

6

Polymer : Struktur folder dan Layout

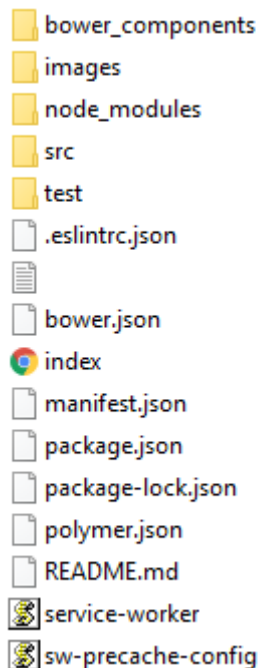
Objektif :

- Memahami struktur folder Polymer
 - Memahami layout Polymer
-

5.1 Struktur folder Polymer

Polymer merupakan salah satu webcomponent yang dimiliki Google. Webcomponent adalah kumpulan fitur pada web browser yang memungkinkan untuk membuat widget (atau yang lebih dikenal dengan istilah component) dalam pengembangan web. Layaknya konsep plugin pada wordpress atau CMS serupa, component ini siap dipakai ulang pada halaman HTML manapun tanpa memerlukan library atau framework tambahan..

Sebagai sebuah webcomponent, Struktur folder pada Polymer sendiri terdiri dari beberapa directory, antara lain : bower_components, images, node_modules, src, dan test.

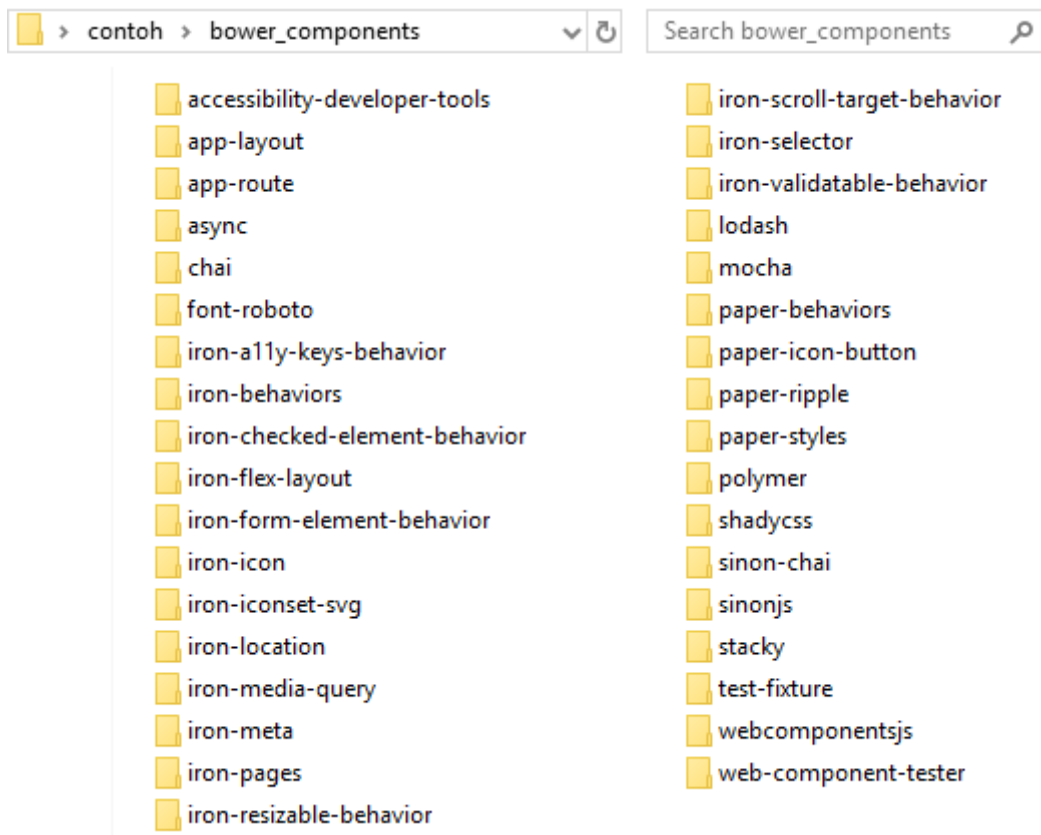


Gambar 6.1 Struktur folder pada Polymer Starter kit

1. Folder bower_components

Folder bower_components adalah folder yang terbentuk setelah menginstall beberapa package, Bower akan membuat sebuah direktory baru dengan nama bower_components

dimana semua package yang diinstall akan berada di dalamnya. Misalnya jika menginstall bower di sebuah diretory dengan nama contoh maka susunan directorynya akan menjadi seperti berikut:

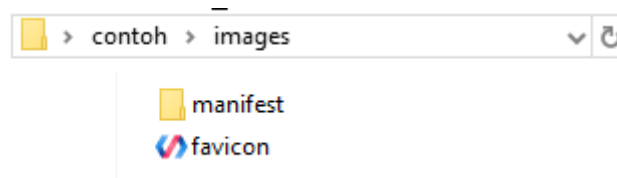


Gambar 6.2 Directory bower_components

Pada directory bower_components ini komponen web berada, baik komponen web yang sudah ada maupun komponen web yang akan ditambahkan.

2. Folder images

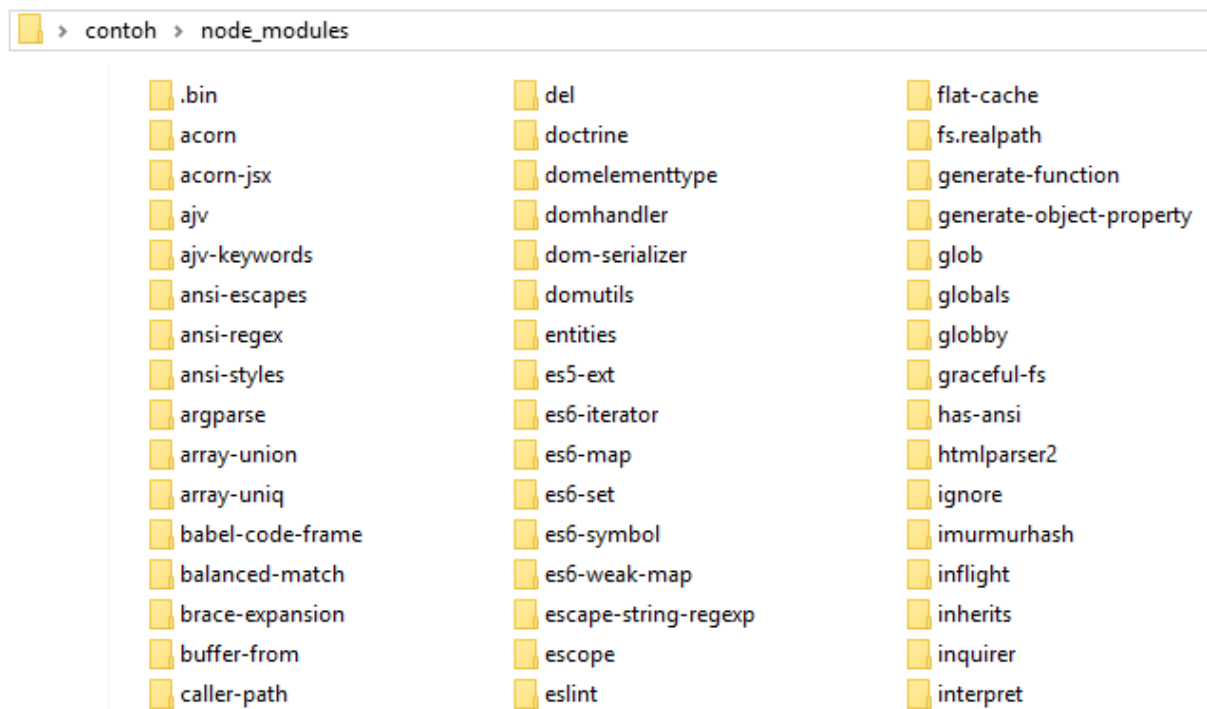
Folder images berisi semua gambar yang digunakan pada polymer.



Gambar 6.3 Directory images

3. Folder node_modules

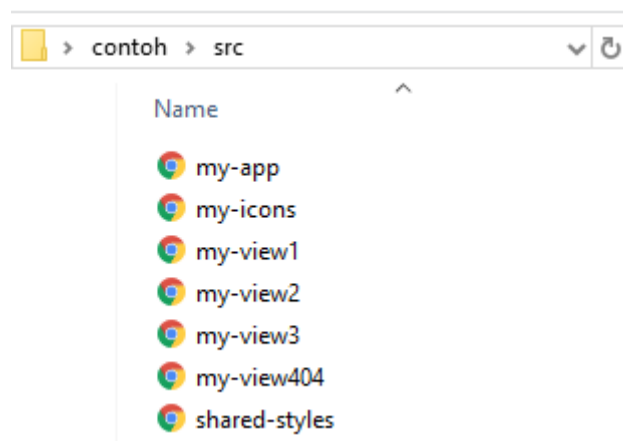
Folder node_modules adalah directory untuk module node yang telah diinstall melalui npm.



Gambar 6.4 Directory node_modules

4. Folder src

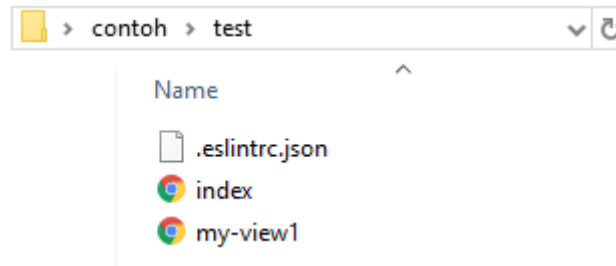
Dalam folder src inilah aplikasi yang akan buat nanti berada. Semua halaman web yang dibuat berada di sini.



Gambar 6.5 Directory src

5. Folder test

Secara default polymer sudah menyediakan library-library yang dibutuhkan untuk dapat melakukan unit testing. Direktori ini berfungsi untuk menyimpan seluruh file test yang dibuat untuk kemudian dijalankan.



Gambar 6.6 Directory test

6. File lainnya

a. Bower.json

Bower.json membantu Bower memahami proyek yang dibuat. Bower merekam meta data seperti nama proyek, versi, pencipta, ijin, dependensi, dan lain-lain. Sebelumnya, file ini diberi nama component.json, tapi kemudian diubah menjadi bower.json agar tidak terjadi konflik dengan alat lain yang menggunakan nama file yang sama.

b. Index.html

Index.html adalah sebuah file yang menjadi indeks sebuah website. File ini otomatis dibuka saat kita membuka website dengan memanggil nama domain website. Index html adalah default file dan merupakan file paling penting bagi website. Tanpa file index.html, jika kita membuka website maka browser akan menampilkan file-file lain dalam bentuk tampilan folder bukan tampilan website sesuai design.

c. Manifest.json

Manifest.json adalah JSON sederhana yang memberikan kemampuan untuk mengontrol bagaimana aplikasi terlihat oleh pengguna sesuai dengan platform yang digunakan (misalnya layar beranda perangkat seluler dan dekstop), mengarahkan apa yang bisa diluncurkan pengguna, dan menentukan tampilannya pada saat peluncuran.

d. Package.json

File ini digunakan oleh npm untuk mengunduh dan memasang library tambahan yang kita butuhkan dalam aplikasi yang akan kita buat.

e. Polymer.json

File ini digunakan untuk menyimpan informasi tentang struktur proyek, dan konfigurasi pembangunan yang diinginkan. Ini digunakan oleh Polimer CLI sebagai cara untuk memahami struktur aplikasi web.

5.2 Layout Polymer

Polymer memungkinkan programmer untuk mengimpor layout pada komponen yang terdapat pada file .html yang terpisah. Contoh sederhananya, misal programmer punya banyak aplikasi web yang dibangun dengan framework yang berbeda-beda. Dengan satu codebase HTML yang sama programmer bisa membuat sebuah footer menjadi component, sehingga footer tersebut dapat dipasang di semua aplikasi web kita menggunakan HTML Imports.

Sebagai contoh mengimpor komponen footer :

```
<link rel="import" href="my-footer.html">
```

Kemudian memanggilnya dengan Custom Element yang sudah kita definisikan.

```
<my-footer></my-footer>
```

Tidak hanya footer, component bisa saja berupa datepicker, login modal, ads banner, custom button bahkan proses autentifikasi single sign on. Banyak hal, baik komponen visual maupun non-visual. Setiap component tidak harus memiliki UI karena sebuah template umumnya terdiri dari blok HTML, CSS dan JS. Bisa saja isinya hanya berupa blok javascript.

Polymer juga dapat melakukan impor komponen secara banyak secara bersamaan.

```
<link rel="import" href="../../bower_components/polymer/polymer-element.html">
<link rel="import" href="../../bower_components/app-layout/app-drawer/app-drawer.html">
<link rel="import" href="../../bower_components/app-layout/app-drawer-layout/app-drawer-layout.html">
<link rel="import" href="../../bower_components/app-layout/app-header/app-header.html">
<link rel="import" href="../../bower_components/app-layout/app-header-layout/app-header-layout.html">
<link rel="import" href="../../bower_components/app-layout/app-scroll-effects/app-scroll-effects.html">
<link rel="import" href="../../bower_components/app-layout/app-toolbar/app-toolbar.html">
<link rel="import" href="../../bower_components/app-route/app-location.html">
<link rel="import" href="../../bower_components/app-route/app-route.html">
<link rel="import" href="../../bower_components/iron-pages/iron-pages.html">
<link rel="import" href="../../bower_components/iron-selector/iron-selector.html">
<link rel="import" href="../../bower_components/paper-icon-button/paper-icon-button.html">
<link rel="import" href="my-icons.html">
```

Gambar 6.7 Impor komponen pada polymer

5.2.1 Fitur pada webcomponents Polymer

Web components itu terdiri dari 4 fitur, yaitu:

1. *HTML Imports*, untuk menggunakan ulang dokumen HTML di dokumen HTML yang lain.
2. *Custom Element*, untuk membuat tag HTML kita sendiri.
3. *HTML Template*, untuk menyimpan subkonten yang dirender saat runtime. Terdiri dari Blok HTML, CSS dan JS.
4. *Shadow DOM*, untuk mengenkapsulasi DOM (termasuk style / css) sebuah *component*.

Contoh kode sederhana untuk membuat sebuah component.

Contoh index.html :

```
<html>
<head>
  <meta name="description" content="Web Components Sample" />
  <meta charset="utf-8">
  <title>Web Components Sample</title>
  <!-- 1. HTML Import -->
  <link id="my-footer-link" rel="import" href="my-footer.html">
</head>
<body>
  <my-footer></my-footer>
</body>
</html>
```

Gambar 6.8 Index.html pada webcomponent

Melakukan impor untuk file my-footer.html lalu memanggilnya dalam tag `<body></body>`.

Contoh my-footer.html

```
<template id="my-footer">
  <style>
    .container { text-align: center; }
  </style>
  <div class="container">
    
  </div>
</template>
<script>
  // Buat class MyFooter -> nama harus camelCase
  class MyFooter extends HTMLElement {
    constructor() {
      // Wajib panggil super di constructor
      super()
      // Ambil template element
      const template =
        document.querySelector('#my-footer-link')
        .import.querySelector('#my-footer')
      // Tambah node template ke dalam shadow root
      this.attachShadow({mode: 'open'})
      .appendChild(
        document.importNode(template.content, true)
      )
    }
  }
  // Custom Element harus di register terlebih dahulu
  window.customElements.define('my-footer', MyFooter)
</script>
```

Gambar 6.8 my-footer.html

Template my-footer.html menyimpan subkonten yang dirender saat runtime yang terdiri dari Blok HTML, CSS dan JS.

1. HTML Import

Yang perlu diperhatikan pada kode diatas adalah penggunaan attribute `id="my-footer-link"` pada baris `<link id="my-footer-link" rel="import" href="my-footer.html">` untuk mengidentifikasi DOM. Karena pengambilan *template* membutuhkan fungsi `import` pada node tersebut. Lihat `document.querySelector('#my-footer-link').import`.

2. Custom Element

Spesifikasi terbaru saat ini adalah Custom Elements v1. Dimana definisi custom elemen berupa class yang mengekstensi elemen lain `class MyFooter extends HTMLElement` dan di register dengan format `window.customElements.define([nama elemen], [nama class])`. Dengan aturan penamaan [nama elemen] harus kebab-case dan [nama class] harus CamelCase.

3. HTML Template

Blok HTML dan CSS yang berada di dalam tag `<template></template>` merupakan HTML Template yang kemudian akan dienkapsulasi di dalam Shadow DOM.

4. Shadow DOM

Bagian selanjutnya adalah tentang Shadow DOM. Kuncinya ada pada kode `this.attachShadow({mode: 'open'}).This` pada baris tersebut adalah node dari elemen `<my-footer></my-footer>`. Dengan perintah `attachShadow` maka shadow root akan di-inject di dalam node `my-footer`. Kemudian baris ini `.appendChild(document.importNode(template.content, true))` akan menambahkan semua DOM (blok HTML dan CSS) yang ada pada tag `<template></template>` berada di bawah Shadow DOM milik `my-footer`.

Kelebihan menggunakan Shadow DOM adalah enkapsulasi. Artinya semua behaviour (termasuk `<style></style>` yang berisi CSS) milik node yang berada di dalam Shadow Root tidak akan berpengaruh ke luar node lain.