

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS WEB
“TUGAS PERTEMUAN 6”**



Disusun Oleh :

Nama : I Gede Aditya Mahardika Pratama

Nim : 1808561028

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 6 yang berjudul "Tugas Pertemuan 6" 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.

Badung, 13 November 2020

Penyusun

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Manfaat	1
BAB II KAJIAN PUSTAKA	2
2.1 JavaScript.....	2
BAB III PEMBAHASAN	3
3.1 Penjelasan Program	3
3.2 Output Program	8
BAB IV PENUTUP	10
4.1 Simpulan.....	10
4.2 Saran	10
DAFTAR PUSTAKA	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Semakin hari teknologi semakin berkembang. Salah satunya teknologi informasi, banyak sekali bidang yang memanfaatkan teknologi Informasi. Tuntutan kebutuhan akan informasi dan penggunaan komputer yang semakin banyak mendorong terbentuknya sebuah jaringan komputer yang mampu melayani berbagai kebutuhan tertentu. Jaringan komputer kita kenal dengan nama internet. Internet bisa diakses dan dimanfaatkan untuk berbagai keperluan, oleh siapa saja dan dimana saja, kapanpun kita mau menggunakannya. Salah satunya adalah web, yang mampu menyediakan informasi dalam bentuk teks, gambar, suara maupun gambar bergerak.

Dengan terus berkembangnya roda perkembangan teknologi, web berkembang menjadi alat bantu yang tidak hanya mampu menyediakan informasi, namun juga mampu untuk mengolah informasi. Proses pengolahan informasi dengan memanfaatkan teknologi web menyebabkan web menjadi media yang dinamis. Pada kesempatan ini penulis akan membahas tentang bahasa pemrograman JavaScript sebagai salah satu perangkat lunak yang turut menentukan dapat digunakannya media web sebagai media pengolah informasi. Bahasa tersebut dapat digunakan agar aplikasi web yang dibuat lebih responsif.

1.2 Tujuan

1. Untuk mengetahui tentang JavaScript serta penggunaannya.

1.3 Manfaat

Adapun manfaat dari pembuatan laporan ini untuk membantu mahasiswa dalam memahami tentang JavaScript dan cara implementasinya dalam pembuatan aplikasi berbasis web.

BAB II

KAJIAN PUSTAKA

2.1 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 (Ariata, 2019).

Terdapat sejumlah kelebihan JavaScript yang menjadikan bahasa pemrograman ini lebih unggul jika dibandingkan dengan kompetitornya, terutama di beberapa kasus tertentu seperti lebih mudah dipelajari, error atau kesalahan lebih mudah dicari dan ditangani, dapat dialih tugaskan ke elemen halaman web atau even tertentu, dan lain-lain. Lalu kekurangan dari JavaScript seperti mengundang para hacker, scammer, dan pihak ketiga berbahaya lainnya untuk mencari celah keamanan.

Biasanya JavaScript di-embed secara langsung ke halaman website atau diarahkan melalui file .js yang terpisah. JavaScript merupakan bahasa dari sisi klien yang berarti script diunduh di perangkat yang dimiliki oleh pengunjung situs Anda, lalu diproses di sana. Berbeda dengan halnya bahasa di sisi server yang dijalankan pada server sebelum bahasa pemrograman tersebut mengirimkan file ke pengunjung situs.

Anggap saja kita sedang membangun sebuah rumah. HTML adalah struktur bangunan rumah dan sandaran bagi pintu dan tembok, sedangkan CSS adalah karpet dan wallpaper yang menghiasi setiap sudut rumah sehingga tampak lebih indah dan menarik. JavaScript, dalam hal ini, menambahkan sifat responsif sehingga Anda bisa membuka pintu dan menyalakan lampu. Anda bisa saja tidak menggunakan lampu atau tidak membutuhkan pintu. Namun, risikonya adalah rumah yang gelap dan usang, layaknya bangunan yang telah berdiri sejak tahun 1995. Hal ini sama seperti ketika Anda memanfaatkan JavaScript untuk webiste.

BAB III

PEMBAHASAN

3.1 Penjelasan Program

Pada program ini saya menggunakan konsep struktur data dari stack atau tumpukan yaitu Notasi Polish. Dimana user akan menginputkan rumus secara infix dan pada program nanti akan diproses menggunakan postfix.

```
151 // INPUT RUMUS
152 var rumus = prompt("Input Rumus\nMisal : ((-2)*5)/((2-5)^3)\nHanya menerima operator : +,-,*,/,%,^2");
153 // HILANGKAN SPASI
154 rumus = rumus.split(' ').join('');
155 // DEKLARASI
156 var stack = [];
157 var postfix = [];
158 var error = false;
159 // PROSES
160 toPostfix(rumus);
161 popping_stack()
162 // OUTPUT
163 // alert('stack : ' + stack);
164 if (!error) {
165 |   alert('Hasil :\n' + rumus + ' = ' + postfix);
166 }
```

Pertama kita bahas dari main program dari file javascript ini. Pertama adalah input rumus dari user dengan menggunakan syntax prompt. Prompt merupakan kolom pop up yang dapat digunakan untuk menginputkan data dari user yang nantinya disimpan dalam bentuk string pada variable rumus.

Pada line 154 digunakan untuk menghilangkan spasi yang ada pada inputan user. Dengan cara memisahkan setiap char berdasarkan spasi dengan menggunakan *split()*, lalu digabungkan lagi tanpa ada spasi menggunakan *join()*.

Pada line 156 sampai 158 merupakan deklarasi dari array *stack* untuk menyimpan tumpukan, array *postfix* untuk menyimpan hasil proses dan variable *error* yang digunakan untuk mengecek apakah terjadi error saat proses berlangsung.

Pada line 160 dan 161 digunakan untuk memanggil fungsi proses. Dan line 164 sampai 166 digunakan untuk menampilkan hasil proses ketika tidak terjadi error.

```

32 // FUNCTION UNTUK VALIDASI INPUTAN
33 function validation(element) {
34     if (!((element.charCodeAt()>=47 && element.charCodeAt()<=57)||element.charCodeAt()==37||((element.charCodeAt()>=40 && element.charC
35         return true;
36     }else return false;
37 }

```

Function *validation()* digunakan untuk mengecek suatu char apakah char tersebut berupa angka, tanda kurung, tambah, kurang, bagi, kali, mod, dan pangkat. Ketika benar akan mengembalikan nilai *true*, jika tidak mengembalikan nilai *false*.

```

1 // MELAKUKAN OPERASI ARITMATIKA
2 function hitung() {
3     var hasil = 0;
4     if(postfix[postfix.length-1]=='+') hasil = parseFloat(postfix[postfix.length-3]) + parseFloat(postfix[postfix.length-2]);
5     else if(postfix[postfix.length-1]=='-') hasil = parseFloat(postfix[postfix.length-3]) - parseFloat(postfix[postfix.length-2]);
6     else if(postfix[postfix.length-1]=='*') hasil = parseFloat(postfix[postfix.length-3]) * parseFloat(postfix[postfix.length-2]);
7     else if(postfix[postfix.length-1]=='^') hasil = parseFloat(postfix[postfix.length-3]) ** parseFloat(postfix[postfix.length-2]);
8     else if(postfix[postfix.length-1]=='%') hasil = parseFloat(postfix[postfix.length-3]) % parseFloat(postfix[postfix.length-2]);
9     else{
10         // ERROR PEMBAGIAN DENGAN 0
11         if(parseFloat(postfix[postfix.length-2])==0){
12             alert('ERROR : TIDAK BISA MEMBAGI DENGAN 0');
13             error = true;
14         }
15         else hasil = parseFloat(postfix[postfix.length-3]) / parseFloat(postfix[postfix.length-2]);
16     }
17     postfix[postfix.length-3] = hasil.toFixed(4);
18     postfix = postfix.slice(0, postfix.length-2);
19 }

```

Function *hitung()* digunakan untuk melakukan operasi aritmatika. Dimana pada proses pembagian program akan memberikan nilai error jika penyebut dari pembagian adalah 0.

```

20 // LEVEL OPERATOR
21 function level(operator) {
22     switch (operator) {
23         case '(': return 0; break;
24         case '+':
25         case '-': return 1; break;
26         case '*':
27         case '/':
28         case '%': return 2; break;
29         case '^': return 3; break;
30     }
31 }

```

Function *level()* digunakan untuk menentukan level atau tingkatan dari operator aritmatika. Dimana tanda kurung buka dengan level 0, tanda tambah dan kurang dengan level 1, tanda bagi, kali dan mod dengan level 2, yang terakhir tanda pangkat dengan level 3.

```

38 // MENGEKLUARKAN DARI STACK ATAU TUMPUKAN
39 function popping_stack() {
40     for (let i = stack.length-1; i >= 0; i--) {
41         // ERROR TIDAK TERDAPAT KURUNG TUTUP
42         if(stack[i]==('(')){
43             alert('ERROR : TIDAK TERDAPAT KURUNG TUTUP');
44             error = true;
45             break;
46         }else{
47             postfix.push(stack[i]);
48             hitung();
49             stack.pop();
50         }
51     }
52 }

```

Function *popping_stack()* digunakan untuk mengeluarkan operator-operator yang masih ada pada stack. Saat melakukan pengeluaran data atau pop akan dilakukan pemasukan atau push pada array postfix. Dimana pada saat menemui tanda kurung buka, maka program akan memberikan nilai error dengan menggunakan pop up berupa *alert*.

```

53 // CONVERT TO POSTFIX
54 function toPostfix(rumus){
55     var array = rumus.split('');
56     for (let i = 0; i < array.length; i++) {
57         // ERROR INPUTAN TIDAK SESUAI
58         if (validation(array[i])) {
59             alert('ERROR : HANYA BISA MENERIMA ANGKA DAN OPERATOR YANG DITENTUKAN');
60             error = true;
61             break;

```

Function *toPostfix()* digunakan untuk mengubah inputan user berupa infix ke bentuk postfix. Diawali dengan mengubah inputan user dari string menjadi array. Dilanjutkan dengan pengecekan setiap char. Statement yang pertama adalah cek validasi inputan dengan menggunakan function *validation()*. Ketika pemanggilan function tersebut bernilai *true* maka program akan memberikan nilai error. Jika tidak akan dilanjutkan ke proses selanjutnya.

```

    }else{
        // INPUTAN KURUNG BUKA
        if(array[i]==('('){
            if (i!=0)
                // ERROR JIKA TERDAPAT ANGKA ATAU KURUNG TUTUP SEBELUM KURUNG BUKA
                if ((array[i-1].charCodeAt()>=48 && array[i-1].charCodeAt()<=57)||array[i-1].charCodeAt()==41) {
                    alert('ERROR : SEBELUM KURUNG BUKA HARUS TERDAPAT OPERATOR');
                    error = true;
                    break;
                }
            stack.push(array[i]);

```

Statement selanjutnya digunakan untuk mengecek char kurung buka. Ketika char sebelum kurung buka adalah angka atau kurung tutup maka program akan

memberikan nilai error. Jika tidak maka char kurung buka akan di push ke array stack.

```
// INPUTAN KURUNG TUTUP
}else if (array[i]==')') {
    var kurungBuka = false;
    for (let j = stack.length-1; j >= 0; j--) {
        if (stack[j]=='('){
            kurungBuka = true;
            stack.pop();
            break;
        }
        postfix.push(stack[j]);
        hitung();
        stack.pop();
    }
    // ERROR TIDAK TERDAPAT KURUNG TUTUP
    if (!kurungBuka) {
        alert("ERROR : TIDAK TERDAPAT KURUNG TUTUP");
        error = true;
        break;
    }
}
```

Statement selanjutnya untuk mengecek inputan kurung tutup. Pada statement ini akan melakukan *poping* atau mengeluarkan data dari array stack sampai menemukan tanda kurung buka pada array tersebut. Jika pada stack tersebut tidak terdapat kurung buka maka program akan memberikan nilai error.

```
// INPUTAN ANGKA
}else if (array[i].charCodeAt()>=48 && array[i].charCodeAt()<=57) {
    if (i==0) {
        postfix.push(array[i]);
    }else{
        if (array[i-1].charCodeAt()>=48 && array[i-1].charCodeAt()<=57) postfix[postfix.length-1] = postfix[postfix.length-1] + array[i];
        else if (array[i-1].charCodeAt()==45 && array[i-2].charCodeAt()==40){
            stack.pop();
            postfix.push('-'+array[i]);
        }
        // ERROR JIKA SETELAH KURUNG TUTUP TERDAPAT ANGKA
        else if (array[i-1].charCodeAt()==41) {
            alert('ERROR : SETELAH KURUNG TERDAPAT ANGKA');
            error = true;
            break;
        }else postfix.push(array[i]);
    }
}
```

Statetment selanjutnya digunakan untuk mengecek char angka. Dimana jika sebelum char ini adalah char angka maka kedua char tersebut akan digabungkan dan char gabungan tersebut akan dipush ke array *postfix*. Jika tidak maka char saat ini akan dipush ke array *postfix*. Ketika char sebelumnya adalah tanda minus dan sebelumnya lagi adalah tanda kurung buka, maka char saat ini akan digabung

dengan tanda minus/kurang. Namun jika sebelum char ini adalah tanda kurung tutup maka program akan memberikan nilai error.

```
//INPUTAN OPERATOR
}else {
    // ERROR JIKA RUMUS DIAWALI OPERATOR
    if (i==0) {
        alert('ERROR : RUMUS HARUS DIAWALI OPERAND ATAU KURUNG BUKA');
        error = true;
        break;
    }
}
```

Statement terakhir digunakan untuk mengecek pada inputan operator. Ketika operator terdapat pada awal rumus, maka program akan memberikan nilai error. Jika tidak akan dilanjutkan pada penjelasan dibawah.

```
}else{
    // ERROR JIKA TERDAPAT OPERATOR YANG BERURUTAN
    if(array[i-1].charCodeAt()===37||array[i-1].charCodeAt()===42||array[i-1].charCodeAt()===43||array[i-1].charCodeAt()===44){
        alert('ERROR : TERDAPAT OPERATOR YANG BERURUTAN');
        error = true;
        break;
    }
}
```

Masih pada statement inputan operator, didalamnya terdapat statement ketika sebelum char operator adalah operator maka program akan memberikan nilai error. Jika tidak akan dilanjutkan pada penjelasan dibawah.

```
}else{
    if(stack.length==0) {
        stack.push(array[i]);
    }else{
        if(level(array[i])>level(stack[stack.length-1])){
            stack.push(array[i]);
        }else{
            for (let j = stack.length-1; j >= 0; j--) {
                if(stack[j]=='(') break;
                else{
                    if(level(array[i])<=level(stack[j])){
                        postfix.push(stack[j]);
                        hitung();
                        stack.pop();
                    }else break;
                }
            }
            stack.push(array[i]);
        }
    }
}
```

Masih dalam statement inputan operator, dilanjutkan pengecekan ketika level dari char operator saat ini lebih besar dari level operator terakhir pada array stack maka char operator saat ini akan dipush kedalam array stack.

Jika tidak maka akan dilakukan popping operator pada array stack dengan syarat level dari operator saat ini lebih kecil atau samadengan level operator terakhir yang ada pada array stack. Hal tersebut akan terus berlangsung ketika operator terakhir pada array stack bukan tanda kurung buka.

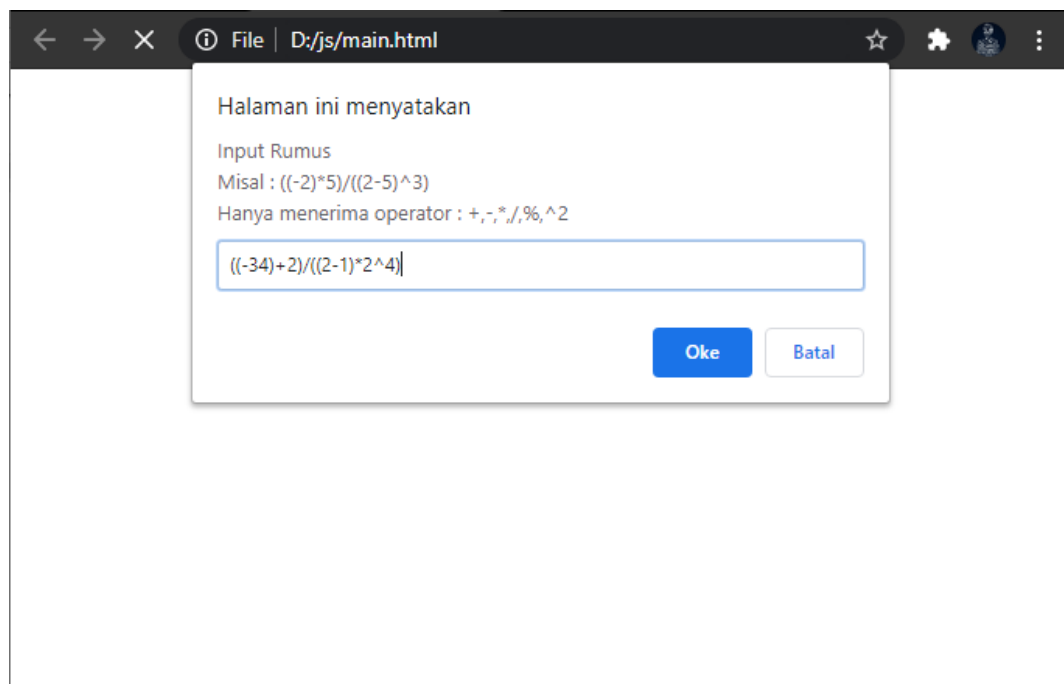
Ketika perulangan tersebut berhenti maka operator saat ini akan dipush kedalam array stack.

```
1 <!DOCTYPE html>
2 <head>
3   <title>CALCULATOR</title>
4 </head>
5 <body>
6   <a onClick="window.location.reload()"><h1>Click me to refresh</h1></a>
7   <script src="javascript.js"></script>
8 </body>
9 </html>
```

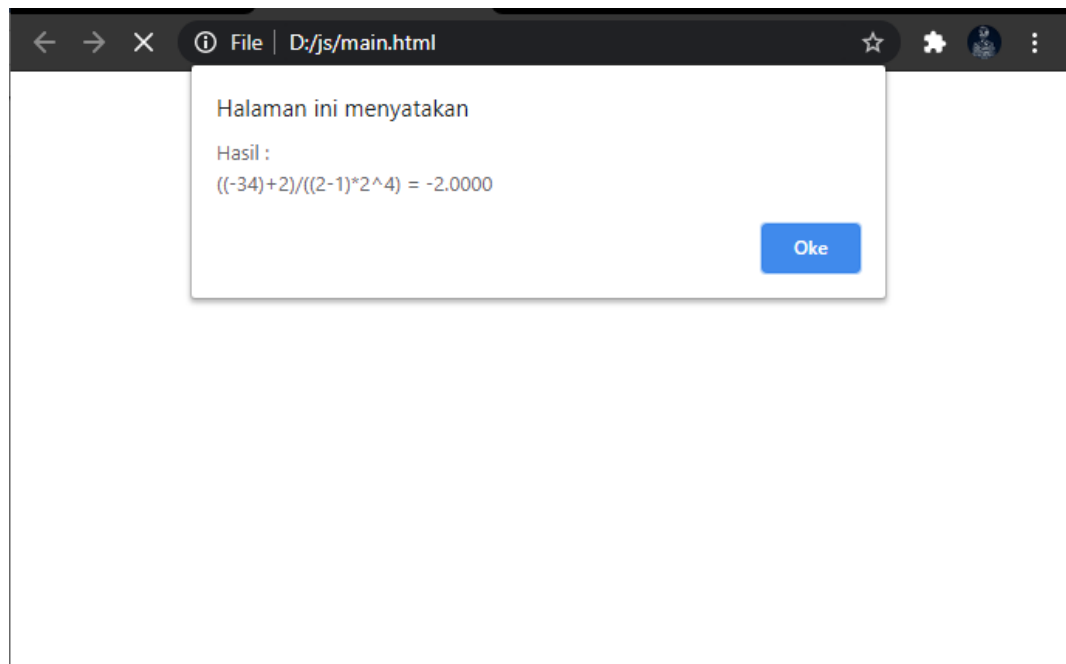
Gambar diatas merupakan file html yang digunakan untuk menjalankan file javascript yang telah dibuat sebelumnya.

3.2 Output Program

Input rumus :



Hasil operasi dari rumus yang diinputkan :



BAB IV

PENUTUP

4.1 Simpulan

JavaScript dibuat dan didesain selama sepuluh hari oleh Brandan Eich, seorang karyawan Netscape, pada bulan September 1995. Terdapat sejumlah kelebihan JavaScript seperti lebih mudah dipelajari, error atau kesalahan lebih mudah dicari dan ditangani. Lalu kekurangan dari JavaScript seperti mengundang para hacker, scammer, dan pihak ketiga berbahaya lainnya untuk mencari celah keamanan. Biasanya JavaScript di-embed secara langsung ke halaman website atau diarahkan melalui file .js yang terpisah. Anggap saja kita sedang membangun sebuah rumah. HTML adalah struktur bangunan rumah dan sandaran bagi pintu dan tembok, sedangkan CSS adalah karpet dan wallpaper yang menghiasi setiap sudut rumah sehingga tampak lebih indah dan menarik. JavaScript, dalam hal ini, menambahkan sifat responsif sehingga Anda bisa membuka pintu dan menyalakan lampu. Anda bisa saja tidak menggunakan lampu atau tidak membutuhkan pintu.

4.2 Saran

Saran yang dapat saya berikan pada laporan ini, para mahasiswa agar dapat lebih memahami mengenai JavaScript. Dan juga para mahasiswa lebih memaksimalkan penggunaan JavaScript dalam membantu pembuatan project aplikasi berbasis web.

DAFTAR PUSTAKA

Ariata, C., 2019. *Apa Itu JavaScript? Pemahaman Dasar Mengenai JavaScript bagi Para Pemula*.
[Online]
Available at: <https://www.hostinger.co.id/tutorial/apa-itu-javascript/>
[Accessed 5 November 2020].