **Penjelasan :**  Rest Parameter ini berguna untuk menggabungkan semua paramater pada function ke dalam array. Dengan menggunakan Rest Parameter ini dapat membantu kita mendefinisikan function dengan rapi serta memberikan parameter yang tidak terbatas pada sebuah function.  Spread Operator digunakan untuk membagi elemen array atau properti pada objek, sehingga elemen array dapat ditambahkan/dimasukan ke dalam array baru. |
| **K.** | Membuat Classes Constructor and Super 1.Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.  class.html    class.js    2.Selanjutnya buka file class.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.    3.Klik OK, maka akan muncul page selanjutnya dengan pesan sebagai berikut.    **Penjelasan :**  Hanya boleh ada satu metode spesial dengan nama "constructor" dalam sebuah kelas. [SyntaxError](https://developer.mozilla.org/id/docs/Web/JavaScript/Reference/SyntaxError)  akan dilemparkan jika suatu kelas memiliki lebih dari satu metode konstruktor.  Konstruktor dapat menggunakan kata kunci [super](https://developer.mozilla.org/id/docs/Web/JavaScript/Reference/Operators/super) untuk memanggil konstruktor orang-tua.  Jika metode konstruktor tidak didefinisikan dalam sebuah kelas, maka  kontruktor asali yang akan digunakan. |

***--Alhamdulillah--***