

LAPORAN PRAKTIKUM  
PEMROGRAMAN BERBASIS FRAMEWORK

Pertemuan Minggu-2

Modul 1 – Modern JavaScript

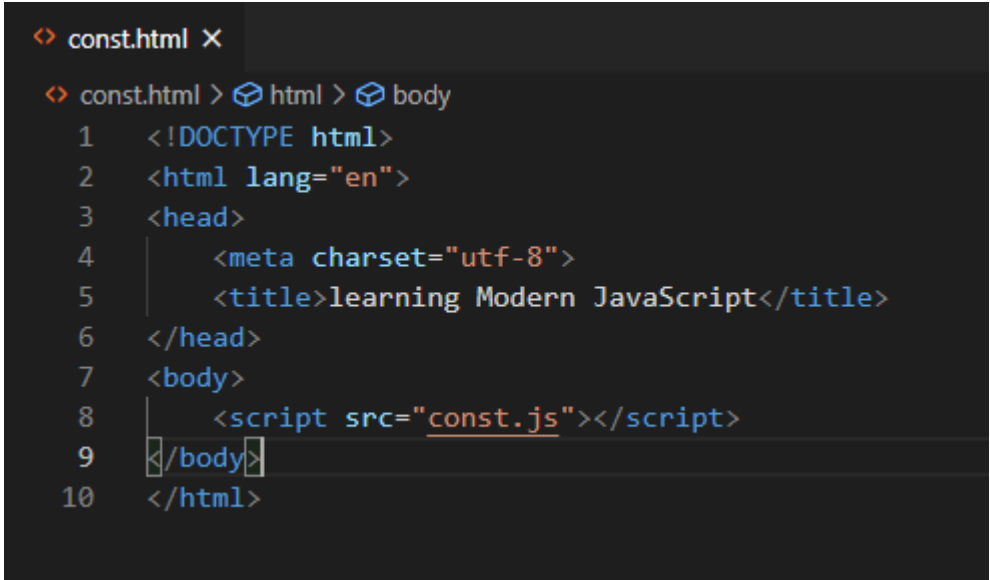
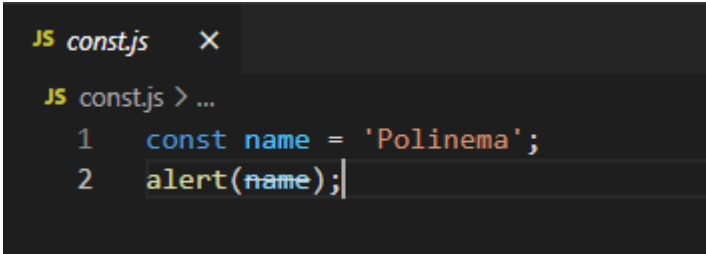
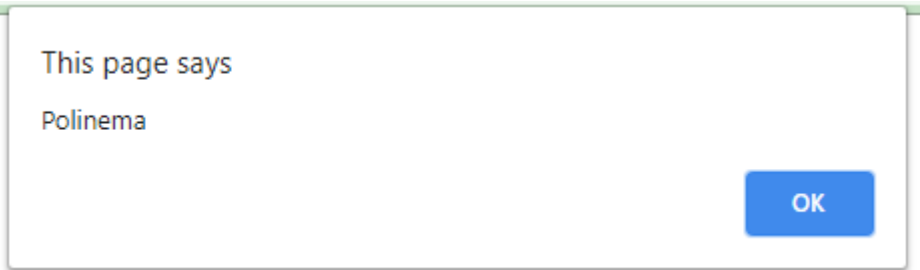


NAMA : DINA RISKY ALIN SAPUTRI  
NIM : 1841720016  
KELAS : 3F

PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG

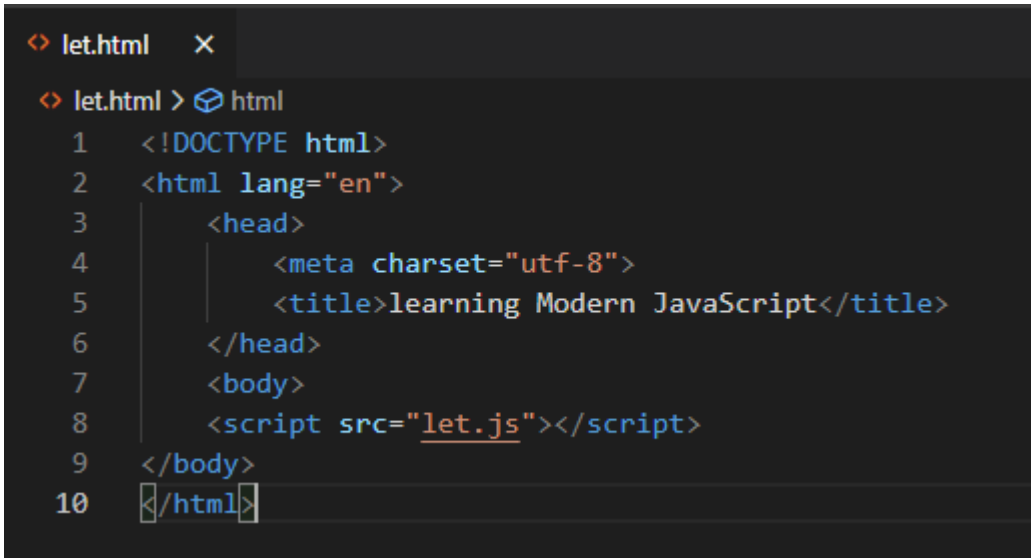
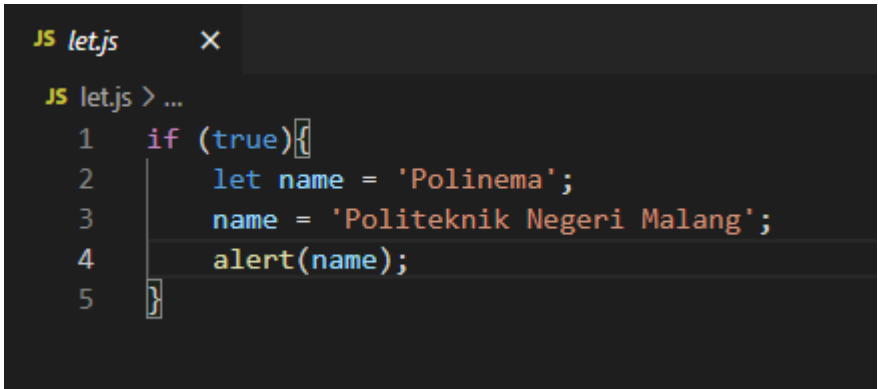
2021

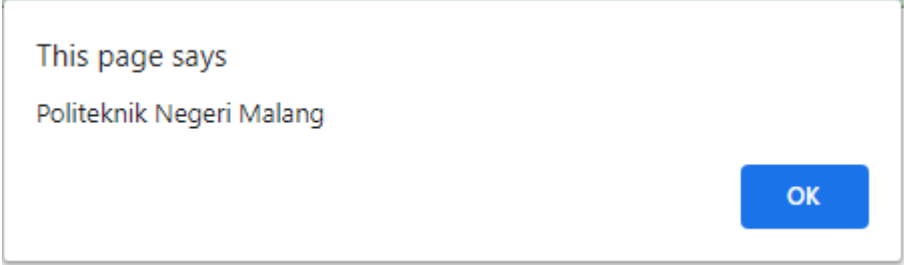
## A. Membuat variabel menggunakan const

Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.
	const.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js
	 <pre>&lt;&gt; const.html X &lt;&gt; const.html &gt; html &gt; body 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3  &lt;head&gt; 4      &lt;meta charset="utf-8"&gt; 5      &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6  &lt;/head&gt; 7  &lt;body&gt; 8      &lt;script src="const.js"&gt;&lt;/script&gt; 9  &lt;/body&gt; 10 &lt;/html&gt;</pre>
	const.js -> code pada file ini terdapat konstanta nama dengan bersifat tetap dan dialog alert yang digunakan untuk menampilkan sebuah pesan peringatan atau informasi dengan parameter yang isinya 'Polinema'.
	 <pre>JS const.js X JS const.js &gt; ... 1  const name = 'Polinema'; 2  alert(name);</pre>
2	Selanjutnya buka file const.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.
	 <p>This page says Polinema OK</p>

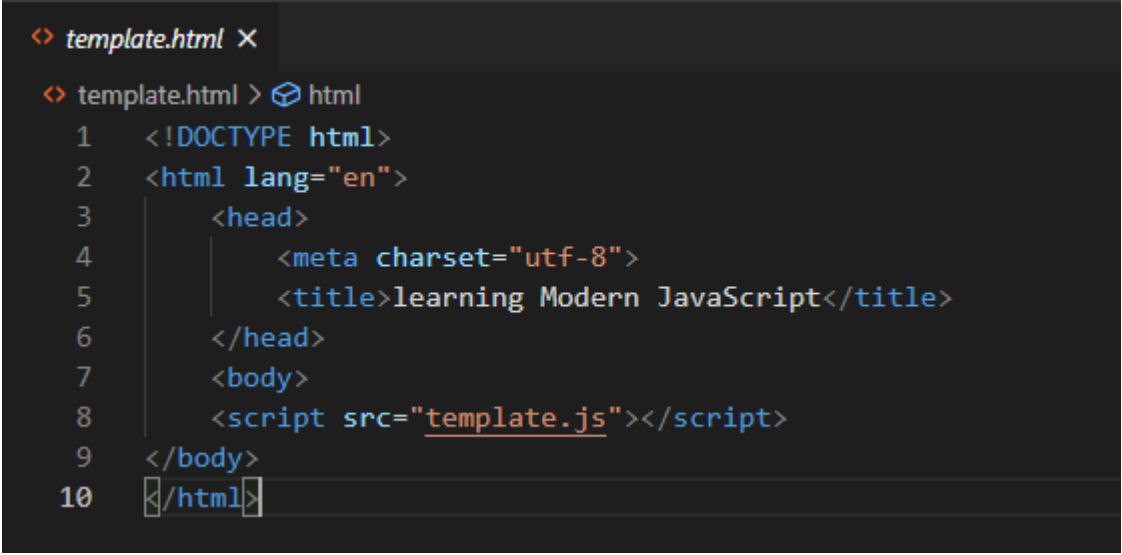
	Setelah membuka file const.html pada browser maka hasilnya akan menampilkan sebuah dialog alert yang isinya 'Polinema'.
--	---

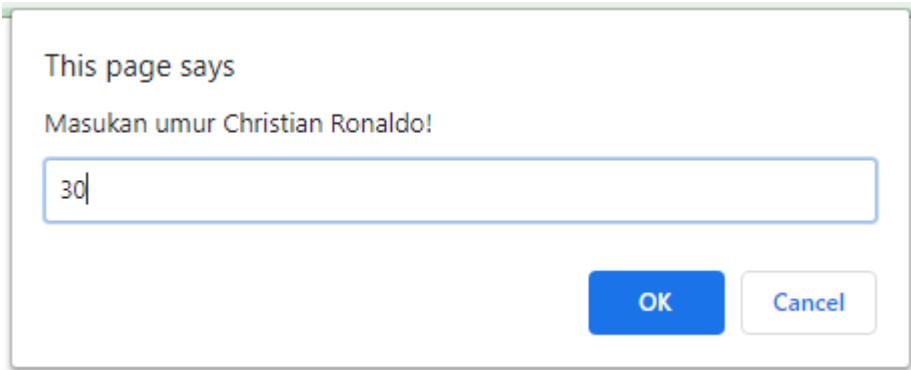
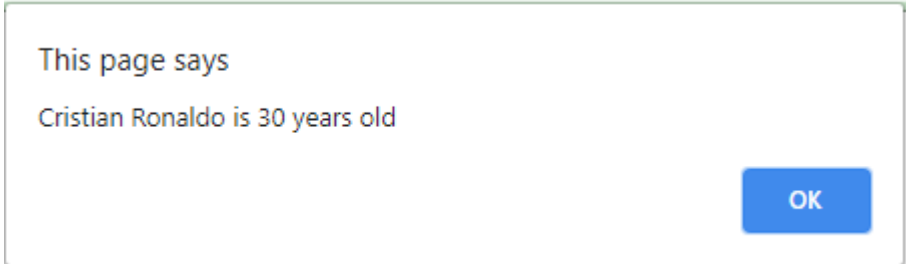
## B. Membuat variabel menggunakan let

Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	let.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js
	 <pre> let.html  X let.html &gt; html 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3    &lt;head&gt; 4      &lt;meta charset="utf-8"&gt; 5      &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6    &lt;/head&gt; 7    &lt;body&gt; 8      &lt;script src="let.js"&gt;&lt;/script&gt; 9    &lt;/body&gt; 10  &lt;/html&gt; </pre>
	let.js -> code pada file ini terdapat sebuah kondisi dimana kondisi tersebut ada let nama, disini let nama sama seperti var yaitu mendeklarasikan sebuah isi dari variabel. Disini let menggantikan isi dari variabel yang isinya 'Polinema' menjadi 'Politeknik Negeri Malang' dengan menampilkannya menggunakan dialog alert.
	 <pre> JS let.js  X JS let.js &gt; ... 1  if (true){ 2    let name = 'Polinema'; 3    name = 'Politeknik Negeri Malang'; 4    alert(name); 5  } </pre>
2	Selanjutnya buka file let.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.

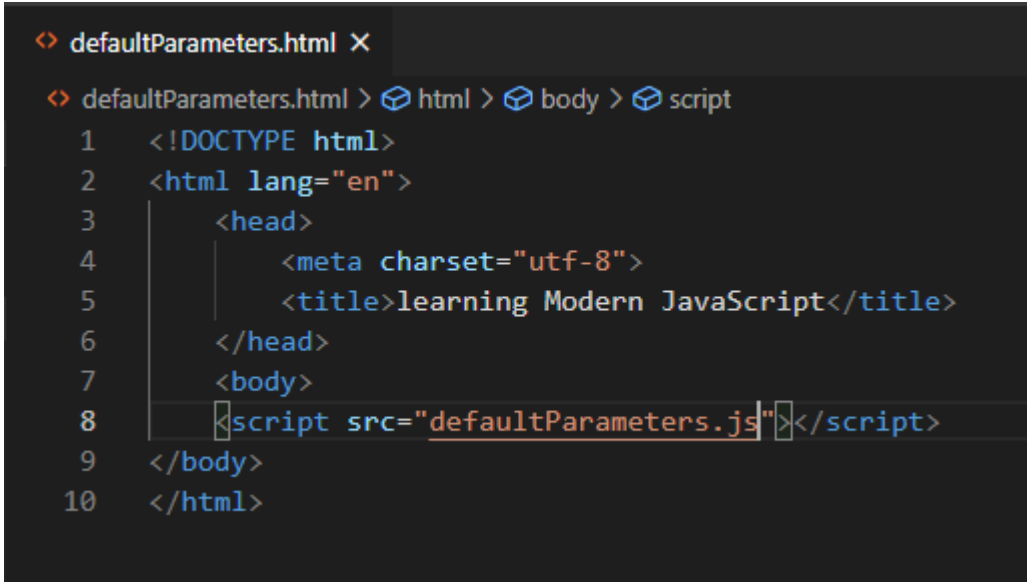
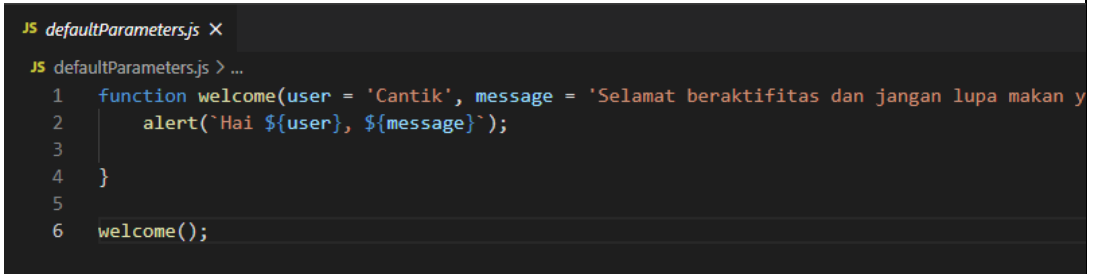
	 <p>Setelah membuka file let.html pada browser maka hasilnya akan menampilkan sebuah dialog alert yang isinya ‘Politeknik Negeri Malang’.</p>
--	---

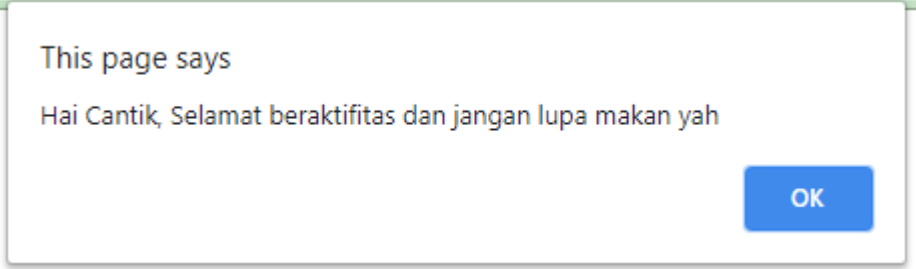
### C. Membuat Template Strings

Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	template.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js
	
	template.js -> code pada file ini terdapat beberapa let dengan mendeklarasikan masing2 variabel, disini let nama sama seperti var yaitu mendeklarasikan sebuah isi dari variabel. Terdapat dialog prompt() untuk mengambil sebuah inputan dari pengguna. Disini dialog prompt() akan mengembalikan nilai string dari apa yang diinputkan oleh pengguna. kemudian let mendeklarasikan masing2 variabel dengan memanggil masing2 deklarasi variabel dan menampilkannya menggunakan dialog alert.

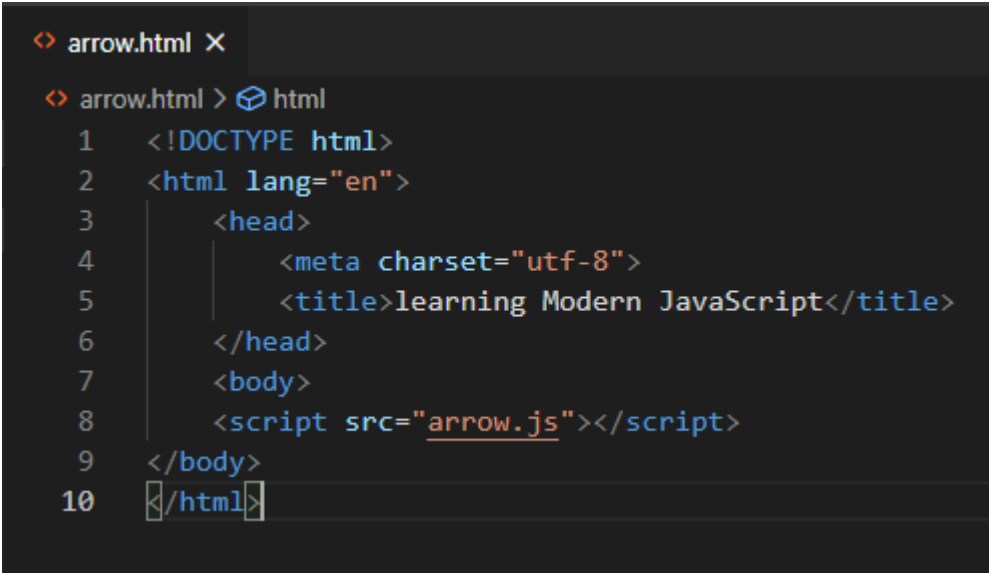
	<pre> JS template.js X JS template.js &gt; ... 1  let fname = "Cristian"; 2  let lname = "Ronaldo"; 3  let age = prompt("Masukan umur Christian Ronaldo!"); 4 5  // Cara lama 6  // let result = fname + ' ' + lname + 'is' + age + 'years old'; 7  // alert(result); 8 9  // Memakai template strings 10 let result = `\${fname} \${lname} is \${age} years old`; 11 alert(result); </pre>
2	<p>Selanjutnya buka file template.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.</p>
	 <p>Setelah membuka file template.html pada browser maka hasilnya akan menampilkan sebuah dialog alert yang isinya 'Masukan umur Christian Ronaldo!'. Ketika melakukan inputan angka '30' maka setelah itu kita lanjutkan dengan klik OK.</p>
	<p>Klik Ok, lalu hasilnya seperti dibawah ini.</p>
	 <p>Setelah Klik OK, maka hasil setelah melakukan inputan angka '30' hasilnya akan muncul sebuah pesan 'Cristian Ronaldo is 30 years old'.</p>


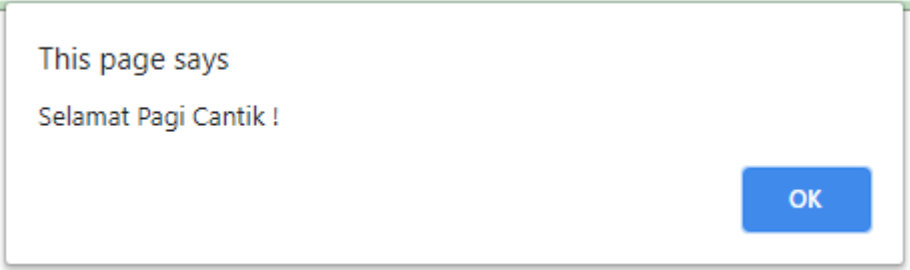
#### D. Membuat default parameters

Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	defaultParameters.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js
	 <pre> &lt;&gt; defaultParameters.html X &lt;&gt; defaultParameters.html &gt; html &gt; body &gt; script 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3    &lt;head&gt; 4      &lt;meta charset="utf-8"&gt; 5      &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6    &lt;/head&gt; 7    &lt;body&gt; 8      &lt;script src="defaultParameters.js"&gt;&lt;/script&gt; 9    &lt;/body&gt; 10  &lt;/html&gt; </pre>
	defaultParameters.js -> code pada file ini terdapat function welcome dengan memiliki 2 parameter bertipe string dimana isi nya untuk parameter pertama dengan nama 'user' dan untuk parameter kedua 'message'. Pada masing2 parameter terdapat deklarasi isi dari masing2 parameter tersebut kemudian menampilkan parameter2 tersebut menggunakan dialog alert dengan memanggil function yang sudah di tentukan.
	 <pre> JS defaultParameters.js X JS defaultParameters.js &gt; ... 1  function welcome(user = 'Cantik', message = 'Selamat beraktifitas dan jangan lupa makan y 2    alert(`Hai \${user}, \${message}`); 3 4  } 5 6  welcome(); </pre>
2	Selanjutnya buka file defaultParameters.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.

	 <p>Setelah membuka file defaultParameters.html pada browser maka hasilnya akan menampilkan sebuah dialog alert dari masing2 paramater yang dideklarasikan yang isinya 'Hai Cantik, Selamat beraktifitas dan jangan lupa makan yah'.</p>
--	--

### E. Membuat Arrow Function 1

Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	arrow.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js
	 <pre> &lt;&gt; arrow.html X &lt;&gt; arrow.html &gt; html 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3    &lt;head&gt; 4      &lt;meta charset="utf-8"&gt; 5      &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6    &lt;/head&gt; 7    &lt;body&gt; 8      &lt;script src="arrow.js"&gt;&lt;/script&gt; 9    &lt;/body&gt; 10  &lt;/html&gt; </pre>
	arrow.js -> code pada file ini terdapat beberapa let dengan mendeklarasikan 1 variabel dengan berparameter message. Kemudian terdapat dialog alert dimana dialog alert tersebut digunakan untuk menampilkan pesan peringatan atau informasi dengan memanggil parameter message dengan tambahan dari parameternya 'Cantik !' .

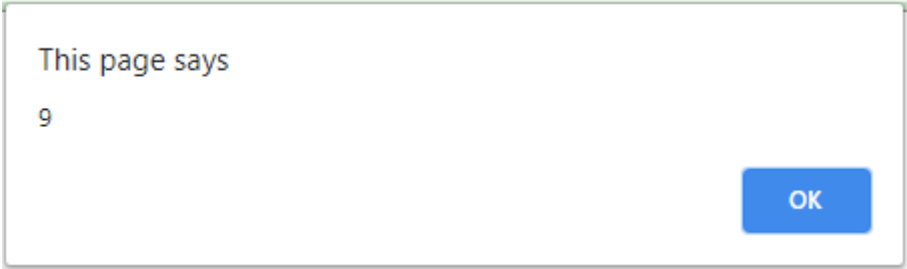
	 <pre> JS arrow.js  X JS arrow.js &gt; ... 1  // Tanpa arrow 2  // function greeting(message){ 3  // return alert(`\${message} Cantik !`) 4  //} 5 6  // Menggunakan arrow 7  let greeting = message =&gt; alert(`\${message} Cantik !`); 8 9  greeting('Selamat Pagi'); </pre>
2	Selanjutnya buka file arrow.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.
	 <p>Setelah membuka file arrow.html pada browser maka hasilnya akan menampilkan sebuah dialog alert dari paramater yang dideklarasikan yang isinya 'Cantik'. Hasilnya bisa seperti "Selamat Pagi Cantik !" karena yang ditampilkan terlebih dahulu adalah deklarasi variabel dengan mendeklarasi sebuah parameter.</p>

## F. Membuat Arrow Function 2

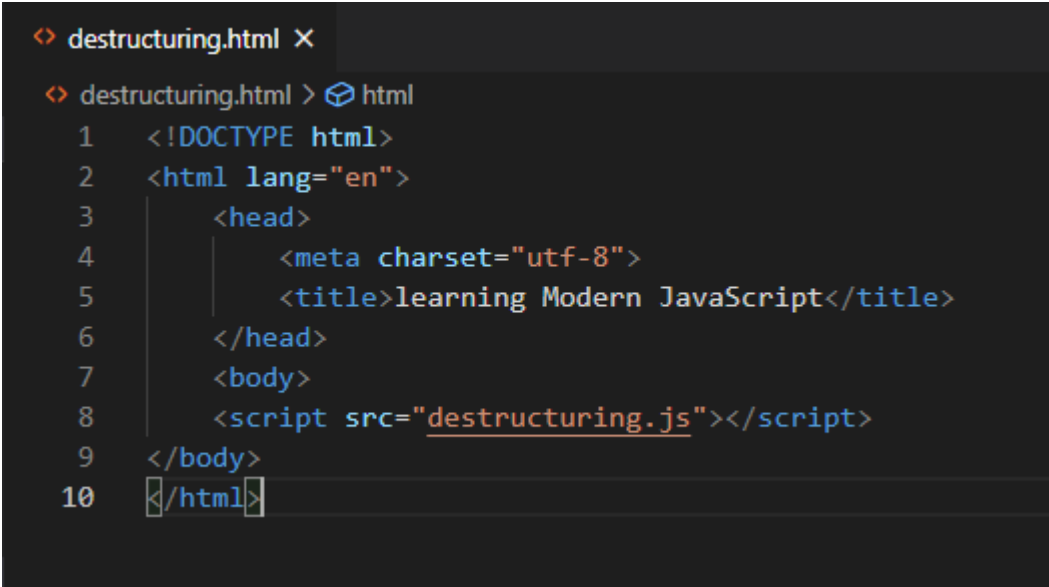
Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	arrowf.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js

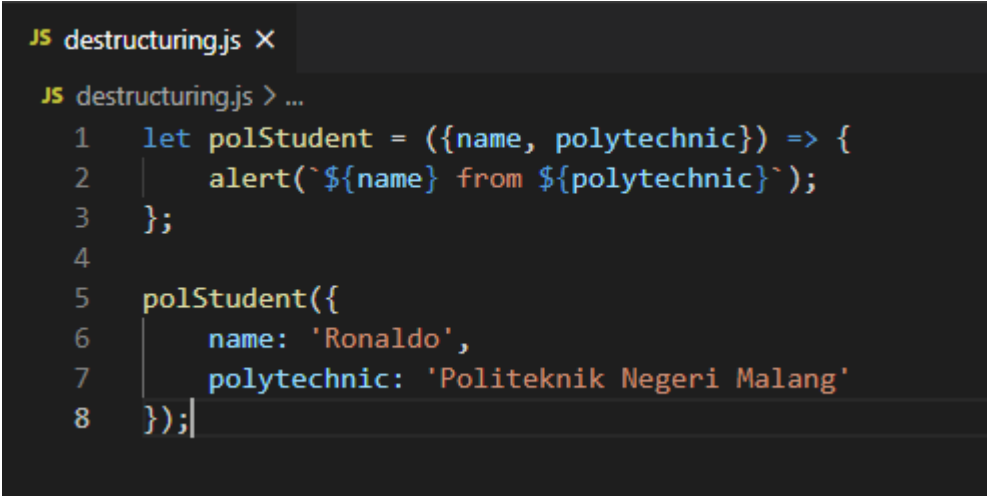
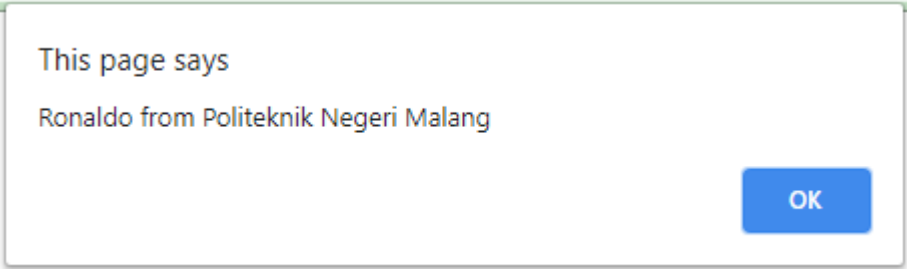


	 <pre> &lt;&gt; arrowf.html X &lt;&gt; arrowf.html &gt; html &gt; head &gt; title 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3    &lt;head&gt; 4      &lt;meta charset="utf-8"&gt; 5      &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6    &lt;/head&gt; 7    &lt;body&gt; 8      &lt;script src="arrowf.js"&gt;&lt;/script&gt; 9    &lt;/body&gt; 10 &lt;/html&gt; </pre>
	<p>arrowf.js -&gt; code pada file ini terdapat sebuah konstanta function disini konstanta gunanya untuk menentukan nilai yang bersifat tetap dan tidak diubah-ubah dengan parameter a dan b kemudian menampilkan parameter2 tersebut menggunakan dialog alert dengan memanggil function serta mendeklarasikan isi dari parameter2 yang sudah di tentukan.</p>
	 <pre> JS arrowf.js X JS arrowf.js &gt; ... 1  // Tanpa arrow 2  // const func = function(a, b){} 3  // return a + b 4  {} 5  // alert(func(3, 3)); 6 7 8  // Menggunakan arrow 9  const func = (a, b) =&gt; { 10    return a + b; 11  }; 12  alert(func(5,4)); </pre>
2	<p>Selanjutnya buka file arrowf.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.</p>

	 <p>Setelah membuka file arrowf.html pada browser maka hasilnya akan menampilkan sebuah dialog alert dari paramater yang pendeklarasian isi dari konstanta dijumlahkan. Maka dari itu hasilnya menampilkan angka '9'.</p>
--	---

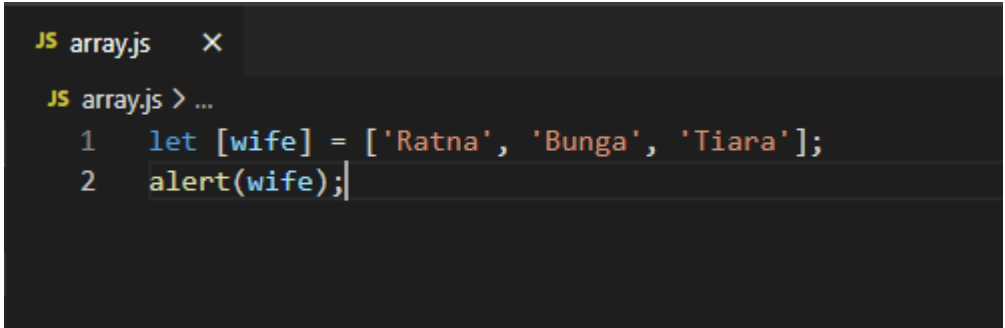
## G. Membuat Destructuring Object

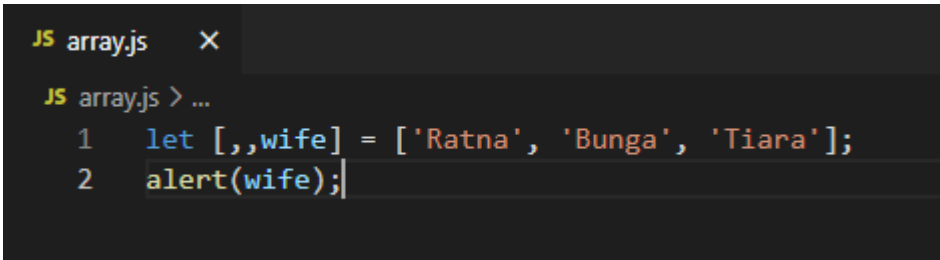
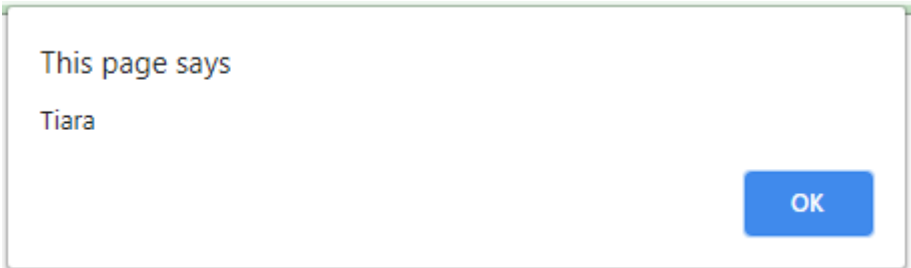
Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	destructuring.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js
	
	destructuring.js -> code pada file ini terdapat let berdeklarasi dengan 2 variabel dimana kemudian variabel2 tersebut ditampilkan dengan menggunakan dialog alert dengan tambahan 'from' diantara 2 variabel tersebut. Kemudian memanggil deklarasi dari let dengan menentukan isi dari 2 properti dari 2 variabel yang sudah ditentukan sebelumnya.

	 <pre> JS destructuring.js X JS destructuring.js &gt; ... 1 let polStudent = ({name, polytechnic}) =&gt; { 2   alert(`\${name} from \${polytechnic}`); 3 }; 4 5 polStudent({ 6   name: 'Ronaldo', 7   polytechnic: 'Politeknik Negeri Malang' 8 }); </pre>
2	Selanjutnya buka file destructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.
	 <p>Setelah membuka file destructuring.html pada browser maka hasilnya akan menampilkan sebuah dialog alert dari 2 variabel yang diantara variabel tersebut ada tambahan 'from' dan variabel2 tersebut sudah dideklarasikan dengan menentukan isi dari deklarasi variabel pada Langkah pemanggilan properti. Maka dari itu hasilnya menampilkan 'Ronaldo from Politeknik Negeri Malang'.</p>

## H. Membuat Destructuring an Array

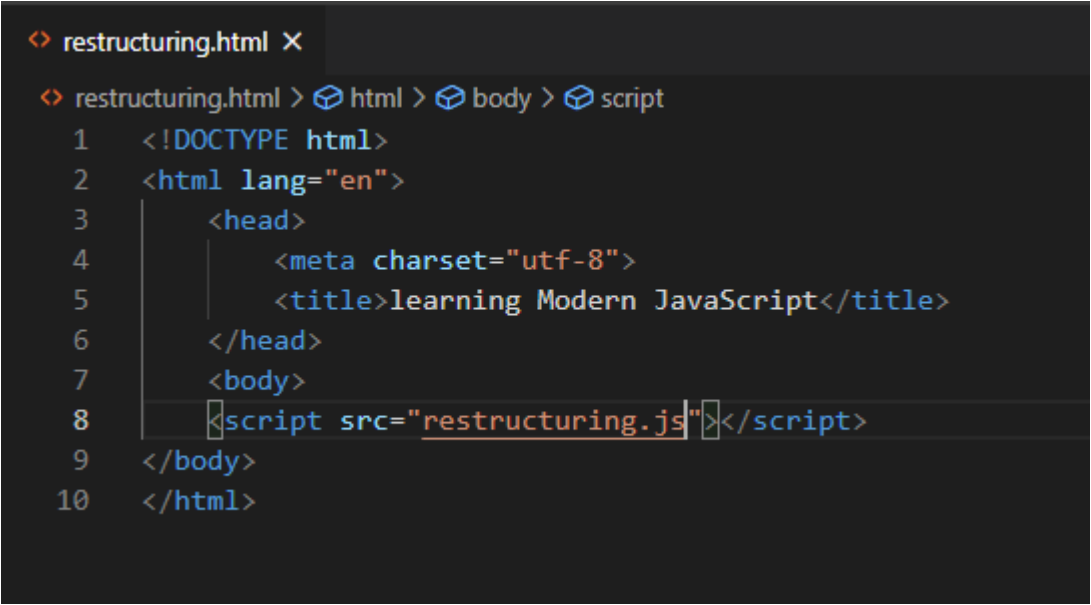
Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	array.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js

	 <pre> &lt;&gt; array.html X &lt;&gt; array.html &gt; html &gt; body &gt; script 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3      &lt;head&gt; 4          &lt;meta charset="utf-8"&gt; 5          &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6      &lt;/head&gt; 7      &lt;body&gt; 8          &lt;script src="array.js"&gt;&lt;/script&gt; 9      &lt;/body&gt; 10 &lt;/html&gt; </pre>
	<p>array.js -&gt; code pada file ini terdapat let dengan mendeklarasikan isi dari variabel array wife bertipe string. Kemudian ditampilkan dengan menggunakan dialog alert dengan memanggil deklarasi variabel dari let sebelumnya.</p>
	 <pre> JS array.js X JS array.js &gt; ... 1  let [wife] = ['Ratna', 'Bunga', 'Tiara']; 2  alert(wife); </pre>
2	<p>Selanjutnya buka file array.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.</p>
	 <p>Setelah melakukan perubahan sedikit code membuka file array.html pada browser maka hasilnya akan menampilkan sebuah dialog alert dari variabel yang sudah dideklarasikan. Maka dari itu hasilnya menampilkan 'Ratna'.</p>

3	Kemudian ganti sedikit code pada array.js seperti dibawah ini
	 <pre> JS array.js  X JS array.js &gt; ... 1  let [,wife] = ['Ratna', 'Bunga', 'Tiara']; 2  alert(wife); </pre>
4	Bukalah kembali, maka hasilnya akan berbeda seperti gambar dibawah ini
	 <p>Setelah melakukan perubahan sedikit code membuka file array.html pada browser maka hasilnya akan menampilkan sebuah dialog alert dari variabel yang sudah dideklarasikan. Disini ada penambahan karakter ‘,’ di dalam deklarasi variabel array. Maka dari itu hasilnya menampilkan ‘Tiara’ bukan ‘Ratna’.</p>

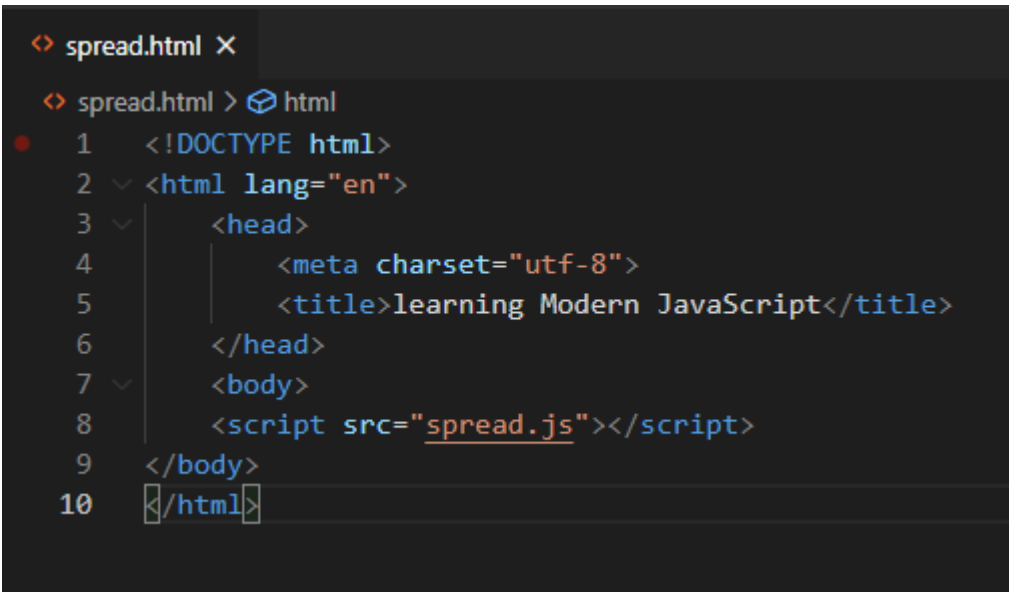
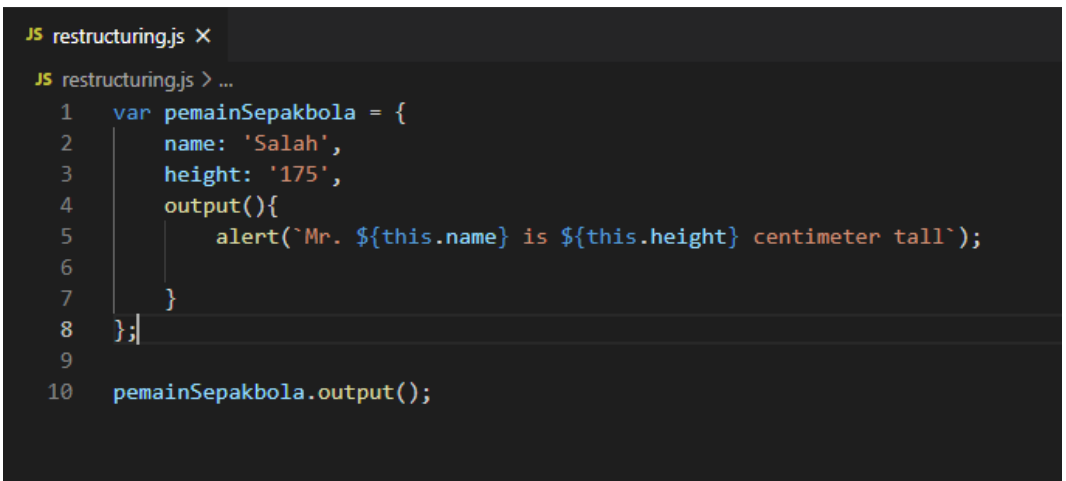
## I. Membuat Restructuring

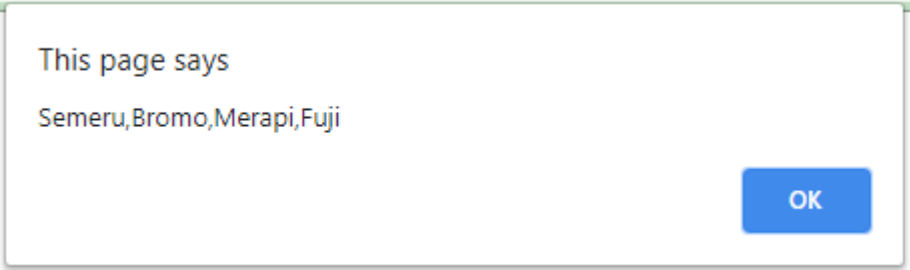
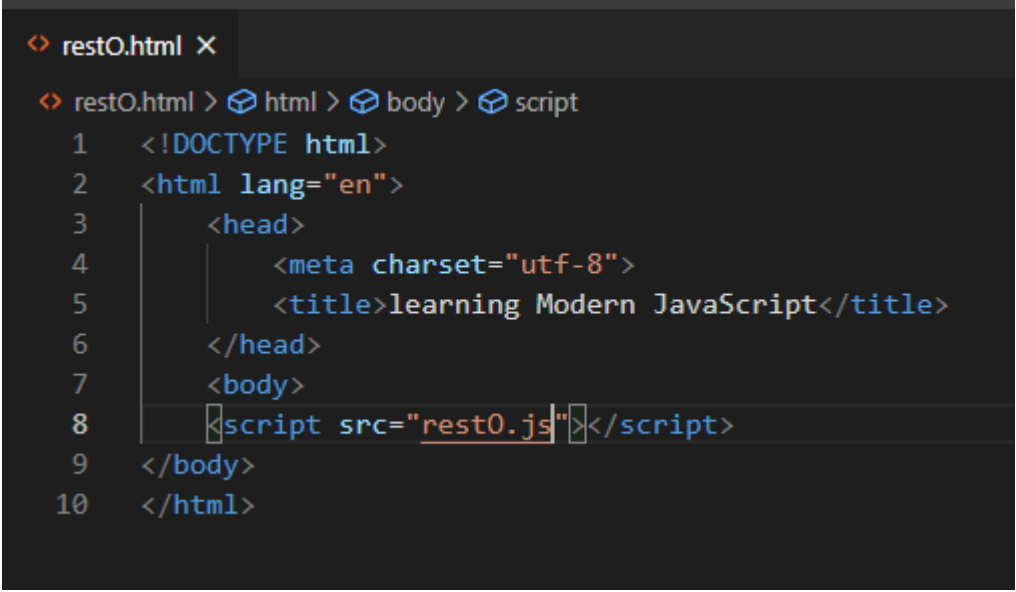
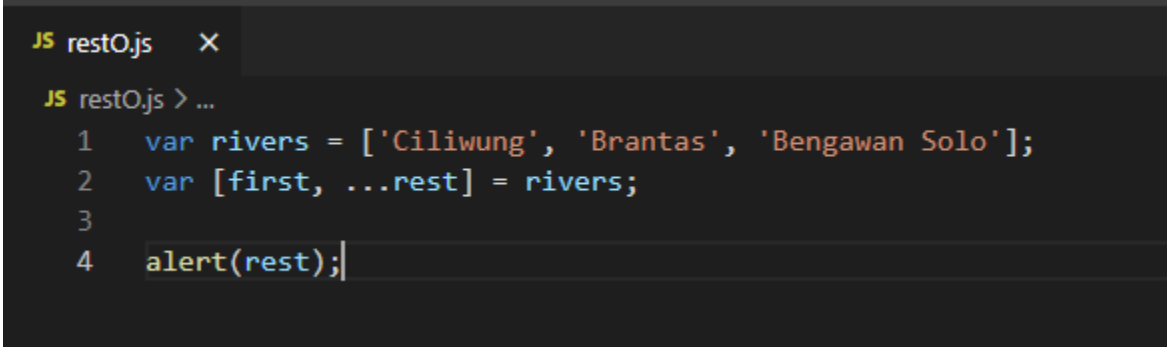
Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	restructuring.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js

	 <pre> &lt;&gt; restructuring.html X &lt;&gt; restructuring.html &gt; html &gt; body &gt; script 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3      &lt;head&gt; 4          &lt;meta charset="utf-8"&gt; 5          &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6      &lt;/head&gt; 7      &lt;body&gt; 8          &lt;script src="restructuring.js"&gt;&lt;/script&gt; 9      &lt;/body&gt; 10 &lt;/html&gt; </pre>
	<p>restructuring.js -&gt; code pada file ini terdapat var yang mendeklarasikan 2 properti dimana 2 properti tersebut juga di deklarasikan dengan bertipe string dan menampilkannya menggunakan dialog alert dengan tambahan beberapa kata untuk kalimat tambahan diantara property tersebut.</p>
	 <pre> JS restructuring.js X JS restructuring.js &gt; ... 1  var pemainSepakbola = { 2      name: 'Salah', 3      height: '175', 4      output(){ 5          alert(`Mr. \${this.name} is \${this.height} centimeter tall`); 6      } 7  }; 8  ; 9 10 pemainSepakbola.output(); </pre>
2	<p>Selanjutnya buka file restructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.</p>
	

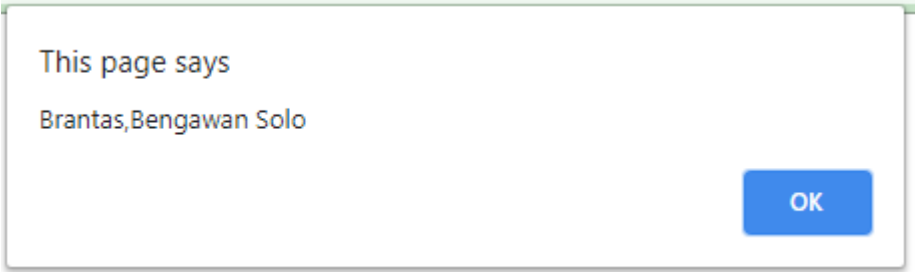
	Setelah membuka file restructuring.html pada browser maka hasilnya akan menampilkan sebuah dialog alert dari 2 property dengan tambahan kalimat diantara property tersebut. Maka dari itu hasilnya menampilkan 'Mr. Salah is 175 centimeter tall'.
--	--

## J. Membuat Spread and Rest operator

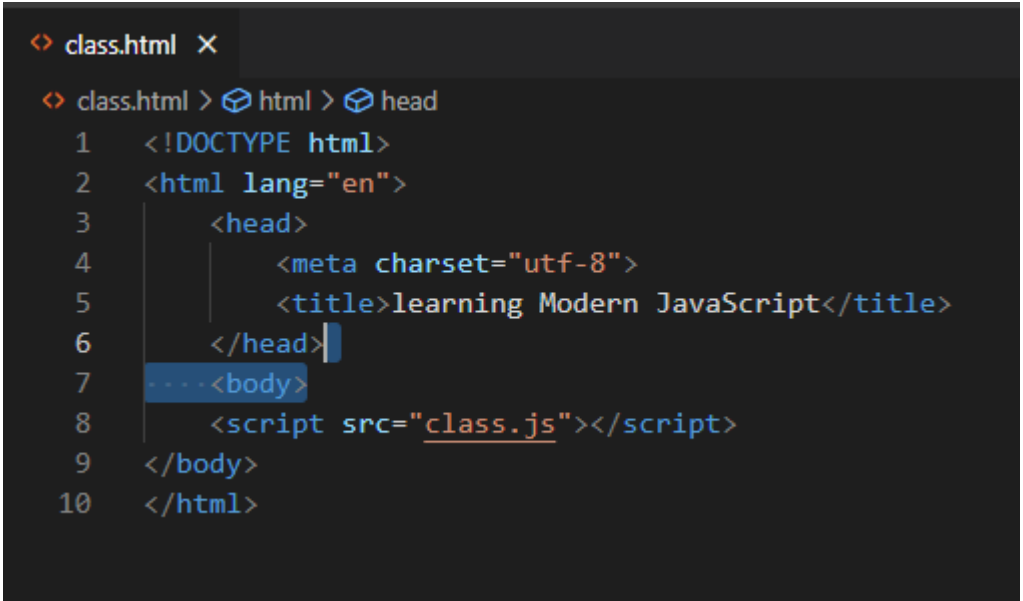
Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	spread.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js
	 <pre> spread.html X spread.html &gt; html 1 &lt;!DOCTYPE html&gt; 2 &lt;html lang="en"&gt; 3   &lt;head&gt; 4     &lt;meta charset="utf-8"&gt; 5     &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6   &lt;/head&gt; 7   &lt;body&gt; 8     &lt;script src="spread.js"&gt;&lt;/script&gt; 9   &lt;/body&gt; 10 &lt;/html&gt; </pre>
	spread.js -> code pada file ini terdapat beberapa var yang mendeklarasikan isi dari pendeklarasian var dengan berbentuk array. Kemudian membuat deklarasi var untuk melakukan pemanggilan semua deklarasi var sebelumnya dan ditampilkan menggunakan dialog alert .
	 <pre> JS restructuring.js X JS restructuring.js &gt; ... 1 var pemainSepakbola = { 2   name: 'Salah', 3   height: '175', 4   output(){ 5     alert(`Mr. \${this.name} is \${this.height} centimeter tall`); 6   } 7 } 8 }; 9 10 pemainSepakbola.output(); </pre>


2	Selanjutnya buka file spread.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.
	 <p>Setelah membuka file spread.html pada browser maka hasilnya akan menampilkan sebuah dialog . Maka dari itu hasilnya menampilkan 'Semeru,Bromo,Merapi,Fuji'.</p>
3	Untuk melakukan praktek rest operation, buatlah dua file .html dan .js seperti dibawah ini
	restO.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js
	 <pre> &lt;&gt; restO.html X &lt;&gt; restO.html &gt; html &gt; body &gt; script 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3    &lt;head&gt; 4      &lt;meta charset="utf-8"&gt; 5      &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6    &lt;/head&gt; 7    &lt;body&gt; 8      &lt;script src="rest0.js"&gt;&lt;/script&gt; 9    &lt;/body&gt; 10  &lt;/html&gt; </pre>
	restO.js -> code pada file ini terdapat var yang dideklarasikan dengan isi dalam bentuk array
	 <pre> JS restO.js X JS restO.js &gt; ... 1  var rivers = ['Ciliwung', 'Brantas', 'Bengawan Solo']; 2  var [first, ...rest] = rivers; 3 4  alert(rest); </pre>

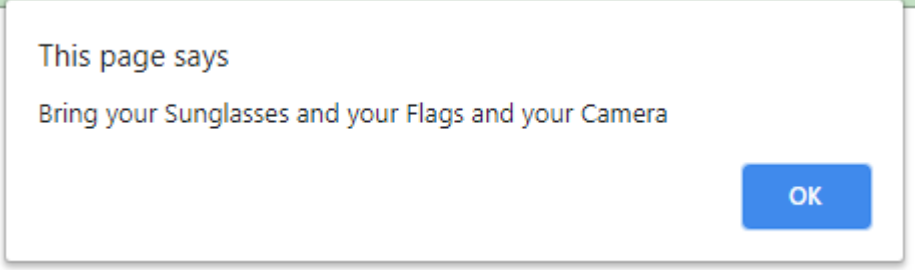


4	Selanjutnya buka file restO.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.
	 <p>Setelah membuka file restO.html pada browser maka hasilnya akan menampilkan sebuah dialog . Maka dari itu hasilnya menampilkan 'Brantas,Bengawan Solo'.</p>

## K. Membuat Classes Constructor and Super

Langkah	Keterangan
1	Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya
	class.html -> file ini digunakan untuk menampilkan hasil dari penulisan code file.js
	 <pre> class.html class.html &gt; html &gt; head 1 &lt;!DOCTYPE html&gt; 2 &lt;html lang="en"&gt; 3   &lt;head&gt; 4     &lt;meta charset="utf-8"&gt; 5     &lt;title&gt;learning Modern JavaScript&lt;/title&gt; 6   &lt;/head&gt; 7   &lt;body&gt; 8     &lt;script src="class.js"&gt;&lt;/script&gt; 9   &lt;/body&gt; 10 &lt;/html&gt; </pre>
	class.js -> code pada file ini terdapat 2 class yaitu super class dan sub class. dimana masing2 class memiliki fungsi yang berbeda. untuk super class dia mendetailkan semua secara detail . sedangkan untuk sub class dia melakukan extends pada class super class maka dari itu pada class sub class tinggal melakukan penambahan jika perlu. karena sub class berkaitan dengan super class.

	<pre> JS class.js  X JS class.js &gt; ... 1  //super class 2  class Holiday { 3      constructor(destination, days){ 4          this.destination = destination; 5          this.days = days; 6      } 7 8 9      info(){ 10         alert(`\${this.destination} will take \${this.days} days.`); 11     } 12 } 13 14 //sub class 15 class Expedition extends Holiday { 16     constructor(destination, days, gear){ 17         super(destination, days); 18         this.gear = gear; 19     } 20 21     info(){ 22         super.info(); 23         alert(`Bring your \${this.gear.join(' and your ')}`); 24     } 25 } 26 27 const tripWithGear = new Expedition('Semeru', 10, ['Sunglasses', 'Flags', 'Camera']); 28 tripWithGear.info(); </pre>
2	<p>Selanjutnya buka file class.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.</p>
	 <p>Setelah membuka file class.html pada browser maka hasilnya akan menampilkan sebuah dialog . Maka dari itu hasilnya menampilkan 'Semeru will take 10 days'.</p>
3	<p>Klik OK, maka akan muncul page selanjutnya dengan pesan sebagai berikut.</p>

	 <p>Setelah membuka file class.html pada browser maka hasilnya akan menampilkan sebuah dialog . Maka dari itu hasilnya menampilkan 'Bring yout Sunglasses and your Flags and your Camera.</p>
--	---

Link github : [GitHub - dinariskyas/PemrogramanBerbasisFramework2021-ModernJavaScript01](https://github.com/dinariskyas/PemrogramanBerbasisFramework2021-ModernJavaScript01)