

ONGIS SCHOOL
FRONTEND DEV

SESI

1. KONSEP WEB COMPONENTS & TASK RUNNER
2. CUSTOM ELEMENTS
3. SHADOW DOM
4. HTML IMPORT
5. HTML TEMPLATE
6. FINISHING PROJECT

SESI 1 : KONSEP WEB COMPONENTS & TASK RUNNER

KONSEP WEB COMPONENTS

Web component adalah konsep kerja / kumpulan API Platform web yang memungkinkan Frontend Dev membuat tag HTML khusus dengan penamaan sesuai keinginan pengguna, yang dapat digunakan kembali pada setiap bagian website, atau dikelompokkan menjadi satu pustaka untuk digunakan pada website atau aplikasi website. Komponen dan widget pada konsep ini khusus dibangun diatas standar pengembangan web, Frontend Dev tidak perlu khawatir akan dukungan browser karena hampir bekerja cukup baik di seluruh browser modern, juga dapat digabungkan penggunaannya dengan pustaka atau kerangka Javascript yang sesuai dengan HTML

KONSEP WEB COMPONENTS

Dalam pengembangan project, web components ada baiknya didasarkan pada standar web yang ada, semua fitur untuk mendukung konsep ini ditambahkan ke spesifikasi HTML dan DOM. Frontend Dev leluasa mengembangkan HTML dengan element baru dengan fitur yang disesuaikan dengan website atau aplikasi web yang sedang dikembangkan.

SPEKIFIKASI UTAMA WEB COMPONENTS :

1. CUSTOM ELEMENTS
2. SHADOW DOM
3. HTML IMPORT
4. HTML TEMPLATE

TAG HTML STANDAR:

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <title> ... </title>  
    <meta charset="UTF-8">  
    <meta name="" content="width=device-width, initial-scale=1">  
    <link href=".css" rel="stylesheet">  
    <script src=".js" type="text/javascript"></script>  
  </head>  
  <body>
```

TAG HTML STANDAR:

<header>

...

</header>

<section>

...

</section>

<footer>

...

</footer>

</body>

</html>

STRUKTUR FOLDER :

css

-----.css

js

-----.js

image

fonts

index.html

TAG HTML DENGAN KONSEP WEB COMPONENTS:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

```
    <title> ... </title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="" content="width=device-width, initial-scale=1">
```

```
    <link href=".css" rel="stylesheet">
```

```
    <script src=".js" type="text/javascript"></script>
```

```
  <link rel="import" href="./app/header.html">
```

```
  <link rel="import" href="./app/section.html">
```

```
  <link rel="import" href="./app/footer.html">
```

TAG HTML DENGAN KONSEP WEB COMPONENTS:

atau

```
<script src="/app/header.js" type="text/javascript"></script>
```

```
<script src="/app/section.js" type="text/javascript"></script>
```

```
<script src="/app/footer.js" type="text/javascript"></script>
```

```
</head>
```

```
<body>
```

```
<header></header>
```

```
<section></section>
```

```
<footer></footer>
```

```
</body>
```

```
</html>
```

STRUKTUR FOLDER :

app

----- .html / .js

css

----- .css

js

----- .js

image

fonts

index.html

KELEBIHAN MENGGUNAKAN KONSEP WEB COMPONENTS:

1. Pengembangan nya seiring dengan standarisasi HTML
2. Konsep modul, artinya memecah keseluruhan bagian web menjadi bagian bagian kecil sehingga jauh lebih memudahkan Frontend Dev dalam penanganan project.
3. Mengurangi masalah render blocking.
4. Memungkinkan website / aplikasi web bekerja lebih ringan karena setiap bagian website / aplikasi web hanya memanggil fungsi yang diperlukan yang diperlukan.
5. ...

DUKUNGAN BROWSER :

1. Chrome
2. Opera
3. Safari
4. Firefox
5. Edge / IE9+
6. Android Browser
7. Apple Mobile Safari (+iOS8)

INTEGRASI PROJECT

STRUKTUR FOLDER PROJECT:

project

----- dist

----- src

----- CSS

----- sass

----- style.sass

----- components

----- _file.sass

----- js

----- file.js

STRUKTUR FOLDER PROJECT:

- img
- app
- file.js / file.html
- fonts
- index.html

TASK RUNNER

APA ITU TASK RUNNER :

Tool pembantu mengerjakan banyak hal secara otomatis

TUJUAN PENGGUNAAN TASK RUNNER :

Kira-kira apa saja kebutuhan khusus sebuah frontend Dev untuk membangun sebuah halaman website / aplikasi web?

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

node.js & npm

library packager

instalasi via terminal :

linux ubuntu :

```
curl -sL https://deb.nodesource.com/setup\_\*.x | sudo -E bash -  
sudo apt-get install -y nodejs
```

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

windows:

...

mac :

`brew install node`

Node JS source :

<https://nodejs.org/en/download/>

<https://nodejs.org/en/download/package-manager/>

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

- a) setelah instalasi buka terminal menuju ke folder project
- b) ketik perintah

`npm init`

- c) isi sesuai kebutuhan pengguna
- d) akhiri dengan jawaban “yes”

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

sass

css preprocessor

instalasi :

linux :

```
sudo apt-get install ruby
```

```
sudo su -c "gem install sass"
```


INSTALASI TASK RUNNER & INTEGRASI PROJECT:

windows :

install ruby package hasil download dari <http://rubyinstaller.org/>
buka terminal ketik

```
gem install sass
```

mac :

```
gem install sass
```

doc :

<http://sass-lang.com/install>

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp

instalasi

```
npm install gulp --save-dev
```

doc :

<https://www.npmjs.com/package/gulp>

Tugas : (?) (1)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-header

instalasi :

```
npm install gulp-header --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-header>

Tugas : (?) (2)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-sourcemaps

instalasi :

```
npm install gulp-sourcemaps --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-sourcemaps>

Tugas : (?) (3)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-sass

instalasi :

```
npm install gulp-sass --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-sass>

Tugas : (?) (4)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-cssnano

instalasi :

```
npm install gulp-cssnano --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-cssnano>

Tugas : (?) (5)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-autoprefixer

instalasi :

```
npm install gulp-autoprefixer --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-autoprefixer>

Tugas : (?) (6)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-concat

instalasi :

```
npm install gulp-concat --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-concat>

Tugas : (?) (7)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-jshint

instalasi :

```
npm install gulp-jshint --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-jshint>

Tugas : (?) (8)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-uglify

instalasi :

```
npm install gulp-uglify --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-uglify>

Tugas : (?) (9)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-watch

instalasi :

```
npm install gulp-watch --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-watch>

Tugas : (?) (10)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-copy

instalasi :

```
npm install gulp-copy --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-copy>

Tugas : (?) (11)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

gulp-zip

instalasi :

```
npm install gulp-zip --save-dev
```

doc:

<https://www.npmjs.com/package/gulp-zip>

Tugas : (?) (12)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

browser-sync

instalasi :

```
npm install browser-sync --save-dev
```

doc:

<https://www.npmjs.com/package/>

Tugas : (?) (13)

INSTALASI TASK RUNNER & INTEGRASI PROJECT:

jshint

instalasi :

npm install jshint --save-dev

doc :

<https://www.npmjs.com/package/jshint>

Tugas : (?) (14)

KONFIGURASI TASK RUNNER

1. edit package.json

```
"author": "Mifan Twan Ardana",
```

menjadi

```
"author": {  
  "name": "Mifan Twan Ardana",  
  "homepage": "http://vidiproject.com/"  
},
```


KONFIGURASI TASK RUNNER

2. buat file baru dengan nama gulpfile.js
3. Isi file gulpfile.js dengan

```
'use strict';  
  
//global variable  
  
var      sources      = require('./package.json'),  
        gulp          = require('gulp'),  
        browserSync   = require('browser-sync').create(),  
        header        = require('gulp-header'),  
        sourcemaps     = require('gulp-sourcemaps'),
```

KONFIGURASI TASK RUNNER

```
banner = [  
    '/**',  
    ' * <%= sources.name %> - <%= sources.description %>',  
    ' * @author <%= sources.author.name %>',  
    ' * @version v<%= sources.version %>',  
    ' * @link <%= sources.author.homepage %>',  
    ' * @license <%= sources.license %>',  
    ' */',  
    ''  
].join('\n');
```

KONFIGURASI TASK RUNNER

```
var browser_support = [  
    'last 2 versions',  
    '> 5%',  
    'Firefox ESR',  
    'ie >= 10',  
    'ie_mob >= 10',  
    'ff >= 30',  
    'chrome >= 34',  
    'safari >= 7',  
    'opera >= 23',  
    'ios >= 7',  
    'android >= 4.4',  
    'bb >= 10'  
],
```

KONFIGURASI TASK RUNNER

```
date = new Date().toISOString().slice(0,10);
```

```
//css-task variable
```

```
var sass = require('gulp-sass'),  
    cssnano = require('gulp-cssnano'),  
    autoprefixer = require('gulp-autoprefixer'),
```

```
//js-task variable
```

```
concat = require('gulp-concat'),  
jshint = require('gulp-jshint'),  
uglify = require('gulp-uglify'),
```

KONFIGURASI TASK RUNNER

```
//staging & delivery variable
```

```
    copy          = require('gulp-copy'),
```

```
    zip           = require('gulp-zip');
```

KONFIGURASI TASK RUNNER

```
// css task
gulp.task('sass', function() {
  return gulp.src("./src/sass/**/*.sass")
    .pipe(sourcemaps.init())
    .pipe(sass({outputStyle: 'compressed', errLogToConsole: true}).on('error', sass.logError))
    .pipe(concat(sources.name + '.min.css'))
    .pipe(cssnano({autoprefixer: {browsers: browser_support, add: true} }))
    .pipe(sourcemaps.write('../maps'))
    .pipe(header(banner, {sources : sources}))
    .pipe(gulp.dest("./src/css/"))
    .pipe(browserSync.stream())
});
```

KONFIGURASI TASK RUNNER

```
// javascript task
gulp.task('javascript', function() {
    return gulp.src('./src/js/**/*.js')
        .pipe(jshint())
        .pipe(jshint.reporter('default'))
        .pipe(jshint.reporter('fail'))
        .pipe(sourcemaps.init())
        .pipe(concat(sources.name + '.min.js'))
        .pipe(uglify())
        .pipe(sourcemaps.write('../maps'))
        .pipe(header(banner, {sources : sources}))
        .pipe(gulp.dest('./src/app/'))
        .pipe(browserSync.stream())
});
```

KONFIGURASI TASK RUNNER

```
// html task
gulp.task('html', function() {
    return gulp.src('./src/*.html')
        .pipe(gulp.dest('./src'))
        .pipe(browserSync.stream());
});
```


KONFIGURASI TASK RUNNER

```
// static server & task watch
gulp.task('default', function() {
    browserSync.init({
        server: "./src"
    });
    gulp.watch('src/sass/**/*.sass', function (event) {
        console.log(event);
        gulp.start('sass');
    });
});
```

KONFIGURASI TASK RUNNER

```
gulp.watch('src/js/**/*.js', function (event) {  
    console.log(event);  
    gulp.start('javascript');  
});  
gulp.watch('src/**/*.html', function (event) {  
    console.log(event);  
    gulp.start('html').on('change', browserSync.reload);  
});  
});
```

KONFIGURASI TASK RUNNER

```
// delivery & compress ( integrated with web & apps )  
gulp.task('dist', function(){  
    return gulp.src('./src/**/*')  
        .pipe(zip( 'prod_' + sources.name + '_' + date + '.zip'))  
        .pipe(gulp.dest('./dist'))  
});
```

KONFIGURASI TASK RUNNER

4. buat file global.js di folder src/js

5. isi dengan :

```
console.log('test global js');
```

6. buat file style.sass di folder src/sass

7. isi dengan :

```
@import components/global
```

8. buat folder baru dengan nama components di folder src/sass

KONFIGURASI TASK RUNNER

9. buat file `_global.sass` di folder `src/sass/components` tadi

10. isi dengan :

*

padding: 0

margin: 0

11. ketik perintah :

gulp

KONFIGURASI TASK RUNNER

12. perhatikan aktifitas terminal setiap kali terjadi event “add” , “change” atau “unlink” pada saat pekerjaan dev berlangsung.
13. keluar gulp dengan keyboard key :

CTRL + C

14. ketik perintah pada terminal :

gulp dist

(?) (15) alur task runner

NEXT ???
CUSTOM ELEMENT

CREDIT:

<http://webcomponents.org>

<http://nodejs.org>

<http://npmjs.com>

<http://jshint.com>

<http://sass-lang.com>

GLOSARIUM :

DOM (DOCUMENT OBJECT MODEL) adalah kumpulan aturan yang diperlukan javascript untuk memanipulasi elemen apapun yang tampil dalam halaman website / aplikasi web.