



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-2: Typescript

Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Pemrograman Web Lanjut

Februari 2019

Topik

- Pengenalan Dasar typescript dan pendeklarasiannya.

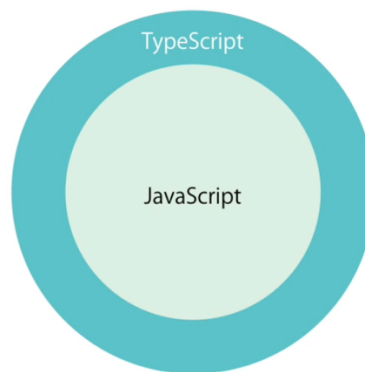
Tujuan

Mahasiswa diharapkan dapat:

1. Memahami Dasar penggunaan typescript.
2. Melakukan pendeklarasian typescript.

Pendahuluan

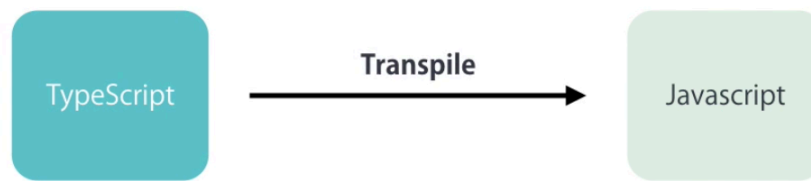
Typescript adalah bahasa pemrograman berbasis JavaScript yang menambahkan fitur strong-typing & konsep pemrograman OOP klasik (class, interface). Di dalam dokumentasinya, TypeScript disebut sebagai super-set dari JavaScript, artinya semua kode JavaScript adalah kode TypeScript juga. Bahasa pemrograman ini menawarkan class, module, dan interface yang membuat developer bisa mengembangkan aplikasi kompleks dengan lebih mudah. Hal inilah yang membedakannya dengan javascript



Typescript memiliki beberapa fitur :

1. Strong typing or static typing
Jika menggunakan bahasa C# dan Java, akan dilakukan pendefinisian variabel sehingga diperlukan penulisan variabel yang spesifik. Pada typescript hal tersebut opsional namun jika menggunakan fitur tersebut akan membuat aplikasi yang dibangun akan lebih terdeteksi dan membuat lebih mudah menemukan error pada saat *debugging*.
2. Object oriented
Typescript juga menggunakan konsep object oriented seperti class, interface, constructors, access modifiers seperti public dan private, fills, properties, generic, dan lain-lain.
3. Compile-time errors
Dengan menggunakan typescript dapat mengerti tentang adanya error pada saat compile dan membenahinya sebelum mendeploy aplikasi yang dibuat.

Browser tidak akan mengenali typescript sehingga agar lebih akurat dibutuhkan mengcompilernya dahulu.



agar browser mengenali typescript yang diberikan, maka harus dilakukan transpile pada code typescript ke dalam javascript. Hal tersebut merupakan bagian dalam membangun sebuah aplikasi sehingga browser dapat mengerti code nya.

Praktikum – Bagian 1: Instalasi Typescript

Langkah	Keterangan
1	Duplicate folder hasil prak1 dan rubah namanya menjadi prak2
2	Buka terminal pada folder yang sudah dibuat pada langkah 1 yaitu prak2
3	Ketik perintah berikut : npm install -g typescript <pre>Yushintias-MacBook-Pro:prak2 YushintiaPramitarini\$ npm install -g typescript</pre>
4	Catat hasilnya (soal 1)
5	Kemudian cek version dari typescript yang terinstall <pre>Yushintias-MacBook-Pro:prak2 YushintiaPramitarini\$ tsc --version Version 3.3.1</pre>
6	Catat hasil nya (soal2)
7	Buatlah direktori baru dengan nama ts-hello dengan perintah : mkdir ts-hello <pre>Yushintias-MacBook-Pro:prak2 YushintiaPramitarini\$ mkdir ts-hello</pre>
8	Kemudian masuklah ke direktori pada no 7 dengan perintah cd ts-hello dan buat lah file baru dengan nama main.ts pada direktori ts-hello
9	Tulislah code dibawah ini : <pre>TS main.ts x 1 2 // tslint:disable-next-line:no-shadowed-variable 3 function log(message) { 4 console.log(message); 5 } 6 7 let message = 'Hello World'; 8 9 log(message); 10</pre>

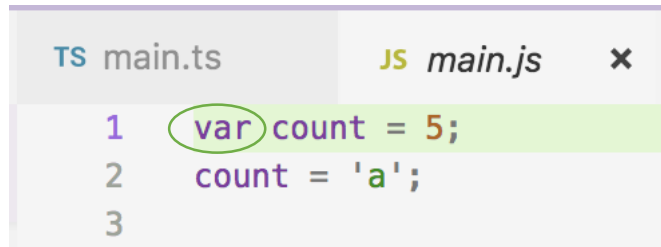
10	Kemudian buka terminal dan lakukan perintah : tsc main.ts untuk melakukan proses transpile file typescript kedalam javascript
11	Cek apakah sudah terbuat file main.js, lakukan perintah ls
12	Catat hasil nya (soal 3)
13	Buka file javascript hasil dari no 10, perhatikan apakah sama isi dari file main.ts dan main.js ? (soal 4)
14	lakukan execute pada file main.js dengan perintah node main.js . catat hasil nya (soal 5).

Praktikum – Bagian 2: Declaring Variables

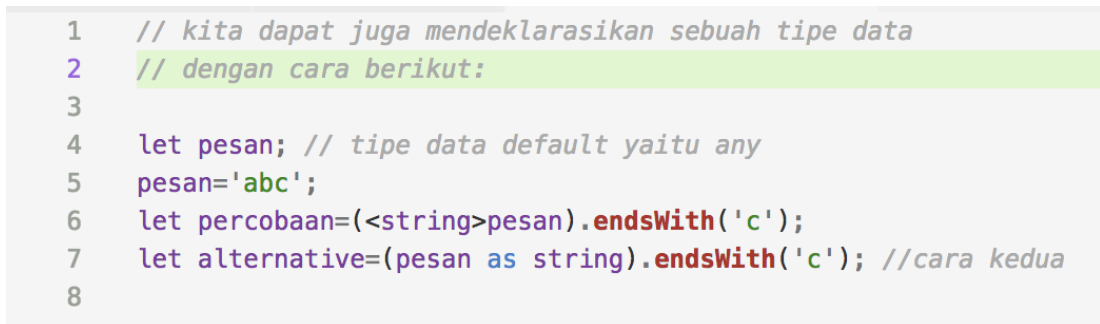
Langkah	Keterangan
1	<p>Buka file main.ts , kemudian tuliskan code seperti berikut :</p> <pre> TS main.ts x JS main.js 1 2 function doSomething() { 3 for (let i = 0; i < 5; i++) { 4 console.log(i); 5 } 6 console.log('Finally: ' + i); 7 } 8 doSomething(); 9 </pre>
2	Remove file main.js hasil praktikum bagian 1 dengan perintah rm main.js , cek apakah sudah terhapus atau belum. Catat hasilnya (soal 6)
3	Lakukan transpile file main.ts dengan perintah yang ada pada praktikum bagian 1 langkah no 10. Catat hasilnya apakah sudah terbuat file javascriptnya. (soal 7)
4	Jika sudah terbuat, execute file javascript yang sudah terbuat pada langkah 3. Kemudian catat hasilnya (soal 8)

Praktikum - Bagian 3: Types


Langkah	Keterangan
1	<p>Buka file main.ts kemudian tuliskan listing program berikut :</p> <pre> TS main.ts x 1 let count = 5; 2 count = 'a'; </pre>

2	Mari kita perhatikan code pada langkah 1, terlihat bahwa terjadi error pada line 2. Namun dengan kita melakukan transpile file typescript ke dalam file javascript, kesalahan tersebut akan dibenarkan dengan sendirinya pada file javascript. Lakukan proses transpile file main.ts dengan perintah tsc main.ts . Catat hasilnya (soal 9)
3	Kemudian buka file main.js, hasilnya harusnya seperti berikut : <div data-bbox="510 353 1173 598" data-label="Code-Block">  <pre> TS main.ts JS main.js x 1 var count = 5; 2 count = 'a'; 3 </pre> </div>

Praktikum – Bagian 4: Type Assertion

Langkah	Keterangan
1	buka file main.ts, kemudian ketikkan code berikut ini : <div data-bbox="312 880 1417 1202" data-label="Code-Block">  <pre> 1 // kita dapat juga mendeklarasikan sebuah tipe data 2 // dengan cara berikut: 3 4 let pesan; // tipe data default yaitu any 5 pesan='abc'; 6 let percobaan=<string>pesan).endsWith('c'); 7 let alternative=(pesan as string).endsWith('c'); //cara kedua 8 </pre> </div>
2	Kemudian buat file main.js (jangan lupa untuk meremove file main.js sebelumnya) kemudian execute seperti praktikum sebelumnya. Catat hasilnya (soal 10)

Praktikum – Bagian 5: Arrow Function

Langkah	Keterangan
1	buka file main.ts, kemudian ketikkan code berikut ini : <div data-bbox="587 1583 1134 1874" data-label="Code-Block">  <pre> 1 2 let log = function pesan(){ 3 console.log(pesan); 4 } 5 6 // code diatas bisa disingkat menggunakan function arrow 7 let doLog=(pesan)=>{ 8 console.log(pesan); 9 } 10 11 // atau seperti berikut 12 let doLog1=(pesan)=>console.log(pesan); 13 </pre> </div>
2	Kemudian buat file main.js (jangan lupa untuk meremove file main.js sebelumnya) kemudian execute seperti praktikum sebelumnya. Catat hasilnya (soal 11)

Praktikum – Bagian 6: Interface

Langkah	Keterangan
1	<p>buka file main.ts, kemudian ketikkan code berikut ini :</p> <pre>1 interface Point{ 2 x:number; 3 y:number; 4 } 5 6 let drawPoint=(point:Point)=>{ 7 // 8 } 9 10 drawPoint({ 11 x:1, 12 y:3, 13 })</pre>
2	<p>Kemudian buat file main.js (jangan lupa untuk meremove file main.js sebelumnya) kemudian execute seperti praktikum sebelumnya. Catat hasilnya (soal 12)</p>

Praktikum – Bagian 7: Classes

Langkah	Keterangan
1	<p>buka file main.ts, kemudian ketikkan code berikut ini :</p> <pre>1 2 class Pointku{ 3 a:number; 4 b:number; 5 6 draw(){ 7 // 8 } 9 10 getDistance(another:Pointku){ 11 // 12 } 13 }</pre>
2	<p>Kemudian buat file main.js (jangan lupa untuk meremove file main.js sebelumnya) kemudian execute seperti praktikum sebelumnya. Catat hasilnya (soal 13)</p>

Praktikum – Bagian 8: Objects

Langkah	Keterangan
1	<p>buka file main.ts, kemudian ketikkan code berikut ini :</p> <pre>1 // jika kita jalankan code berikut maka akan terjadi error 2 // hal ini disebabkan karena coins tidak mengenal property a dan b 3 // maka buatlah objek baru dengan nama oboekcoins line 18 4 5 class Pointku1{ 6 a:number; 7 b:number; 8 9 draw(){ 10 console.log('X' + this.a + ',Y'+this.b); 11 } 12 13 getDistance(another:Pointku1){ 14 // 15 } 16 } 17 18 let coins:Pointku1 19 coins.draw(); 20 21 // let objekcoins= new Pointku1; 22 // objekcoins.a=1; 23 // objekcoins.b=3; 24 // objekcoins.draw(); 25 26</pre>
2	<p>Kemudian buat file main.js (jangan lupa untuk meremove file main.js sebelumnya) kemudian execute seperti praktikum sebelumnya. Catat hasilnya (soal 14)</p>