

TUGAS PENDAHULUAN

Praktikum Rekayasa Perangkat Lunak Berbasis Komponen

Modul 1

Pengenalan Javascript dan React

Nama : Firsta Adi Pradana

NIM : 21120118130059

Kelompok : 40

1. *Scope* dalam javascript adalah cakupan yang menentukan bagaimana suatu variabel atau fungsi dapat diakses. Terdapat 2 jenis *scope* pada javascript yaitu *root scope* dan *child scope*. Status ini juga dapat dipengaruhi oleh penggunaan `let`, `const`, dan `var`.

2.

```
// arrow function
var greeting_arrow = (name) => {
  return `Hello, ${name}`
}
// normal function
var greeting_normal = function(name){
  return `Hello, ${name}`
}
```

3. `OnClick`: adalah method yang akan menjalankan suatu fungsi ketika terjadi event klik pada element tersebut. Contoh: Counter akan bertambah 1 setiap tombol ditekan oleh pengguna.
4. Proses `synchronous` bersifat `blocking`, artinya suatu baris kode harus selesai dahulu sebelum baris kode sesudahnya dapat dijalankan. Sedangkan, pada dasarnya `asynchronous` bersifat `non-blocking` sehingga baris kode berikutnya dapat dijalankan tanpa menunggu baris kode sebelumnya selesai dijalankan (kecuali menggunakan `await`).

Firsta Adi

Pradana

21120118130059



TUGAS PENDAHULUAN

Praktikum Rekayasa Perangkat Lunak Berbasis Komponen

Modul 1

Pengenalan Javascript dan React

Nama : Damarjati Rahmandaru

NIM : 21120118130074

Kelompok : 40

1. Scope pada javascript adalah sebuah konsep yang digunakan untuk membatasi akses pada suatu variable. Terdapat 2 tipe scope yaitu lokal dan global.

2.

```
// normal function
function namaFungsi() {
    console.log("Selamat Datang!");
}

// arrow function
var namaFungsi = () => {
    console.log("Selamat Datang!");
}
```

3. Onload adalah event ketika jika saat element atau halaman dibuka. Dapat juga diartikan sebagai sebuah fungsi dimana ketika halaman pada browser kita refresh akan menampilkan sesuatu. Contoh: kita menaruh event onload di tag, artinya jika halaman web sudah diloat semua, maka kode javascript akan dieksekusi.

4. **Synchronous:** Menunggu untuk setiap operasi untuk menyelesaikan, setelah itu hanya mengeksekusi operasi berikutnya.

Asynchronous: tidak pernah menunggu untuk setiap operasi untuk menyelesaikan, melainkan menjalankan semua operasi yang pertama PERGI saja.



Damarjati Rahmandaru

21120118130074