

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN BERBASIS WEB**  
**“Kalkulator Dengan JavaScript”**



Disusun Oleh :

Nama : I Kadek Gowinda

Nim : 1808561033

Kelas : B

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS UDAYANA**  
**BADUNG**  
**2020**

## KATA PENGANTAR

Puji dan syukur ke hadirat Tuhan Yang Maha Kuasa atas segala rahmat yang diberikan-Nya sehingga tugas Laporan Praktikum 06 yang berjudul "*Kalkulator Dengan JavaScript*" ini dapat saya selesaikan. Laporan ini saya buat sebagai kewajiban untuk memenuhi tugas. Dalam kesempatan ini, penulis menghaturkan terimakasih yang dalam kepada semua pihak yang telah membantu menyumbangkan ide dan pikiran mereka demi terwujudnya makalah ini. Akhirnya saran dan kritik pembaca yang dimaksud untuk mewujudkan kesempurnaan makalah ini penulis sangat hargai.

Gianyar, 16 November 2020

I Kadek Gowinda

## DAFTAR ISI

KATA PENGANTAR .....	ii
DAFTAR ISI.....	iii
BAB I.....	1
PENDAHULUAN.....	1
1.1    Latar Belakang .....	1
1.2    Tujuan .....	1
1.3    Manfaat .....	1
BAB II .....	2
KAJIAN PUSTAKA .....	2
2.1    HTML.....	2
2.2    JavaScript.....	2
2.3    Implementasi JavaScript.....	2
BAB III.....	4
PEMBAHASAN .....	4
3.1    Program Kalkulator .....	4
3.2    Source HTML .....	5
3.3    Source CSS .....	6
3.4    Source JavaScript .....	8
BAB IV .....	12
PENUTUP.....	12
4.1. Simpulan .....	12
4.2. Saran .....	12
DAFTAR PUSTAKA .....	13

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Sebuah website yang menarik tentunya tak lepas dari sifat interaktif serta dinamis yang dimiliki website tersebut. Semakin menarik dan semakin nyaman pengguna dalam website tentunya akan memberikan pengaruh besar terhadap durasi serta frekuensi waktu pengguna di dalam website.

Menggunakan JavaScript menjadi salah satu pilihan untuk membuat suatu website yang dinamis dan interaktif. JavaScript merupakan Bahasa pemrograman yang berdasarkan pada Bahasa pemrograman java, namun lebih sering dipakai untuk pembuatan website Bersama dengan HTML dan CSS. JavaScript dapat dipakai dalam format file atau ada didalam file HTML sebagai internal Script. Jika berada didalam file HTML, penggunaan JavaScript akan berada didalam tag `<script>`.

Pembuatan sebuah kalkulator sederhana akan memberikan pengalaman pemula yang bagus bagi seorang programmer. Dengan mencoba membuat sesuatu yang sederhana, kita dapat mempelajari beberapa hal-hal dasar dan mengulangi hal tersebut akan menambah keahlian dimasa depan.

### **1.2 Tujuan**

- 1.2.1 Mengetahui dasar-dasar Bahasa pemrograman Javascript.
- 1.2.2 Membuat sebuah kalkulator sederhana yang berfungsi dengan baik.
- 1.2.3 Melakukan implementasi dasar Bahasa pemrograman JavaScript.

### **1.3 Manfaat**

- 1.3.1 Memiliki pemahaman dasar tentang Bahasa pemrograman JavaScript.
- 1.3.2 Memiliki kemampuan untuk membuat sebuah program dengan JavaScript.
- 1.3.3 Memiliki pengalaman awal menggunakan Bahasa pemrograman JavaScript.

## BAB II

### KAJIAN PUSTAKA

#### 2.1 HTML

HTML(Hypertext Markup Language) bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan user untuk mengorganisir dan memformat dokumen, sama seperti Microsoft Word.

HTML menggunakan struktur kode yang sederhana (tag dan *attribute*) untuk markup halaman website. Misalnya, pembuatan sebuah paragraf dengan menempatkan *enclosed text* di antara tag pembuka `<p>` dan tag penutup `</p>`. HTML sebagai bahasa markup sangatlah mudah untuk dipahami bahkan bagi webmaster pemula di bidang web development sekalipun.

#### 2.2 JavaScript

JavaScript dibuat dan didesain selama sepuluh hari oleh Brandan Eich, seorang karyawan Netscape, pada bulan September 1995. Awalnya bahasa pemrograman ini disebut Mocha, kemudian diganti ke Mona, lalu LiveScript sebelum akhirnya resmi menyandang nama JavaScript. Cara penulisan JavaScript mirip dengan penulisan bahasa pemrograman web lainnya seperti **PHP** dan **CSS**, yakni dengan menyisipkan kode JavaScript di dalam HTML.

#### 2.3 Implementasi JavaScript

```
<script type="text/javascript">
  <!--
      Kode-kode JavaScript
  // -->
</script>
```

Diatas merupakan cara pengimplementasian JavaScript jika diletakkan didalam file HTML. Code-code yang akan digunakan ditaruh didalam tag `<script>`.

Namun cara ini sedikit kurang efektif untuk sebuah website yang rumit dengan code yang panjang. Kebanyakan orang akan merekomendasikan untuk menggunakan JavaScript sebagai file external sehingga code lebih terstruktur.

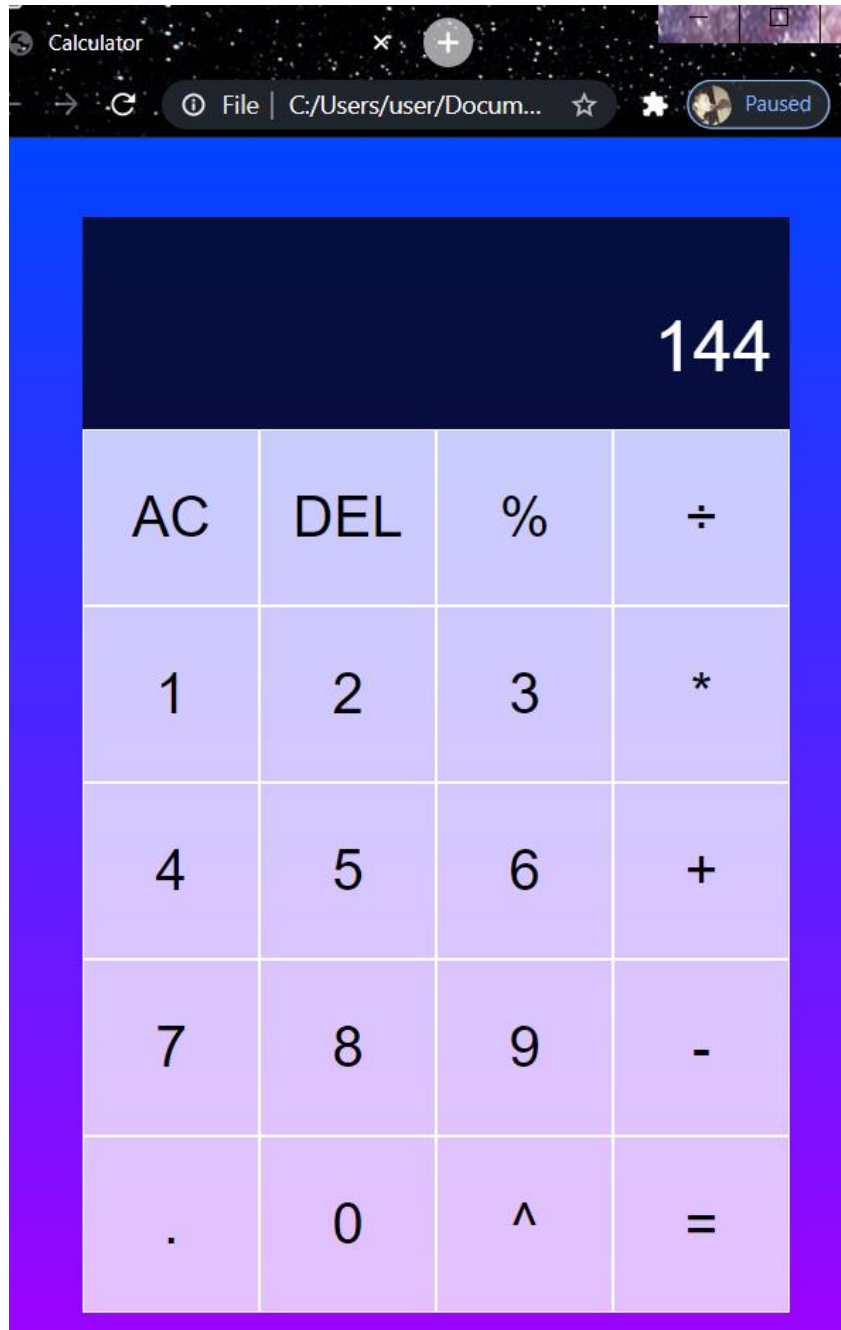
Penggunaan file eksternal dapat dilakukan dengan membuat sebuah file dengan tipe “file.js” didalam folder projek yang sama dengan file HTML. Lalu file eksternal ini akan dipanggil dengan atribut “src” pada tag <script>. Contohnya

```
<script src="script.js" defer></script>
```

## BAB III

### PEMBAHASAN

#### 3.1 Program Kalkulator

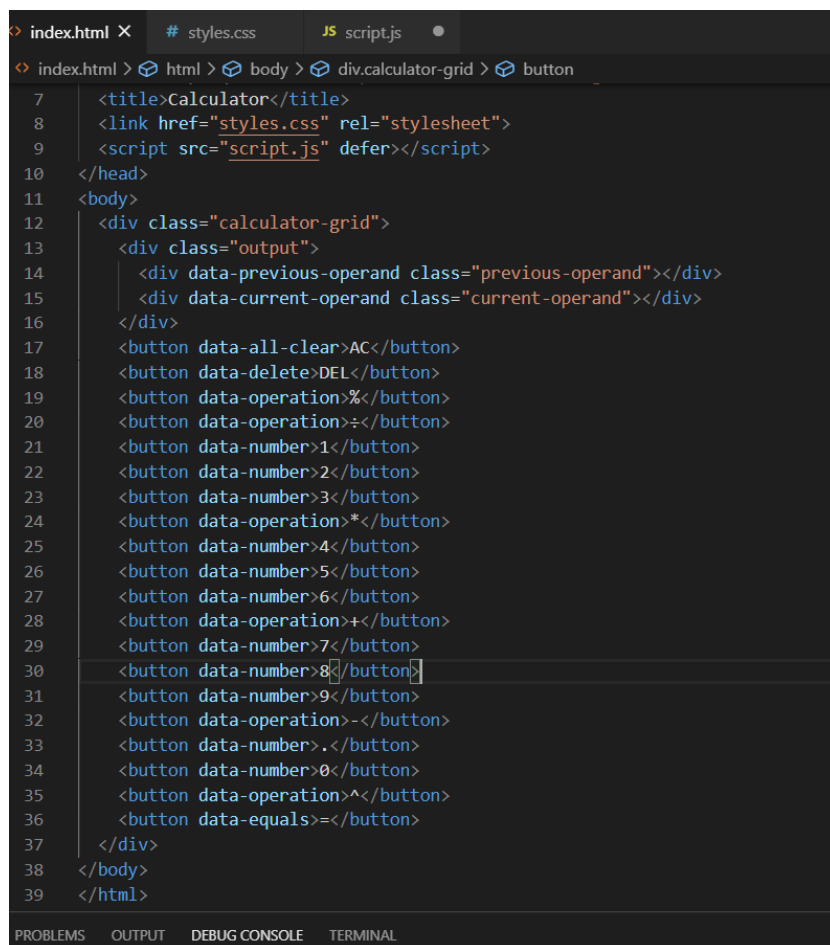


Untuk membuat program diatas, penulis menggunakan HTML, CSS, dan tentunya JavaScript. Penulisan kode dimulai dengan menentukan apa hal yang akan

tampil pada layer. Dengan memperhatikan arahan tugas serta melihat bagaimana tampilan umum kalkulator, penulis memutuskan ada beberapa hal yang perlu ditampilkan. Antaralain :

1. Beberapa tombol
  - Tombol AC(All Clear) → membersihkan isi dari hal yang diketikkan sebelumnya.
  - Tombol Del → berfungsi seperti 'backspace'
  - Tombol untuk angka 0-9 dan titik untuk desimal
  - Tombol operator (+, -, \*, /, %, ^)
  - Tombol "=" untuk mendapatkan hasil.
2. Textfield untuk menampilkan hasil/output serta menampilkan inputan yang akan diolah.

### 3.2 Source HTML



```
index.html X # styles.css JS script.js
index.html > html > body > div.calculator-grid > button
7 <title>Calculator</title>
8 <link href="styles.css" rel="stylesheet">
9 <script src="script.js" defer</script>
10 </head>
11 <body>
12 <div class="calculator-grid">
13 <div class="output">
14 <div data-previous-operand class="previous-operand"></div>
15 <div data-current-operand class="current-operand"></div>
16 </div>
17 <button data-all-clear>AC</button>
18 <button data-delete>DEL</button>
19 <button data-operation>%</button>
20 <button data-operation>÷</button>
21 <button data-number>1</button>
22 <button data-number>2</button>
23 <button data-number>3</button>
24 <button data-operation>*</button>
25 <button data-number>4</button>
26 <button data-number>5</button>
27 <button data-number>6</button>
28 <button data-operation>+</button>
29 <button data-number>7</button>
30 <button data-number>8</button>
31 <button data-number>9</button>
32 <button data-operation>-</button>
33 <button data-number>.</button>
34 <button data-number>0</button>
35 <button data-operation>^</button>
36 <button data-equals>=</button>
37 </div>
38 </body>
39 </html>
```



Diatas ini merupakan isi dari file HTML, yang memiliki link dengan “style.css” dan “script.js” yang dapat dilihat pada line 8 dan 9. Seperti yang disebutkan sebelumnya, kalkulator yang akan dibuat memiliki beberapa tombol serta sebuah textfield yang dapat menampilkan hasil perhitungan serta angka yang di inputkan. Textfield ini dibuat dengan menggunakan tag <div> sehingga memudahkan dalam menentukan letak dari hasil.

Pada line 13, <div> output memiliki dua tag <div> lagi didalamnya yang akan berfungsi sebagai operand yang telah dimasukkan dan satu lagi sebagai operand yang sedang diketikan atau simpelnya, angka pertama dan kedua. Sisanya hanyalah berupa <button> yang dikelompokkan menjadi beberapa kelas seperti operation dan number.

### 3.3 Source CSS

```
body {
  padding: 0;
  margin: 0;
  background: linear-gradient(to top, #a200ff, #0044ff);
}

.calculator-grid {
  display: grid;
  justify-content: center;
  align-content: center;
  min-height: 100vh;
  grid-template-columns: repeat(4, 100px);
  grid-template-rows: minmax(120px, auto) repeat(5, 100px);
}

.calculator-grid > button {
  cursor: pointer;
  font-size: 2rem;
  border: 1px solid white;
  outline: none;
  background-color: rgba(255, 255, 255, .75);
}

.calculator-grid > button:hover {
  background-color: rgba(255, 255, 255, .9);
}
```

Dapat dilihat dalam file css bagaimana style dari website dibuat. Untuk background body, digunakan linear-gradient untuk membuat sebuah gradien warna.

Class calculator-grid digunakan untuk menentukan bagaimana format atau rangka dari kalkulator akan ditampilkan. Dengan align-content: center akan membuat button-button menyatu ditengah layar dan tidak memiliki gap. untuk button, saat cursor ada diatasnya maka akan berubah menjadi pointer, lalu saat button: hover maka warna background dari button akan semakin potih dengan mengubah transparansi menjadi 0.9 atau 90%.

```
34  .output {
35    grid-column: 1 / -1;
36    background-color: rgba(0, 0, 0, .75);
37    display: flex;
38    align-items: flex-end;
39    justify-content: space-around;
40    flex-direction: column;
41    padding: 10px;
42    word-wrap: break-word;
43    word-break: break-all;
44  }
45
46  .output .previous-operand {
47    color: rgba(255, 255, 255, .75);
48    font-size: 1.5rem;
49  }
50
51  .output .current-operand {
52    color: white;
53    font-size: 2.5rem;
54  }
```

Pada class output, display: flex akan memudahkan untuk menentukan agar content didalamnya dapat berada dibagian kanan dengan align-item: flex-end dan untuk posisi saling meunpuk (atas-bawah) dengan flex-direction: column;

Word-wrap: break-word pada line 42 berfungsi agar saat content melebihi dari lebar output maka dapat dipotong dan dilanjutkan dibawahnya. Dan break-all berfungsi agar pemotongan dapat dilakukan dibagian tulisan manapun.

Agar lebih nyaman untuk dilihat, previous-operand pada line 46 dibuat lebih kecil serta lebih gelap dari current-operand line 51. Previous-operand juga dapat dikatakan angka yang telah dimasukkan sebelum sebuah operator di click,

sedangkan current-operand dapat dikatakan sebagai angka yang sedang diketuk atau angka yang akan dilakukan operasi dengan angka pertama.

### 3.4 Source JavaScript

```
77 const numberButtons = document.querySelectorAll('[data-number]')
78 const operationButtons = document.querySelectorAll('[data-operation]')
79 const equalsButton = document.querySelector('[data-equals]')
80 const deleteButton = document.querySelector('[data-delete]')
81 const allClearButton = document.querySelector('[data-all-clear]')
82 const previousOperandTextElement = document.querySelector('[data-previous-operand]')
83 const currentOperandTextElement = document.querySelector('[data-current-operand]')
84
85 const calculator = new Calculator(previousOperandTextElement, currentOperandTextElement)
86
```

Hal yang kita lakukan pertama dalam JavaScript adalah untuk mengambil data dari file HTML dengan membuatnya sebagai sebuah konstanta. Setelah itu buat sebuah class bernama Calculator dengan konstruktor yang mengambil dua buah element yaitu previous operand atau angka pertama dan current operand atau angka kedua.

```
87 numberButtons.forEach(button => {
88   button.addEventListener('click', () => {
89     calculator.appendNumber(button.innerText)
90     calculator.updateDisplay()
91   })
92 })
93
94 operationButtons.forEach(button => {
95   button.addEventListener('click', () => {
96     calculator.chooseOperation(button.innerText)
97     calculator.updateDisplay()
98   })
99 })
100
101 equalsButton.addEventListener('click', button => {
102   calculator.compute()
103   calculator.updateDisplay()
104 })
105
106 allClearButton.addEventListener('click', button => {
107   calculator.clear()
108   calculator.updateDisplay()
109 })
110
111 deleteButton.addEventListener('click', button => {
112   calculator.delete()
113   calculator.updateDisplay()
114 })
```

Selanjutnya kita tentukan event listener untuk tiap button dan apa yang dilakukan Ketika button tersebut di click.

```

65     updateDisplay() {
66         this.currentOperandTextElement.innerText = this.currentOperand
67         if (this.operation !== null) {
68             this.previousOperandTextElement.innerText =
69             ` ${this.previousOperand} ${this.operation}`
70         } else {
71             this.previousOperandTextElement.innerText = ''
72         }
73     }

```

Function `updateDisplay` dipanggil pada setiap akhir event listener. Fintion ini berguna untuk menampilkan hasil terbaru dari current operand, yang berfungsi sebagai angka kedua dan juga sebagai tempat menunjukkan hasil operasi.

Ada sebuah pengkondisian, yaitu jika variable `operation` tidak null atau sudah ada symbol operasi yang dimasukkan maka, operasi tersebut juga akan ditampilkan disamping angka sebelumnya yang telah dimasukkan. Contohnya seperti berikut :



Seperti yang dapat dilihat, jika operasi yang dilakukan adalah pembagian. Sehingga pengguna dapat lebih focus untu menentukan nilai yang akan dimasukkan selanjutnya.

```

8     clear() {
9         this.currentOperand = ''
10        this.previousOperand = ''
11        this.operation = undefined
12    }
13
14    delete() {
15        this.currentOperand = this.currentOperand.toString().slice(0, -1)
16    }
17
18    appendNumber(number) {
19        if (number === '.' && this.currentOperand.includes('.')) return
20        this.currentOperand = this.currentOperand.toString() + number.toString()
21    }
22

```

Selanjutnya ada fungsi `clear` dan `delete` yang sudah jelas kegunaanya, untuk fungsi `delete`, digunakan syntax `slice(0,-1)` yang sederhananya berarti nilai yang diambil dari element ke-0 sampai 1 element sebelum element terakhir.

Lalu ada `appendNumber` yang berfungsi agar pengguna dapat memasukkan angka tanpa harus merubah nilainya setiap saat karena event listener. Kondisi if disini digunakan untuk menghindari dimasukkannya “.” (titik) lebih dari sekali.

Lalu mengubah nilai yang akan menjadi current operand dengan string agar tidak dilakukan operasi tapi membuat suatu 'string yang menyatu' simpelnya agar dapat membuat "1+1 = 11" dan bukan "1+1 =2".

```
23     chooseOperation(operation) {
24         if (this.currentOperand === '') return
25         if (this.previousOperand !== '') {
26             this.compute()
27         }
28         this.operation = operation
29         this.previousOperand = this.currentOperand
30         this.currentOperand = ''
31     }
32
33     compute() {
34         let computation
35         const prev = parseFloat(this.previousOperand)
36         const current = parseFloat(this.currentOperand)
37         if (isNaN(prev) || isNaN(current)) return
38         switch (this.operation) {
39             case '+':
40                 computation = prev + current
41                 break
42             case '-':
43                 computation = prev - current
44                 break
45             case '*':
46                 computation = prev * current
47                 break
48             case '÷':
49                 computation = prev / current
50                 break
51             case '%':
52                 computation = prev % current

```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

Function chooseOperation() disini berfungsi untuk menentukan jika angka pertama telah selesai dan menjadi previous operand, dan memberikan nilai kepada variable operation untuk nantinya digunakan saat melakukan pemilihan komputasi. Sebelum itu ada pengkondisian yang berfungsi untuk mengetahui apakah nilai operand kosong, agar tidak dilakukan komputasi jika tombol operasi diclik saat belum ada angka yang dimasukkan.

Pada pengkondisian selanjutnya adalah pengecekan jika sudah ada angka yang dimasukkan, maka saat tombol operasi lainnya dimasukkan, nilai sebelumnya akan di komputasi tanpa memerlukan tombol “=” yang artinya: dua angka pertama akan dikomputasikan setelah operasi baru masuk dan sebelum angka ketiga dimasukkan.

Lalu yang terakhir adalah function dari komputasi itu sendiri. Tentunya karena ada pembagian, nilai dari kedua angka akan diubah menjadi float dengan `parseFloat()` lalu dilanjutkan dengan `switch...case` yang menggunakan variable operation yang diambil dari button operation yang ditekan.

Berikut ini adalah hasil percobaan setiap operasi.

20 +  
34

54

420 -  
69

351

12 %  
5

2

12 \*  
12

144

12 ÷  
4

3

4 ^  
2

16

## **BAB IV**

### **PENUTUP**

#### **4.1. Simpulan**

JavaScript merupakan Bahasa pemrograman yang berdasarkan pada Bahasa pemrograman java, namun lebih sering dipakai untuk pembuatan website Bersama dengan HTML dan CSS. JavaScript dapat dipakai dalam format file atau ada didalam file HTML sebagai internal Script. Jika berada didalam file HTML, penggunaan JavaScript akan berada didalam tag `<script>`. Menggunakan JavaScript menjadi salah satu pilihan untk membuat suatu website yang dinamis dan interaktif.

#### **4.2. Saran**

JavaScript menjadi suatu pilihan terbaik dalam memulai untuk membuat suatu pemrograman berbasis web yang interaktif. Memberikan sebuah project yang memerlukan penggunaan JavaScript namun tetap memberikan kebebasan mahasiswa untuk berkreasi agar menjadi lebih paham serta lebih banyak mencari tahu tentang penggunaan JavaScript tentu akan menjadi sesuatu yang baik.

## DAFTAR PUSTAKA

*Belajar HTML Dasar: Cara Membuat Form di HTML (tag form) / Duniaikom* (no date). Available at: <https://www.duniaikom.com/belajar-html-cara-membuat-form-di-html-tag-form/> (Accessed: 20 October 2020).

*Apa Itu JavaScript? Pemahaman Dasar Mengenai JavaScript bagi Para Pemula* (no date). Available at: <https://www.hostinger.co.id/tutorial/apa-itu-javascript/> (Accessed: 16 November 2020).

*Pemrograman Javascript: Langkah Awal Belajar Javascript* (no date). Available at: <https://www.petanikode.com/javascript-dasar/> (Accessed: 16 November 2020).

*Tutorial Belajar JavaScript: Cara Menjalankan Kode Program JavaScript / Duniaikom* (no date). Available at: <https://www.duniaikom.com/tutorial-belajar-javascript-cara-menjalankan-kode-program-javascript/> (Accessed: 16 November 2020).