

## SKRIPSI

# PEMBUATAN ULANG APLIKASI WSDC 2017 BALI DENGAN IONIC 5



Rajasa Cikal Maulana Solihin

NPM: 2017730084

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2021



# DAFTAR ISI

<b>DAFTAR ISI</b>	<b>iii</b>
<b>DAFTAR GAMBAR</b>	<b>v</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	3
1.6 Sistematika Pembahasan . . . . .	3
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 WSDC 2017 Bali . . . . .	5
2.2 Angular . . . . .	6
2.3 Ionic Framework . . . . .	8
2.3.1 Native API . . . . .	9
2.3.2 UI Component . . . . .	10
2.3.3 Migrasi Ionic 3 ke Ionic 5 . . . . .	15
<b>3 ANALISIS</b>	<b>23</b>
3.1 Analisis Sistem Kini . . . . .	23
3.1.1 Skenario Pengguna . . . . .	24
3.1.2 Struktur Ionic 3 . . . . .	26
3.2 Analisis Sistem Usulan . . . . .	44
3.2.1 Analisis Kebutuhan . . . . .	44
3.2.2 Permasalahan Pengembangan Sistem Usulan . . . . .	49
<b>DAFTAR REFERENSI</b>	<b>51</b>



## **DAFTAR GAMBAR**

2.1 Aplikasi WSDC 2017 Bali Saat Ini pada Perangkat Android . . . . .	5
2.2 Aplikasi WSDC 2017 Bali Saat Ini pada Perangkat Android . . . . .	6
3.1 <i>Use Case Diagram</i> Aplikasi WSDC 2017 Bali . . . . .	24
3.2 Wireframe Aplikasi WSDC 2017 Bali . . . . .	28
3.3 Wireframe Aplikasi WSDC 2017 Bali . . . . .	35



<sup>1</sup>

## BAB 1

<sup>2</sup>

### PENDAHULUAN

#### <sup>3</sup> 1.1 Latar Belakang

<sup>4</sup> Dalam pengembangan suatu aplikasi seluler, setiap vendor perangkat lunak memiliki alat-alat  
<sup>5</sup> yang unik yang hanya dimiliki oleh vendor tersebut, yaitu *Software Development Kit* (SDK) [1].  
<sup>6</sup> Setiap sistem operasi memiliki perangkat *native* masing-masing, yang digunakan sebagai alat untuk  
<sup>7</sup> memungkinkan pengembang dalam pengembangan aplikasi seluler pihak ketiga. Namun karena  
<sup>8</sup> SDK yang digunakan berbeda-beda tergantung kepada jenis sistem operasi perangkat seluler, maka  
<sup>9</sup> terdapat permasalahan, yaitu tidak dapat membuat aplikasi untuk platform yang berbeda, namun  
<sup>10</sup> dengan baris kode yang sama. Untuk mengatasi masalah tersebut, terdapat sebuah teknologi  
<sup>11</sup> yang dikembangkan oleh Adobe yaitu Apache Cordova. Apache Cordova memungkinkan aplikasi  
<sup>12</sup> berbasis web seperti HTML, CSS, dan Javascript, dikemas sebagai aplikasi seluler *native*, dan juga  
<sup>13</sup> menyediakan akses ke fitur perangkat.

<sup>14</sup> Aplikasi yang dibangun menggunakan Cordova disebut sebagai aplikasi *hybrid*, sebuah aplikasi  
<sup>15</sup> berbasis web yang berjalan di dalam sebuah tempat dari aplikasi *native* [1]. Aplikasi *hybrid* yang  
<sup>16</sup> berjalan merupakan aplikasi *native* yang menjalankan aplikasi berbasis web di tampilan web ponsel.  
<sup>17</sup> Aplikasi tersebut bersikap dan berperilaku normal seperti aplikasi pada umumnya, dan juga memiliki  
<sup>18</sup> keseluruhan akses terhadap perangkat.

<sup>19</sup> Salah satu kerangka kerja yang dapat digunakan untuk mengembangkan aplikasi *hybrid* adalah  
<sup>20</sup> Ionic Framework. Ionic Framework merupakan sebuah kerangka kerja *open source* lintas platform  
<sup>21</sup> yang memungkinkan untuk mengembangkan aplikasi *hybrid* yang bekerja pada berbagai macam  
<sup>22</sup> platform seluler seperti *android*, iOS, dan Windows [2]. Ionic memiliki berbagai macam *front-end*  
<sup>23</sup> *library* dan *User Interface(UI)* *Components* yang digunakan untuk perancangan aplikasi menggu-  
<sup>24</sup> nakan teknologi web seperti HTML, *Cascading Style Sheets* (CSS), dan Javascript. Dengan Ionic  
<sup>25</sup> Framework, memungkinkan sebuah aplikasi yang dapat berjalan menggunakan fitur *native* dari  
<sup>26</sup> sebuah perangkat.

<sup>27</sup> Pada Ionic 5, terdapat beberapa kerangka Javascript yang dapat diimplementasikan menggu-  
<sup>28</sup> nakan *framework* Ionic, seperti Angular, React, dan Vue. Angular pada awalnya diciptakan oleh  
<sup>29</sup> karyawan Google, Misko Hevert dan Adam Abrons pada tahun 2008, yang masih bernama AngularJS  
<sup>30</sup> dan dikembangkan dalam JavaScript [3]. Pada saat itu sebagian besar situs web menggunakan  
<sup>31</sup> aplikasi multi-halaman, yaitu ketika pengguna mengklik tautan, maka browser harus mengambil  
<sup>32</sup> dokumen HTML yang diminta dari server. Namun dengan Angular, digunakannya *Single-page*  
<sup>33</sup> *Application* (SPA), yaitu ketika halaman awal dimuat, semua yang dibutuhkan untuk membuat dan  
<sup>34</sup> menampilkan sebuah halaman diunduh, kemudian ditampilkan kedalam layar. Dengan begitu, bro-

1 wser tidak perlu melakukan *refresh* terhadap halaman tersebut [4]. React adalah *library* JavaScript  
2 *open source* untuk membangun antarmuka pengguna, dikelola oleh Facebook, dapat digunakan dalam  
3 berbagai skenario termasuk aplikasi iOS dan Android [3]. Sedangkan Vue merupakan *framework*  
4 progresif untuk membangun antarmuka pengguna untuk web, yang dapat digunakan baik untuk  
5 projek kecil dan untuk *Single-Page Applications* (SPAs) [3].

6 *World Schools Debating Championships* (WSDC) merupakan sebuah turnamen debat bahasa  
7 inggris tahunan untuk tim-tim tingkat sekolah menengah yang mewakili berbagai negara [5]. Pada  
8 awalnya, kompetisi universitas dunia akan diselenggarakan di Sydney pada bulan Juli 1988. Anggota  
9 Federasi Debat Australia menyadari bahwa tidak ada acara serupa untuk siswa sekolah menengah.  
10 Namun kejuaraan universitas dunia ini menunjukkan potensi yang sangat besar untuk kompetisi  
11 debat internasional yang melibatkan siswa dari seluruh dunia. Pada tahun 1991, kejuaraan diadakan  
12 di Edinburgh. Dan sejak saat itu nama World Schools Debating Championships digunakan dan  
13 berlangsung hingga saat ini.

14 WSDC yang diselenggarakan di Bali, Indonesia pada tahun 2017 memiliki sebuah aplikasi  
15 bernama WSDC 2017 Bali yang dikembangkan oleh PT DNArtworks menggunakan *framework*  
16 Ionic 3 untuk menunjang acara tersebut. Terdapat beberapa fungsi penting di dalam aplikasi  
17 ini, diantaranya adalah jadwal untuk kegiatan peserta, berita tentang acara WSDC yang sedang  
18 berlangsung, pemberitahuan mengenai kegiatan acara kepada peserta, informasi lokasi dan penunjuk  
19 arah ke lokasi kegiatan acara yang sedang berlangsung, dan notifikasi untuk peserta.

20 Aplikasi WSDC 2017 Bali yang dibangun pada tahun 2017 oleh PT DNArtworks menggunakan  
21 Ionic versi 3. Sedangkan Ionic versi 3 saat ini sudah tidak mendapat pembaruan lagi. Saat ini  
22 Ionic semakin berkembang dan sudah mencapai Ionic versi 5. Maka dari itu, pada skripsi ini akan  
23 dibuat sebuah aplikasi pembaruan dari aplikasi WSDC 2017 Bali saat ini, dengan menggunakan  
24 *framework* Ionic versi 5. *Framework* yang lebih baru memungkinkan perawatan yang lebih efisien,  
25 serta dukungan teknologi yang lebih terbarukan.

## 26 **1.2 Rumusan Masalah**

27 Rumusan masalah yang akan dibahas pada skripsi ini adalah bagaimana melakukan migrasi aplikasi  
28 *android* WSDC ke *framework* Ionic versi 5?

## 29 **1.3 Tujuan**

30 Tujuan yang ingin dicapai dari penulisan skripsi ini adalah melakukan migrasi aplikasi *android*  
31 WSDC ke *framework* Ionic versi 5.

## 32 **1.4 Batasan Masalah**

33 Dalam skripsi ini dibuat batasan-batasan masalah dalam pembuatan perangkat lunak. Batasan-  
34 batasan masalah yang ditetapkan adalah sebagai berikut:

- 35 1. Aplikasi ini tidak akan memiliki fitur notifikasi, karena acara WSDC 2017 Bali sudah selesai  
36 dan tidak diperlukan kembali fitur notifikasi.
- 37 2. Aplikasi hanya akan berjalan pada *platform mobile* berbasis android.

## 1 1.5 Metodologi

- 2 Langkah-langkah yang dilakukan dalam skripsi ini adalah sebagai berikut:
- 3 1. Melakukan studi mengenai *framework* Ionic versi 3 dan versi 5.
- 4 2. Menganalisis aplikasi WSDC 2017 Bali.
- 5 3. Mempelajari bagaimana cara melakukan migrasi Ionic versi 3 ke versi 5.
- 6 4. Mendesain kelas aplikasi.
- 7 5. Membangun aplikasi WSDC dengan *framework* Ionic versi 5.
- 8 6. Melakukan pengujian dan eksperimen.
- 9 7. Menulis dokumen skripsi.

## 10 1.6 Sistematika Pembahasan

11 Sistematika penulisan setiap bab pada skripsi ini adalah sebagai berikut:

- 12 1. Bab Pendahuluan  
Bab 1 berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan yang digunakan untuk menyusun skripsi ini.
- 13 2. Bab Dasar Teori  
Bab 2 berisi teori-teori yang digunakan dalam pembuatan skripsi ini. Teori-teori tersebut  
yaitu WSDC, Angular, Ionic, Capacitor, Cordova, UI Components, dan Migrasi Ionic.
- 14 3. Bab Analisis  
Bab 3 berisi analisis yang dilakukan pada skripsi ini, meliputi analisis sistem kini, analisis  
kebutuhan aplikasi WSDC 2017 Bali yang akan dibangun, serta permasalahan pembangunan  
sistem usulan.
- 15 4. Bab Perancangan  
Bab 4 berisi perancangan aplikasi.
- 16 5. Bab Implementasi dan Pengujian  
Bab 5 berisi implementasi dan pengujian aplikasi.
- 17 6. Bab Kesimpulan dan Saran  
Bab 6 berisi kesimpulan dari hasil pembangunan aplikasi beserta  
saran untuk pengembangan selanjutnya.



1

## BAB 2

2

### LANDASAN TEORI

- 3 Pada bab ini akan menjelaskan dasar-dasar teori mengenai Ionic, berikut dengan cara untuk  
4 melakukan migrasi dari Ionic 3 ke Ionic 5. Akan dibahas pula aplikasi WSDC 2017 Bali saat ini,  
5 Native API berupa Capacitor dan Cordova, dan UI Components.

#### 6 2.1 WSDC 2017 Bali

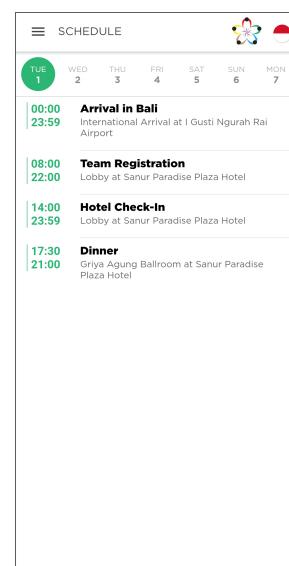
- 7 Aplikasi WSDC 2017 Bali digunakan untuk menunjang keberlangsungan acara WSDC 2017 yang dise-  
8 lenggarakan di Bali, Indonesia. Aplikasi WSDC 2017 Bali dapat diunduh untuk sistem operasi *andro-*  
9 *id* melalui URL <https://play.google.com/store/apps/details?id=org.wsdc2017indonesia.app>. Aplikasi ini dibangun dan dikembangkan oleh PT DNArtworks Komunikasi Visual yang rilis  
10 di Play Store pada tanggal 30 Juli 2017, dengan versi terakhir adalah versi 1.1.2 yang rilis pada 1  
11 Agustus 2017. Selain rilis pada perangkat *android*, aplikasi ini juga rilis untuk perangkat bergerak  
12 berbasis sistem operasi iOS. Namun saat ini aplikasi tersebut sudah diturunkan dari App Store  
13 pada perangkat berbasis sistem opearsi iOS. Untuk membuka dan memakai aplikasi WSDC 2017  
14 Bali saat ini, pengguna tidak diperlukan *login* agar dapat mengakses seluruh fitur yang tersedia.  
15 Lalu, untuk kepentingan skripsi ini, peneliti memiliki akses ke dalam kode program aplikasi WSDC  
16 2017 Bali.  
17



(a) Halaman Utama



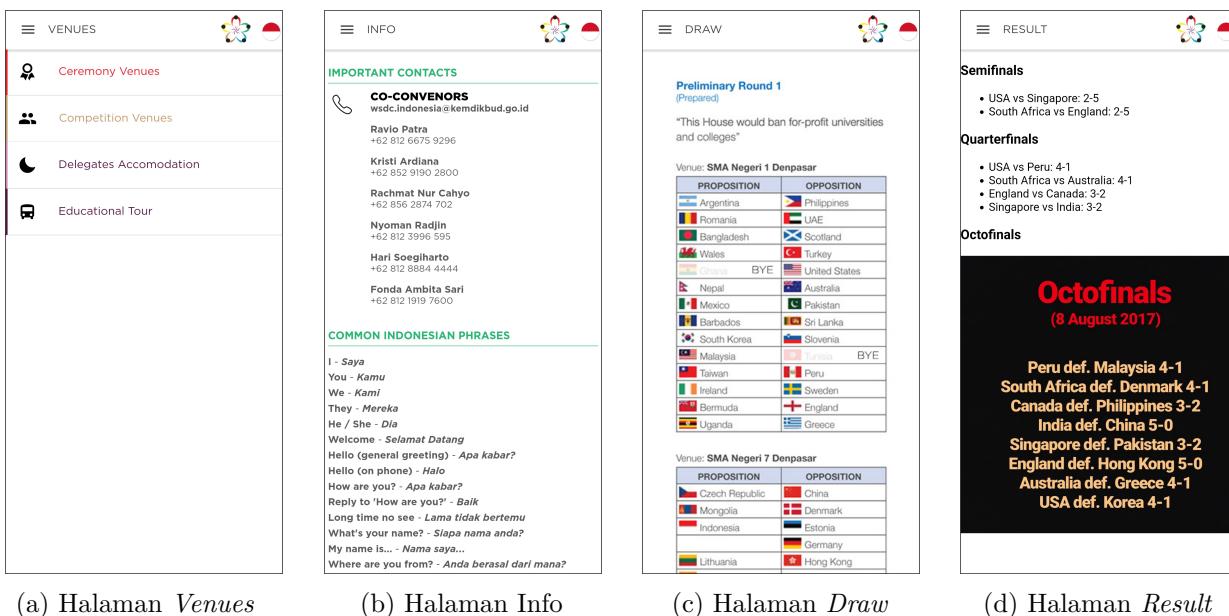
(b) Halaman *Announcements*



(c) Halaman *Schedule*

Gambar 2.1: Aplikasi WSDC 2017 Bali Saat Ini pada Perangkat Android

- Fitur-fitur yang terdapat di aplikasi WSDC 2017 Bali saat ini yaitu :
1. *Newsletter*: Pengguna dapat mengunduh dan berita-berita terkait acara WSDC 2017 Bali (Gambar 2.1a).
  2. *Announcements*: Pengguna dapat melihat pemberitahuan tentang berjalannya acara WSDC 2017 Bali (Gambar 2.1b).
  3. *Schedule*: Pengguna dapat melihat jadwal acara WSDC 2017 Bali yang sudah diadakan (Gambar 2.1c).
  4. *Venues*: Pengguna dapat melihat berbagai macam lokasi acara WSDC 2017 Bali, mulai dari lokasi upacara, lokasi kompetisi, dan lokasi wisata edukasi. Masing-masing dari lokasi tersebut dapat menunjukkan arah dan jarak dari lokasi tempat pengguna berada (Gambar 2.2a).
  5. *Info*: Pengguna dapat melihat informasi terkait dengan tim pengembang dari aplikasi WSDC 2017 Bali, kontak-kontak penting yang dapat dihubungi, dan kosa kata penting dalam Bahasa Indonesia (Gambar 2.2b).
  6. *Draw*: Pengguna dapat melihat melihat pembagian *venue* dan kubu proposisi atau oposisi dari hasil pengundian untuk para negara peserta WSDC 2017 Bali (Gambar 2.2c).
  7. *Result*: Pengguna dapat melihat informasi terkait hasil dari pertandingan pada semi final, perempat final, dan perdelapan final WSDC 2017 Bali (Gambar 2.2d).



Gambar 2.2: Aplikasi WSDC 2017 Bali Saat Ini pada Perangkat Android

## 2.2 Angular

Angular merupakan kerangka kerja Javascript terbuka, yang dikembangkan oleh Google, dan merupakan penerus dari versi sebelumnya yaitu AngularJS [6]. Angular versi pertama dirilis pada tahun 2016 dengan nama Angular 2. Aplikasi Angular dapat dibangun dengan menggunakan JavaScript, atau TypeScript.

Angular dapat digunakan untuk membuat aplikasi Single-page-application (SPA), yaitu ketika halaman awal dimuat, semua yang dibutuhkan untuk membuat dan menampilkan sebuah halaman

1 diunduh, kemudian ditampilkan kedalam layar [4]. Tampilan pada awalnya merupakan halaman  
2 dengan kode-kode HTML yang belum utuh, kemudian tampilan halaman dibuat dan diunduh  
3 untuk ditampilkan. Dengan begitu, *browser* tidak perlu melakukan penyegaran tampilan lagi untuk  
4 menampilkan halaman baru.

5 Angular dapat dibangun dengan menggunakan JavaScript atau TypeScript. Namun, penggunaan  
6 TypeScript saat ini menjadi lebih produktif dibandingkan dengan JavaScript [6]. TypeScript  
7 mengikuti perkembangan terakhir dari ECMAScript, dan menambahkan *types*, *interface*, *decorators*,  
8 *class member variables*, *generic*, *enum*, dan *keyword* seperti *public*, *protected*, dan *private*. Lalu  
9 selain itu, TypeScript juga dapat dimungkinkan untuk mendeklarasi jenis tipe khusus yang dapat  
10 di kustomisasi. Maka dari itu, Framework Angular sendiri ditulis menggunakan TypeScript.

11 Kelas TypeScript akan digunakan oleh Angular di dalam sebuah komponen. Komponen pada  
12 Angular merupakan sebuah penyusun utama untuk aplikasi Angular. Setiap komponen yang ada  
13 pada aplikasi terdiri dari *template* HTML untuk mendeklarasi halaman yang akan dimuat, kelas  
14 TypeScript, serta CSS Selector.

15 Di dalam aplikasi Angular, terdapat beberapa komponen, seperti Parent Component, dan Child  
16 Component. Parent Component dapat mengirimkan data ke properti Child Component-nya. Namun,  
17 Child Component tidak tahu siapa yang mengirimkannya. Sedangkan Child Component juga dapat  
18 mengirimkan data ke Parent Component, meskipun dia tidak tahu siapa Parent Component-nya.  
19 Arsitektur seperti ini dapat membuat komponen menjadi mandiri dan dapat digunakan kembali.

20 Di dalam TypeScript, sebuah kelas komponen memiliki anotasi dengan sebuah *decorator* (Ko-  
21 de 2.1) yang ditempatkan sesuai dengan komponen UI nya berada. Komponen akan berisi *instance*  
22 dari service class yang diimplementasi dari *business logic* tanpa UI. Sebuah service class merupakan  
23 implementasi dari *business logic*. Angular akan menempatkan *services* ke dalam komponen atau ke  
24 dalam *services* lainnya dengan menggunakan *dependency injection* (DI).

Kode 2.1: Anotasi Komponen dengan *Decorator*

```
25 1 @Component({  
26 2   ...  
27 3 }  
28 4 export class AppComponent{  
29 5   ...  
30 6 })
```

33 Komponen-komponen yang ada akan dikelompokkan menjadi Angular modules, yaitu sebuah  
34 kelas yang diberi **@NgModule()**. Angular module biasanya merupakan sebuah kelas kecil yang  
35 tidak memiliki isi, kecuali menulis kode bootstrap secara manual ke dalam aplikasi. Contohnya,  
36 ketika sebuah aplikasi merupakan turunan dari AngularJS yang lama, *decorator* **@NgModule()**  
37 menampilkan semua komponen dan *artifacts* lain termasuk *services*, *directive*, dan yang lainnya,  
38 maka semua itu harus disertakan ke dalam module (Kode 2.2).

Kode 2.2: Module dengan Komponen

```

1 @NgModule({
2   declarations: [
3     AppComponent
4   ],
5   imports: [
6     BrowserModule
7   ],
8   bootstrap: [AppComponent]
9 })
10 export class AppModule{
11   ...
12 }
13

```

15 Komponen juga bisa terdapat HTML *template* yang berada di dalam komponen tersebut,  
 16 atau di dalam file yang berada di luar file komponen yang direferensikan dengan menggunakan  
 17 properti **templateUrl** di dalam komponen. File HTML *template* yang terpisah dengan komponen,  
 18 memberikan efek kode yang lebih bersih di dalam komponen nya. Selain HTML *template*, file lain  
 19 seperti file CSS dapat diletakan terpisah dari file komponen dengan menggunakan styleUrls untuk  
 20 mereferensikan file CSS ke dalam komponen (Kode 2.3). Lalu terdapat juga properti selector untuk  
 21 mendefinisikan nama dari tag yang bisa digunakan atau dipanggil oleh komponen lain.

Kode 2.3: Contoh Merefrensikan HTML dan CSS di Dalam Komponen

```

22 @Component({
23   selector: 'app-search',
24   templateUrl: './search.component.html',
25   styleUrls:[ './search.component.css' ]
26 })
27 export class SearchComponent{
28   ...
29 }
30

```

### 32 2.3 Ionic Framework

33 Ionic Framework merupakan sebuah kerangka kerja *open source* lintas platform yang memungkinkan  
 34 untuk mengembangkan aplikasi hibrida yang bekerja pada berbagai macam platform seluler seperti  
 35 android, iOS, dan Windows [2]. Ionic memiliki berbagai macam *front-end library* dan komponen  
 36 *User Interface*(UI) yang digunakan untuk perancangan aplikasi menggunakan teknologi web seperti  
 37 HTML, CSS, dan Javascript, dengan integrasi untuk berbagai *framework* seperti Angular, React,  
 38 dan Vue. Saat pertama kali dibuat, Ionic menggunakan AngularJS. Namun, seiring saat Angular  
 39 versi 2 yang menggunakan Typescript dirilis, Ionic versi 2 dan selanjutnya menggunakan Angular.  
 40 Lalu, pada tahun 2019, Ionic mendukung penggunaan *framework* lain selain Angular, yaitu React  
 41 dan Vue. Di dalam Ionic, Angular digunakan untuk membangun aplikasi dan perutean, sehingga  
 42 aplikasi dapat sejalan dengan ekosistem Angular lainnya. Ionic menyediakan *toolkit* Angular untuk  
 43 membangun aplikasi dan terintegrasi dengan Angular CLI resmi yang menyediakan fitur khusus  
 44 untuk aplikasi Ionic Angular. Pada saat skripsi ini dibuat, Ionic versi terbaru adalah Ionic versi 5,  
 45 sedangkan Angular yang digunakan adalah Angular versi 12.

### **2.3.1 Native API**

Native API memungkinkan pengembangan aplikasi langsung terintegrasi ke dalam platform. Pengembang dapat membuat aplikasi pada perangkat *mobile* untuk dapat diimplementasikan ke berbagai *platform*, seperti iOS dan Android, setelah pengembangan selesai di dalam *framework native* tanpa perubahan, dan tidak mempengaruhi performa dari aplikasi tersebut [7].

Ionic mendukung komunikasi dengan menggunakan Native API yang terintegrasi untuk menambahkan fungsionalitas ke dalam aplikasi Ionic apapun dengan menggunakan Capacitor atau Cordova. Dengan terpasangnya Ionic Native, maka aplikasi akan memiliki antar muka yang diperlukan untuk berinteraksi dengan salah satu *plug-in*, yaitu Capacitor atau Cordova.

#### **2.3.1.1 Capacitor**

Tujuan dari Capacitor adalah untuk menyediakan akses ke perangkat *native* dan fitur platform, serta untuk menyediakan satu set API untuk mengembangkan aplikasi seluler secara hibrida, *Progressive Web Apps* berbasis web, dan aplikasi komputer berbasis Electron [8]. Capacitor merupakan penerus dari Cordova, dengan tujuan untuk memungkinkan aplikasi web modern berjalan di semua platform utama. Capacitor juga mendapat dukungan terhadap banyak *plugin* Cordova.

#### **2.3.1.2 Cordova**

Cordova merupakan *framework open source* yang dapat membuat pengembang untuk menggunakan teknologi seperti HTML, JavaScript, dan CSS untuk membangun aplikasi untuk perangkat bergerak yang dapat berjalan pada beberapa sistem operasi *mobile* [9]. Cordova menyediakan antarmuka antara WebView dan lapisan *native* pada perangkat [7]. Selain dapat bekerja pada dua platform seluler Android dan iOS, Cordova juga dapat digunakan pada platform seluler seperti Windows Phone, Blackberry, dan FireOS.

Untuk mengonfigurasi proyek Cordova, saat ini dapat menggunakan *Command Line Tool* (CLI). CLI membuat proyek dasar dan mengonfigurasinya agar berfungsi dengan platform seluler apa pun yang didukung yang dapat digunakan. Cordova CLI juga dapat membuat pengembang memiliki integrasi dan pengelolaan *plug-in*. Selain itu, CLI juga dapat mengkompilasi aplikasi untuk berjalan pada simulator atau pada perangkat *native*. Serupa dengan Capacitor, Cordova membuat pengembang dapat mengakses fitur *native* dari sebuah perangkat, seperti kamera, papan ketik, dan geolokasi, menggunakan *plugin*. Framework Ionic telah terdapat berbagai macam TypeScript *wrapper* untuk *plugins* Cordova. Untuk dapat menggunakan Cordova Plugins, yaitu dengan memasang Cordova Plugins terlebih dahulu (Kode 2.4), dan memperbaruiinya ke versi terakhir (Kode 2.5) yang dapat dilakukan melalui CLI. Setiap *plugins* memiliki dua komponen, yaitu kode *native* (Cordova), dan kode TypeScript (Ionic Native). Cordova Plugins juga dibungkus di dalam Promise atau Observable untuk menyediakan antarmuka *plug-in*.

#### Kode 2.4: Kode untuk Memasang Cordova Plugins

```
1 21 npm install cordova-plugin-name
2 22 npx cap sync
```

#### Kode 2.5: Kode untuk Memperbarui Cordova Plugins

```
5 61 npm install cordova-plugin-name@2
8 2 npx cap update
```

### 9 2.3.2 UI Component

10 Framework Ionic menggunakan kemampuan Angular dalam memperluas kosakata HTML, yaitu  
 11 menyertakan *tag* khusus untuk menciptakan seluruh rangkaian komponen [7]. Semua komponen  
 12 memiliki awalan ion, sehingga dapat dikenali dalam markup. Sama seperti *tag* HTML standar,  
 13 komponen Ionic juga dapat menerima berbagai macam atribut sebagai pengaturan dari *tag* tersebut,  
 14 seperti mengatur id atau mendefinisikan kelas CSS tambahan. Terdapat beberapa komponen yang  
 15 ada pada *framework* Ionic yaitu :

- Action Sheet

17 Merupakan dialog yang menampilkan serangkaian opsi, yang muncul di atas konten aplikasi  
 18 dan harus ditutup secara manual oleh pengguna sebelum pengguna dapat melanjutkan interaksi  
 19 dengan aplikasi. Untuk menutup Action Sheet terdapat beberapa cara, termasuk mengetuk  
 20 latar belakang atau menekan tombol escape di desktop.

- Alert

22 Alert merupakan dialog yang menampilkan informasi kepada pengguna, atau mengumpulkan  
 23 informasi dari pengguna menggunakan input. Alert muncul di atas konten aplikasi, dan harus  
 24 ditutup secara manual oleh pengguna sebelum pengguna dapat melanjutkan interaksi dengan  
 25 aplikasi. Secara opsional, terdapat header, sub header, dan pesan yang ada pada Alert.

- Badge

27 Merupakan elemen *inline block* yang biasanya muncul di dekat elemen lain, berisi angka atau  
 28 karakter lain, yang digunakan sebagai pemberitahuan bahwa ada item tambahan yang terkait  
 29 dengan suatu elemen dan menunjukkan berapa banyak item yang ada. Penggunaan Badge  
 30 dengan menggunakan *tag* <ion-badge> (Kode 2.6).

#### Kode 2.6: Potongan Kode Program dari Badge Component

```
31 32 <ion-badge>99</ion-badge>
```

- Button

35 Merupakan elemen yang dapat diklik, biasanya digunakan dalam formulir atau di mana pun  
 36 yang membutuhkan fungsionalitas tombol. Button biasanya menampilkan teks, ikon, atau  
 37 bisa juga keduanya. Button dapat pula menggunakan atribut untuk menampilkannya dengan  
 38 penampilan tertentu. Penggunaan Button dengan menggunakan *tag* <ion-button> (Kode 2.7).

#### Kode 2.7: Potongan Kode Program dari Button Component

```
39 40 <ion-button>Default</ion-button>
```

- Card

Merupakan bagian standar dari tampilan antarmuka yang berfungsi sebagai titik masuk ke dalam informasi yang lebih detail. Card dapat menjadi satu komponen, tetapi sering kali terdiri dari beberapa header, judul, sub judul, dan konten. Penggunaan Card dengan menggunakan tag `<ion-card>` yang dapat berisi *header*, *subtitle*, *title*, dan *content* (Kode 2.8).

Kode 2.8: Potongan Kode Program dari Card Component

```

6 1 <ion-card>
7 2   <ion-card-header>
8 3     <ion-card-subtitle>Card Subtitle</ion-card-subtitle>
9 4     <ion-card-title>Card Title</ion-card-title>
10 5   </ion-card-header>
11 6
12 7   <ion-card-content>
13 8     Card Content
14 9   </ion-card-content>
15 10 </ion-card>
16

```

- Content

Komponen content merupakan penyedia area konten yang bisa digunakan untuk mengontrol area yang dapat digulir. Dalam satu tampilan, setidaknya terdapat satu buah content. Content juga dapat dimodifikasi padding, margin, dan lainnya menggunakan *global style* yang berada di CSS Utilites atau mengubahnya secara individual dengan menggunakan CSS. Penggunaan Content dengan menggunakan tag `<ion-content>` (Kode 2.9).

Kode 2.9: Potongan Kode Program dari Content Component

```

24 1 <ion-content
25 2   [scrollEvents]="true"
26 3   (ionScrollStart)="logScrollStart()"
27 4   (ionScroll)="logScrolling($event)"
28 5   (ionScrollEnd)="logScrollEnd()">
29 6     <h1>Main Content</h1>
30 7
31 8     <div slot="fixed">
32 9       <h1>Fixed Content</h1>
33 10    </div>
34 11 </ion-content>
35

```

- Date and Time Pickers

Datetime merupakan penampil antarmuka untuk pengguna memilih tanggal dan waktu. Terdapat kolom yang dapat digulir yang dapat digunakan untuk memilih tahun, bulan, hari, jam, dan menit secara individual. Komponen ini menampilkan nilai di dua tempat, yaitu di komponen `<ion-datetime>` (Kode 2.21), dan di antarmuka pemilih yang ditampilkan dari bawah layar.

Kode 2.10: Kode Program dari Datetime Component dengan Format Bulan-Hari-Tahun

```

43 1 <ion-datetime displayFormat="MM_DD_YY" placeholder="Select Date"></ion-datetime>
44

```

- Infinite Scroll

Komponen Infinite Scroll memanggil sebuah action yang akan dilakukan ketika pengguna menggulir dengan jarak tertentu dari bawah atau atas halaman. Penggunaan Infinite Scroll dengan menggunakan tag `<ion-infinite-scroll>` (Kode 2.11).

Kode 2.11: Potongan Kode Program dari Infinite Scroll Component

```
1 <ion-infinite-scroll threshold="100px" (ionInfinite)="loadData($event)">
2   <ion-infinite-scroll-content
3     loadingSpinner="bubbles"
4     loadingText="Loading more data...">
5   </ion-infinite-scroll-content>
6 </ion-infinite-scroll>
```

- Icon

Icon merupakan komponen yang berupa gambar kecil, yang merepresentasikan sebuah berkas, dan folder di dalam aplikasi. Penggunaan Icon adalah dengan menggunakan tag `<ion-icon>` (Kode 2.12).

Kode 2.12: Potongan Kode Program dari Icon Home

```
13
14 1 <ion-icon name="home"></ion-icon>
```

- Item

Item merupakan elemen yang dapat berisi teks, ikon, avatar, gambar, masukan, dan elemen asli atau kustom lainnya. Biasanya, item ditempatkan di dalam sebuah *list* bersamaan dengan item lainnya dengan tag `<ion-item>` (Kode 2.13). Dapat dilakukan *swipe*, dihapus, disusun ulang, dan diedit.

Kode 2.13: Potongan Kode Program dari Item Component

```
21      1 <ion-item>
22          2   <ion-label>
23              3     Item
24          4   </ion-label>
25      5 </ion-item>
```

- List

Komponen List terdiri dari beberapa baris *item* yang dapat berisi teks, tombol, *toggles*, ikon, thumbnails, dan komponen-komponen lainnya menggunakan tag `<ion-list>` (Kode 2.14). List mendukung untuk pengguna melakukan *swiping*, *dragging*, dan penghapusan *item*.

Kode 2.14: Potongan Kode Program dari List Component

```
32      1 <ion-list>
33          2   <ion-item>
34              3     <ion-label>Pokemon Yellow</ion-label>
35          4   </ion-item>
36          5   <ion-item>
37              6     <ion-label>Mega Man X</ion-label>
38          7   </ion-item>
39          8   <ion-item>
40              9     <ion-label>The Legend of Zelda</ion-label>
41          10  </ion-item>
42          11  <ion-item>
43              12    <ion-label>Pac-Man</ion-label>
44          13  </ion-item>
45          14  <ion-item>
46              15    <ion-label>Super Mario World</ion-label>
47          16  </ion-item>
48
49      17 </ion-list>
```

1     • Menu

2       Komponen Menu merupakan panel navigasi samping yang dapat dilakukan *slides* dari sisi  
 3       pada tampilan halaman saat ini menggunakan tag <ion-menu> (Kode 2.15). Pada dasarnya,  
 4       Menu muncul dari kiri, tetapi sisi kemunculan menu dapat diganti.

Kode 2.15: Potongan Kode Program dari Menu Component

```
5   1<ion-menu side="start" menuId="first" contentId="main">
6   2   <ion-header>
7   3     <ion-toolbar color="primary">
8   4       <ion-title>Start Menu</ion-title>
9   5     </ion-toolbar>
10  6   </ion-header>
11  7   <ion-content>
12  8     <ion-list>
13  9       <ion-item>Menu Item</ion-item>
14 10       <ion-item>Menu Item</ion-item>
15 11       <ion-item>Menu Item</ion-item>
16 12       <ion-item>Menu Item</ion-item>
17 13       <ion-item>Menu Item</ion-item>
18 14       <ion-item>Menu Item</ion-item>
19 15     </ion-list>
20 16   </ion-content>
21 17 </ion-menu>
```

23     • Modal

24       Modal merupakan kotak dialog yang muncul diatas konten aplikasi lain, dan harus diutup  
 25       secara manual oleh pengguna sebelum pengguna dapat melanjutkan menggunakan aplikasi.  
 26       Modal berguna sebagai komponen pilihan ketika ada banyak opsi untuk dipilih, atau melakukan  
 27       penyaringan isi di dalam daftar, serta beberapa kasus serupa lainnya (Kode 2.16).

Kode 2.16: Kode Program dari Modal

```
28 1 import { Component, Input } from '@angular/core';
29 2
30 3 @Component({
31 4   selector: 'modal-page',
32 5 })
33 6 export class ModalPage {
34 7   constructor() {}
35 8 }
```

38     • Navigation

39       Navigation adalah komponen mandiri yang digunakan untuk membuat komponen baru  
 40       ke dalam *stack*. Navigation tidak terikat kepada *router* tertentu, mengakibatkan jika kita  
 41       membuat komponen Navigation dan melakukan *push* komponen lain ke dalam *stack*, komponen  
 42       tersebut tidak akan mempengaruhi router aplikasi secara keseluruhan. Sesuai dengan kasus  
 43       penggunaan dimana ketika pengguna bisa memilih modal, yang membutuhkan sub-navigasinya  
 44       sendiri, tanpa membuatnya terikat ke URL aplikasi.

45     • Segment

46       Segment berfungsi untuk menampilkan pilihan tombol bagi pengguna untuk beralih di antara  
 47       tampilan berbeda di dalam satu halaman yang sama. Segment menampilkan sekelompok  
 48       tombol-tombol yang dapat diklik, dalam baris horizontal. Penggunaan Segment dengan  
 49       menggunakan tag <ion-segment> (Kode 2.17).

Kode 2.17: Kode Program dari Segment

```

1 <ion-segment (ionChange)="segmentChanged($event)">
2   <ion-segment-button value="friends">
3     <ion-label>Friends</ion-label>
4   </ion-segment-button>
5   <ion-segment-button value="enemies">
6     <ion-label>Enemies</ion-label>
7   </ion-segment-button>
8 </ion-segment>

```

- Slides

Komponen Slides merupakan sebuah *multi-section container* yang setiap bagianya dapat dilakukan *swipe* atau *drag*. Untuk menggunakan komponen ini dengan menggunakan tag `<ion-slides>` (Kode 2.18).

Kode 2.18: Kode Program dari Slides

```

15 <ion-slides pager="true" [options]="slideOpts">
16   <ion-slide>
17     <h1>Slide 1</h1>
18   </ion-slide>
19   <ion-slide>
20     <h1>Slide 2</h1>
21   </ion-slide>
22   <ion-slide>
23     <h1>Slide 3</h1>
24   </ion-slide>
25 </ion-slides>
26

```

- Tabs

Tabs merupakan navigasi *top-level* yang mengimplementasi sebuah *tab-based navigation*. Tabs dapat digunakan dengan tag `<ion-tabs>` (Kode 2.19) yang tidak memiliki *styling* apapun dan bekerja sebagai *router outlet* untuk menangani navigasi.

Kode 2.19: Kode Program dari Tabs

```

32 <ion-tabs>
33   <ion-tab-bar slot="bottom">
34     <ion-tab-button tab="schedule">
35       <ion-icon name="calendar"></ion-icon>
36       <ion-label>Schedule</ion-label>
37       <ion-badge>6</ion-badge>
38     </ion-tab-button>
39
40     <ion-tab-button tab="speakers">
41       <ion-icon name="person-circle"></ion-icon>
42       <ion-label>Speakers</ion-label>
43     </ion-tab-button>
44   </ion-tab-bar>
45 </ion-tabs>
46

```

- Toast

Komponen Toast merupakan sebuah notifikasi yang digunakan di aplikasi modern untuk memberikan umpan balik atau menampilkan pesan sistem. Komponen ini muncul diatas konten aplikasi, dan dapat ditutup untuk melanjutkan penggunaan aplikasi. Komponen ini menggunakan `ToastController` yang dimiliki oleh *library* Ionic Angular (Kode 2.20)

Kode 2.20: Kode Program dari Toast

```

53
54   1 async presentToast() {
55   2     const toast = await this.toastController.create({
56   3       message: 'Your settings have been saved.',
57   4       duration: 2000
58   5     });
59   6     toast.present();
60   7   }

```

1     • Toolbar

2     Toolbar dapat diposisikan di atas ataupun di bawah konten. Ketika toolbar ditempatkan di  
 3     header <ion-header> akan muncul di bagian atas konten, sedangkan ketika ditempatkan di  
 4     footer <ion-footer> akan muncul tetap di bagian bawah. Toolbar menggunakan tag <ion-  
 5     toolbar>, yang di dalamnya dapat berisi button, dan dapat menggunakan border (Kode 2.21).

Kode 2.21: Kode Program dari Toolbar dengan Button di Dalamnya

```
6 1 <ion-toolbar>
7 2   <ion-buttons slot="start">
8 3     <ion-back-button></ion-back-button>
9 4   </ion-buttons>
10 5   <ion-title>Back Button</ion-title>
11 6 </ion-toolbar>
```

14 Selain komponen-komponen yang telah disebutkan, tertapat beberapa komponen lainnya yang tidak  
 15 disebutkan disini. Komponen-komponen tersebut yaitu Checkbox, Chip, Floating Action Button,  
 16 Grid, Icon, Input, Popover, Progress Indicator, Radio, Refresher, Reorder, Routing, Searchbar,  
 17 Select, dan Toggle <sup>1</sup>.

18 **2.3.3 Migrasi Ionic 3 ke Ionic 5**

19 Untuk melakukan migrasi dari Ionic 3 ke Ionic 5 memerlukan dua tahap, yaitu migrasi dari Ionic 3  
 20 ke Ionic 4, dan migrasi Ionic 4 ke Ionic 5. Tahapan migrasi tersebut adalah sebagai berikut:

21 1. Migrasi Ionic 3 ke Ionic 4

22     Ada beberapa langkah untuk melakukan migrasi dari Ionic 3 ke dalam Ionic 4, yaitu:

23     (a) Membuat Proyek Ionic Baru

24         Untuk membuat projek Ionic baru tanpa *template* apapun dengan menggunakan perintah  
 25         **ionic start myApp blank** dan memilih Angular sebagai *frameworknya* 2.22.

Kode 2.22: Perintah Membuat Proyek Ionic Baru

```
26 1 ionic start myApp blank
```

29     (b) Menyalin Angular Services

30         Menyalin Angular Services yang pada Ionic 3 berada di **src/providers**, menjadi **src/ap-**  
 31         **p/services** pada Ionic 4.

32     (c) Menyalin *Root-level Items*

33         Menyalin seluruh *Root-level Items* pada Ionic versi 3 dengan direktori yang sama atau  
 34         dengan beberapa perubahan. Terdapat beberapa perubahan baik itu nama maupun letak  
 35         dari suatu fungsi atau berkas. Perubahan tersebut yaitu:

36         i. *Imports*

37         Terjadi perubahan dalam mengimpor *package* di Ionic 3 dan Ionic 4. Daftar  
 38         perubahan tersebut adalah sebagai berikut :

39         A. Component Imports

40         Untuk kepentingan konsistensi dengan *framework* lain, maka untuk mengimpor  
 41         komponen Ionic diawali dengan ion 2.23.

---

<sup>1</sup> ‘UI Components’ <https://ionicframework.com/docs/components>, Diakses pada 17 April 2022.

#### Kode 2.23: Impor Komponen pada Ionic 4

```
1 import { IonInput, IonList, IonSlides } from '@ionic/angular';
```

#### B. Package Name

Terdapat perubahan pada Ionic 4, dimana nama *package* diubah menjadi @ionic/angular. Untuk dapat menggunakannya dengan cara mencopot pemasangan Ionic 3 dan memasang Ionic 4 dengan nama *package* yang baru (Kode 2.24).

#### Kode 2.24: Pencopotan Ionic 3 dan Pemasangan Ionic 4 dengan nama *package* baru

```
1 npm uninstall ionic-angular
2 npm install @ionic/angular>
```

#### ii. Penamaan Berkas

Terjadi perubahanan penamaan pada berkas di Ionic 3 dan Ionic 4. Daftar perubahan tersebut adalah sebagai berikut:

##### A. *Page*

Terdapat perbedaan nama *file* pada folder Pages. Perbedaan tersebut adalah sebagai berikut :

Pada Ionic 3 : home.html

Terdapat perubahan pada Ionic 4 menjadi : home.page.html

##### B. *App*

Terdapat perbedaan nama *file* pada direktori App. Perbedaan tersebut adalah sebagai berikut :

Pada Ionic 3 : app.html

Terdapat perubahan pada Ionic 4 menjadi : app-component.html

(d) Menyalin Global Scss dari **src/app/app.scss** pada Ionic 3, menjadi **src/global.scss** pada Ionic 4.

(e) Menyalin Bagian-bagian Aplikasi

Menyalin keseluruhan bagian yang ada pada aplikasi, baik itu halaman maupun fitur yang ada, dengan ketentuan sebagai berikut :

- Shadow DOM sudah aktif secara *default*.
- Halaman atau komponen Sass tidak lagi dibungkus dengan tag halaman / komponen dan harus menggunakan opsi styleUrls milik Angular dari dekorator @Component.
- RxJS

Pada Ionic 3, RxJS yang digunakan adalah versi 5. Sedangkan pada Ionic 4, RxJS yang digunakan adalah versi 6.

- Lifecycle Hooks tertentu harus digantikan dengan Angular Hooks.
- Perubahan markup yang mungkin saja dibutuhkan.

Sejak Ionic 4 dipindahkan ke elemen kustom, terdapat perubahan yang signifikan terkait dengan markup untuk setiap komponen. Semua perubahan ini dibuat untuk mengikuti spesifikasi dari elemen kustom. Komponen-komponen yang berubah tersebut yaitu :

##### – *Button*

Terdapat perbedaan pada *tag* untuk membuat Button, yang semula pada Ionic 3 adalah <button> menjadi <ion-button> pada Ionic 4 (Kode 2.25).

Kode 2.25: Penggunaan Button pada Ionic 4

```

1 <ion-button (click)="doSomething()">
2   Default Button
3 </ion-button>

```

– Floating Action Button (FAB)

Terdapat perbedaan pada *tag* di dalam `<ion-fab>`, yang semula pada Ionic 3 adalah `<button>` menjadi `<ion-fab-button>` pada Ionic 4 (Kode 2.26).

Kode 2.26: Penggunaan Floating Action Button pada Ionic 4

```

9 <ion-fab>
10   <ion-fab-button>
11     <ion-icon name="add"></ion-icon>
12   </ion-fab-button>
13   <ion-fab-list>
14     <ion-fab-button>
15       <ion-icon name="logo-facebook"></ion-icon>
16     </ion-fab-button>
17   </ion-fab-list>
18 </ion-fab>
19

```

– Item

Terdapat perbedaan pada *tag* `<ion-item>`, dimana pada Ionic 3, *tag* `<ion-label>` akan secara otomatis ditambahkan ke dalam `<ion-item>`. Sedangkan pada Ionic 4 diharuskan untuk menambahkan *tag* `<ion-label>` secara manual ke dalam komponen item (Kode 2.27).

Kode 2.27: Penggunaan Item pada Ionic 4

```

26 <ion-item>
27   <ion-label>
28     Default Item
29   </ion-label>
30 </ion-item>
31

```

– Label

Pada Ionic 4, atribut untuk mengatur posisi dari label digabungkan dengan atribut *position* (Kode 2.28).

Kode 2.28: Penggunaan Atribut *Position* pada Ionic 4

```

36 <ion-item>
37   <ion-label position="floating">Floating Label</ion-label>
38   <!-- input -->
39 </ion-item>
40

```

– Menu

Terdapat beberapa perubahan nama pada Ionic 4, yaitu :

\* Perubahan Nama Properti Terdapat perubahan nama properti pada Ionic 4.

Perubahan-perubahan tersebut adalah sebagai berikut :

- swipeEnable

Terdapat perubahan `swipeEnable` pada Ionic 4. Perubahan tersebut adalah sebagai berikut :

Pada Ionic 3 : `swipeEnabled`

Sedangkan pada Ionic 4 menjadi : `swipeGesture`

- 1 · content  
2 Terdapat perubahan content pada Ionic 4. Perubahan tersebut adalah  
3 sebagai berikut :

4 Pada Ionic 3 : content

5 Sedangkan pada Ionic 4 menjadi : contentId

- 6 \* Perubahan Nama Events Terdapat perubahan nama *events* pada Ionic 4.  
7 Perubahan-perubahan tersebut adalah sebagai berikut :

- 8 · ionClose

9 Terdapat perubahan ionClose pada Ionic 4. Perubahan tersebut adalah  
10 sebagai berikut :

11 Pada Ionic 3 : ionClose

12 Sedangkan pada Ionic 4 menjadi : ionDidClose

- 13 · ionOpen

14 Terdapat perubahan ionOpen pada Ionic 4. Perubahan tersebut adalah  
15 sebagai berikut :

16 Pada Ionic 3 : ionOpen

17 Sedangkan pada Ionic 4 menjadi : ionDidOpen

18 – Nav

19 Terdapat perubahan Nav pada Ionic 4. Perubahan-perubahan tersebut adalah  
20 sebagai berikut :

- 21 \* Perubahan Nama Method Terdapat perubahan nama *method* pada Ionic 4.  
22 Perubahan-perubahan tersebut adalah sebagai berikut :

- 23 · remove

24 Terdapat perubahan remove pada Ionic 4. Perubahan tersebut adalah  
25 sebagai berikut :

26 Pada Ionic 3 : remove

27 Sedangkan pada Ionic 4 untuk menghindari konflik dengan HTML, berubah  
28 menjadi : removeIndex

- 29 · getActiveChildNavs

30 Terdapat perubahan getActiveChildNavs pada Ionic 4. Perubahan tersebut  
31 adalah sebagai berikut :

32 Pada Ionic 3 : getActiveChildNavs

33 Sedangkan pada Ionic 4 menjadi : getChildNavs

- 34 \* Perubahan Nama Prop

35 Terdapat perubahan nama prop pada Ionic 4. Perubahan tersebut adalah  
36 sebagai berikut :

37 Pada Ionic 3 : swipeBackEnabled

38 Sedangkan pada Ionic 4 menjadi : swipeGesture

39 – Navbar

40 Pada Ionic 4, terdapat penghapusan terhadap komponen <ion-navbar> karena  
41 untuk menjaga agar selalu menggunakan <ion-toolbar> dengan *back button* yang  
42 eksplisit [2.29](#).

Kode 2.29: Penggunaan Navbar pada Ionic 4 dengan *Back Button*

```

1 <ion-toolbar>
2   <ion-buttons slot="start">
3     <ion-back-button></ion-back-button>
4   </ion-buttons>
5   <ion-title>My Navigation Bar</ion-title>
6 </ion-toolbar>

```

Selain yang telah disebutkan, terdapat beberapa perubahan lainnya yang tidak ditulis seperti Action Sheet, Alert, Colors, Content, Datetime, Dynamic Mode, Fixed Content, Grid, Icon, Infinite Scroll, Item, Item Divider, Item Options, Item Sliding, List Header, Loading, Modal, Option, Overlays, Popover, Radio, Range, Refresher, Scroll, Segment Button, Select, Show When, Hide When, Spinner, Tabs, Typography, Theming, Toast, dan Toolbar <sup>2</sup>.

## 2. Migrasi Ionic 4 ke Ionic 5

Migrasi aplikasi dari Ionic 4 ke Ionic 5 memerlukan beberapa pembaruan mengenai properti API, CSS, dan *package dependencies* yang terpasang. Perubahan-perubahan tersebut yaitu :

- CSS

- CSS Utilities

Karena pada versi sebelumnya, yaitu Ionic versi 4, terdapat masalah dengan menggunakan atribut CSS dengan *framework* yang menggunakan JSX dan TypeScript, Ionic *Framework* menambahkan dukungan untuk beberapa *framework*, dan pada Ionic 5 menambahkan kelas CSS. Ionic versi 5 menghapus atribut CSS dan mendukung konsistensi. Selain itu, Ionic versi 5 juga mengubah ke kelas dengan diawali ion untuk menghindari konflik dengan atribut asli dan CSS dari pengguna (Kode 2.30).

Kode 2.30: Contoh Kode Kelas CSS *Utility* pada Ionic 5

```

26 <ion-header class="ion-text-center"></ion-header>
27 <ion-content class="ion-padding"></ion-content>
28 <ion-label class="ion-text-wrap"></ion-label>
29 <ion-item class="ion-wrap"></ion-item>
30

```

- *Display Classes*

Kelas dari *responsive display* yang ditemukan di dalam berkas display.css memiliki kueri media yang diperbarui untuk lebih mencerminkan bagaimana cara kerjanya.

- *Activated, Focused, Hover States*

Kelas .activated secara otomatis ditambahkan ke komponen yang dapat diklik, mengalami perubahan nama menjadi .ion-activated. Selain itu juga memperbarui komponen Action Sheet sehingga variabel akan diawali dengan *button*. Hal ini dapat memungkinkan aplikasi tetap memiliki kontrol atas *opacity* jika diinginkan, tetapi saat memperbarui status, hanya perlu mengatur variabel utama, yaitu -background-activated, -background-focused, -background-hover. Hal tersebut penting saat mengubah tema global, karena memperbarui warna *toolbar* akan secara otomatis memperbarui *hover states* untuk semua *buttons* di *toolbar* (Kode 2.31).

---

<sup>2</sup> ‘Breaking Changes’ <https://github.com/ionic-team/ionic-framework/blob/main/angular/BREAKING.md>, Diakses pada 13 November 2021.

### Kode 2.31: Contoh Kode *Hover States* pada Ionic 5

```

1  /* Setting the button background on hover to solid red */
2  ion-button {
3      --background-hover: red;
4      --background-hover-opacity: 1;
5  }
6
7  /* Setting the action sheet button background on focus to an opaque green */
8  ion-action-sheet {
9      --button-background-focus: green;
10     --button-background-focus-opacity: 0.5;
11 }
12
13 /*
14  * Setting the fab button background on hover to match the text color with
15  * the default --background-hover-opacity on md
16 */
17 .md ion-fab-button {
18     --color: #222;
19     --background-hover: #222;
20 }
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

- *Distributed Scss*

Berkas scss telah dihapus dari dist/. Sebagai gantinya, variabel CSS harus digunakan untuk tema.

- Komponen

Terdapat perubahan beberapa komponen pada Ionic 5, yaitu :

- Back Button dan Button

Perubahan terdapat pada penambahan penamaan kelas .activated yang secara otomatis ditambahkan ke komponen yang dapat diklik, menjadi .ion-activated.

- Controllers

Terdapat beberapa komponen yang dihapus dari Ionic sebagai elemen, yaitu ion-action-sheet-controller, ion-alert-controller, ion-loading-controller, ion-menu-controller, ion-modal-controller, ion-picker-controller, ion-popover-controller, dan ion-toast-controller. Sebagai gantinya, maka harus diimpor dari @ionic/core.

- Header dan Footer

Atribut no-border dihapus, dan sebagai gantinya yaitu dengan menggunakan kelas ion-no-border.

- List Header

Konten berupa teks apa pun di dalam <ion-list-header> harus dibungkus dengan <ion-label> sesuai dengan gaya desain yang baru (Kode 2.32). Jika label tidak ada, maka perataan tombol di header bisa saja terlihat tidak aktif.

### Kode 2.32: Kode Program untuk List Header

```

1 <ion-list-header>
2     <ion-label>New This Week</ion-label>
3     <ion-button>See All</ion-button>
4 </ion-list-header>
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

- Menu

Fungsi swipeEnable() telah dihapus di Angular, sebagai gantinya menggunakan swipeGesture(). Lalu nilai *left* dan *right* telah dihapus, gunakan *start* dan *end* sebagai gantinya. Selain itu ada penghapusan atribut utama, sebagai gantinya yaitu dengan menggunakan content-id (untuk vanila JS atau Vue) dan contentId (untuk Angular atau React) (Kode 2.33).

Kode 2.33: Kode Program untuk Menu

```

1 <ion-menu content-id="main"></ion-menu>
2 <ion-content id="main">...</ion-content>

```

5 – Select Option

6 Properti selected telah dihapus. Sebagai gantinya harus mengatur properti nilai  
7 pada ion-select induk agar sesuai dengan opsi terpilih yang diinginkan (Kode 2.34).

Kode 2.34: Kode Program untuk Select Option

```

8 <ion-select value="two">
9   <ion-select-option value="one">One</ion-select-option>
10  <ion-select-option value="two">Two</ion-select-option>
11 </ion-select>

```

1 – Toast

2 Properti close button seperti showCloseButton dan closeButtonText telah dihapus.  
3 Sebagai gantinya, gunakan buttons array untuk fungsi batal (Kode 2.35).

Kode 2.35: Kode Program untuk Toast

```

4 async presentToast() {
5   const toast = await this.toastController.create({
6     message: 'Your_settings_have_been_saved.',
7     buttons: [
8       {
9         text: 'Close',
10        role: 'cancel',
11        handler: () => {
12          console.log('Close_clicked');
13        }
14      }
15    ]
16  });
17  toast.present();
18}
19
20
21
22
23
24
25
26

```

Selain yang sudah disebutkan, terdapat beberapa komponen lain yang mendapat perubahan di Ionic 5, namun tidak ditulis di dalam dokumen skripsi ini. Komponen-komponen tersebut antara lain Action Sheet, Anchor, Card, FAB, Item, Menu Button, Nav Link, Radio, Segment, Segment Button, Skeleton Text, Split Pane, dan Tabs <sup>3</sup>.

- Warna

Terdapat perubahan terhadap warna bawaan milik ionic (Tabel 2.1).

Nama Warna	Kode HEX
primary	#3880ff
secondary	#3dc2ff
tertiary	#5260ff
success	#2dd36f
warning	#ffc409
danger	#eb445a
light	#f4f5f8
medium	#92949c
dark	#222428

Tabel 2.1: Tabel Warna Bawaan di Ionic 5

<sup>3</sup> ‘Breaking Changes’ <https://github.com/ionic-team/ionic-framework/blob/main/BREAKING.md>, Diakses pada 20 November 2021.

27     • Events

28       Pada Ionic 5, Events services di @ionic/angular telah dihapus. Sebagai gantinya gunakan  
29       Observables untuk arsitektur pub/sub, dan Redux untuk *advanced state management*.

30     • *Package* dan *Dependencies*

31       Untuk memasang *package* dan *dependencies* pada Angular, dapat memanfaatkan npm  
32       pada CLI, dengan menjalankan pemasangan pada *package* ionic-angular (Kode 2.36).  
33       Namun jika ingin membuat proyek baru, dapat dibuat dari CLI dan aplikasi yang ada  
34       dapat dimigrasikan secara manual.

Kode 2.36: Kode untuk Memasang *Package* dan *Dependencies* pada Angular

35  
36 1 `npm install @ionic/angular@latest @ionic/angular-toolkit@latest --save`

<sup>1</sup>

## BAB 3

<sup>2</sup>

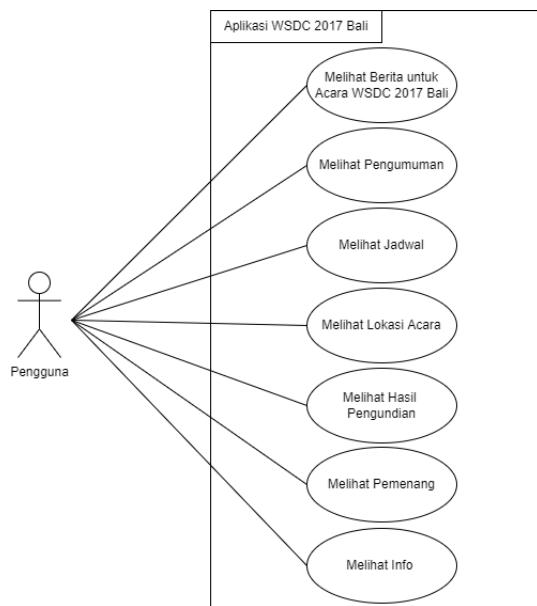
## ANALISIS

- <sup>3</sup> Pada bab ini akan dijelaskan analisis aplikasi WSDC 2017 Bali saat ini dan aplikasi WSDC yang  
<sup>4</sup> akan dibangun. Analisis yang akan dibahas meliputi analisis *use case*, analisis kebutuhan sistem,  
<sup>5</sup> dan analisis pembangunan aplikasi Android menggunakan Ionic.

### <sup>6</sup> 3.1 Analisis Sistem Kini

<sup>7</sup> Aplikasi WSDC 2017 Bali digunakan untuk menunjang keberlangsungan acara WSDC 2017 yang  
<sup>8</sup> diselenggarakan di Bali, Indonesia. Pada halaman utama, pengguna dapat melihat berita-berita  
<sup>9</sup> terkait acara WSDC 2017 Bali dan tombol *read more* yang apabila ditekan akan mengarahkan  
<sup>10</sup> pengguna untuk melihat berita terkait acara WSDC 2017 Bali dengan format pdf. Aplikasi WSDC  
<sup>11</sup> 2017 Bali dapat digunakan untuk melihat berita acara, pengumuman, jadwal peserta, lokasi acara,  
<sup>12</sup> hasil pengundian, info, serta pengumuman pemenang dari acara WSDC 2017 Bali (Gambar 3.1).

<sup>13</sup> Aplikasi WSDC 2017 Bali dibangun menggunakan *framework* Ionic versi 3, dan Angular versi  
<sup>14</sup> 4.1.3. Dengan digunakannya Ionic Framework, maka memungkinkan aplikasi WSDC 2017 Bali  
<sup>15</sup> menggunakan teknologi web seperti HTML, dan CSS. Lalu untuk membangun aplikasi WSDC 2017  
<sup>16</sup> Bali agar dapat berjalan secara *native*, digunakanlah Cordova. Penggunaan Cordova memungkinkan  
<sup>17</sup> aplikasi WSDC 2017 Bali kompatibel dengan perangkat berbasis Android dan IOS, tanpa perlu  
<sup>18</sup> mengimplementasikannya kembali ke dalam bahasa masing-masing platform.



Gambar 3.1: *Use Case Diagram* Aplikasi WSDC 2017 Bali

### **3.1.1 Skenario Pengguna**

Terdapat *sidebar* untuk pengguna agar dapat bermigrasi ke dalam menu-menu yang terdapat pada aplikasi WSDC 2017 Bali. Untuk mengakses *sidebar*, pengguna dapat menekan tombol navigasi berada di sebelah kiri atas aplikasi WSDC 2017 Bali. Selain itu dapat pula dengan cara mengusap layar dari kiri ke kanan. Untuk menutup *sidebar*, pengguna dapat menekan area di luar *sidebar*, atau dengan cara menekan tombol silang di sebelah kiri atas *sidebar*. Terdapat fitur-fitur yang ada pada aplikasi WSDC 2017 Bali yang dapat diakses melalui *sidebar*. Fitur-fitur tersebut adalah sebagai berikut :

1. *Home*

Pada halaman ini, pengguna dapat melihat halaman utama aplikasi WSDC 2017 Bali yang berisi berita acara WSDC 2017 Bali, serta pemberitahuan terakhir terkait acara WSDC 2017 Bali. Halaman ini merupakan halaman awal yang ditampilkan saat aplikasi WSDC 2017 Bali pertama kali dibuka (Tabel 3.1). Untuk mengakses halaman ini, dapat melalui *sidebar*.

No	Aksi Aktor	Reaksi Sistem
1	Pengguna membuka aplikasi WSDC 2017 Bali	Aplikasi WSDC 2017 Bali menampilkan halaman selamat datang.
2		Aplikasi WSDC 2017 Bali menampilkan halaman utama
3	Pengguna mengklik <i>card Announcements</i>	Aplikasi WSDC 2017 Bali menampilkan halaman Pemberitahuan.

Tabel 3.1: Tabel Skenario dari Halaman Utama

2. *Newsletter*

Pada bagian *newsletter* yang terdapat di *home*, pengguna dapat melihat berita-berita terkait acara WSDC 2017 Bali dengan format pdf. Untuk mengakses halaman ini, dapat melalui *sidebar* (Tabel 3.2).

No	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol <i>read more</i> pada berita di halaman utama aplikasi WSDC 2017 Bali.	Aplikasi WSDC 2017 Bali menampilkan berita pada acara WSDC 2017 Bali

Tabel 3.2: Tabel Skenario dari Berita

1      3. *Announcement*

2      Pengguna dapat melihat berbagai pengumuman mengenai keberlangsungan acara WSDC 2017  
 3      Bali yang tersusun berdasarkan tanggal dirilisnya pengumuman tersebut. Untuk mengakses  
 4      halaman ini, dapat melalui sidebar (Tabel 3.3).

No	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol Announcement pada <i>sidebar</i>	Aplikasi WSDC 2017 Bali menampilkan halaman pengumuman.

Tabel 3.3: Tabel Skenario dari Halaman Pemberitahuan

5      4. *Schedule*

6      Pada halaman ini, pengguna dapat melihat jadwal acara WSDC 2017 Bali yang ditampilkan  
 7      berdasarkan tanggal dan hari. Jadwal yang ditampilkan berupa waktu mulai dan waktu  
 8      selesai, lokasi acara, serta nama acara. Untuk mengakses halaman ini, dapat melalui sidebar  
 9      (Tabel 3.4).

No	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol Schedule pada <i>sidebar</i>	Aplikasi WSDC 2017 Bali menampilkan halaman jadwal.
2	Pengguna menekan tanggal yang berada di atas halaman jadwal	Aplikasi WSDC 2017 Bali menampilkan jadwal berdasarkan tanggal yang dipilih oleh pengguna dengan detail waktu, lokasi, dan nama kegiatan.

Tabel 3.4: Tabel Skenario dari Halaman Jadwal

10     5. *Venues*

11     Pada halaman ini, pengguna dapat melihat lokasi dari berlangsungnya acara WSDC 2017  
 12    Bali. Untuk mengakses halaman ini, dapat melalui sidebar (Tabel 3.5).

No	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol Venues pada <i>sidebar</i>	Aplikasi WSDC 2017 Bali menampilkan halaman Venues yang berisi <i>Ceremony Venues</i> , <i>Competition Venues</i> , <i>Delegates Accommodation</i> , dan <i>Educational Tour</i> .
2	Pengguna menekan kategori <i>venues</i> yang diinginkan.	Aplikasi WSDC 2017 Bali menampilkan peta, nama lokasi acara dengan disertai penanda yang ada di dalam peta, dan jarak antara lokasi pengguna saat ini dan lokasi acara.

Tabel 3.5: Tabel Skenario dari Halaman *Venues*

1      6. *Draw*

2      Pada halaman ini, pengguna dapat melihat pembagian *venue* serta pembagian kubu proposisi  
 3      dan oposisi dari hasil pengundian untuk para negara peserta WSDC 2017 Bali. Untuk  
 4      mengakses halaman ini, dapat melalui sidebar (Tabel 3.6).

No	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol Draw pada sidebar	Aplikasi WSDC 2017 Bali menampilkan halaman Draw yang dapat digulir kebawah untuk menampilkan keseluruhan tabel.

Tabel 3.6: Tabel Skenario dari Halaman *Draw*

5      7. *Result*

6      Pada halaman ini, pengguna dapat melihat pemenang dari kompetisi WSDC 2017 Bali. Untuk  
 7      mengakses halaman ini, dapat melalui sidebar (Tabel 3.7).

No	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol Result pada sidebar	Aplikasi WSDC 2017 Bali menampilkan halaman Result yang berisi pemenang dari babak semifinal, perempatfinal, dan perdelapanfinal.

Tabel 3.7: Tabel Skenario dari Halaman Hasil

8      8. Info

9      Pada halaman ini, pengguna dapat melihat info-info seputar kontak-kontak penting yang  
 10     dapat dihubungi, kosa kata dalam Bahasa Indonesia sehari-hari, serta *credits* kepada pembuat  
 11     aplikasi WSDC 2017 Bali. Untuk mengakses halaman ini, dapat melalui sidebar (Tabel 3.8).

No	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol <i>hamburger</i> di pojok kiri atas atau melakukan <i>swipe</i> dari kiri layar ke kanan layar aplikasi WSDC 2017 Bali.	Aplikasi WSDC 2017 Bali menampilkan sidebar
2	Pengguna menekan tombol Info	Aplikasi WSDC 2017 Bali menampilkan halaman Info

Tabel 3.8: Tabel Skenario dari Halaman Info

12     3.1.2 Struktur Ionic 3

13    Aplikasi WSDC 2017 Bali saat ini menggunakan Ionic versi 3, Angular versi 4.0.0, dan Cordova.  
 14    Dengan Ionic Framework yang disusun berdasarkan arsitektur Angular, maka aplikasi WSDC 2017  
 15    memungkinkan untuk ditulis menggunakan bahasa pemrograman web seperti HTML, CSS, dan  
 16    Javascript. Pada Ionic Framework versi 3 juga terdapat UI Component 2.3.2 yang digunakan dalam  
 17    aplikasi WSDC 2017 Bali, diantarnya yaitu Badge, Button, Card, Content, Grid, Icons, Items, List,  
 18    Menu, Segment, Slides, Tabs, dan Toolbar. Kemudian dengan digunakannya Cordova, maka seluruh  
 19    kode program yang menggunakan bahasa pemrograman web tersebut, dapat hidup dan berjalan  
 20    seperti halnya aplikasi *native* di dalam perangkat seluler.

1 Anatomi pada Ionic Framework memiliki struktur proyek Cordova. Pada saat pertama kali  
 2 dijalankan, aplikasi WSDC 2017 Bali secara *default* akan membuka file index.html yang berada  
 3 di folder src/index.html. File ini merupakan file pertama yang dijalankan untuk aplikasi WSDC  
 4 2017 Bali. Tujuan dari file ini adalah untuk melakukan pengaturan terhadap script, CSS, serta  
 5 menjalankan aplikasi. Di dalam file index.html ini terdapat sebuah tag <ion-app>. Tag ini yang  
 6 pertama dicari dan dijalankan oleh Ionic untuk membuka komponen *root* dari aplikasi WSDC 2017  
 7 Bali. Pada saat pertama menjalankan aplikasi, kode di dalam folder src akan ditranspilasikan  
 8 ke versi JavaScript yang dapat dipahami browser. Dengan begitu, aplikasi dapat menjalankan  
 9 TypeScript yang dikompilasi ke bentuk JavaScript.

10 Setelah index.html dijalankan, titik masuk ke dalam aplikasi WSDC 2017 Bali adalah file  
 11 app.module.ts yang berada di src/app/app.module.ts. Di dalam file ini terdapat NgModule untuk  
 12 mendeklarasi komponen apa saja yang akan digunakan, mengimpor module, bootstrap apa yang  
 13 digunakan, dan menyediakan *services* apa yang akan digunakan oleh komponen (Kode 3.1).

Kode 3.1: NgModule pada app.module.ts

```

14 @NgModule({
15   declarations: [
16     MyApp, HomePage, AnnouncementsPage, SchedulePage, VenuesPage, VenuesMapPage, DrawPage, ResultPage, InfoPage
17   ],
18   imports: [
19     BrowserModule, HttpModule, IonicModule.forRoot(MyApp), IonicModule.forRoot(), CloudModule.forRoot(cloudSettings)
20   ],
21   bootstrap: [IonicApp],
22   entryComponents: [
23     MyApp, HomePage, AnnouncementsPage, SchedulePage, VenuesPage, VenuesMapPage, DrawPage, ResultPage, InfoPage
24   ],
25   providers: [
26     StatusBar, SplashScreen, InAppBrowser, {provide: ErrorHandler, useClass: IonicErrorHandler}, Geolocation,
27   ]
28 }
29 )
30 export class AppModule {}
```

32 Lalu, untuk komponen *root*, diatur ke MyApp. Komponen tersebut berada di folder src/ap-  
 33 p/app.component.ts. Karena pada file app.component.ts, *root* telah diatur ke dalam MyApp, maka  
 34 komponen tersebut menjadi komponen pertama yang dibuka ke dalam aplikasi WSDC 2017 Bali. Di  
 35 dalam komponen tersebut terdapat templateUrl yang digunakan sebagai template utama dari apli-  
 36 kasi WSDC 2017 Bali, yaitu file app.html (Kode 3.2). Di dalam template, terdapat tag <ion-menu>  
 37 yang digunakan untuk menampilkan sidebar, lalu tag <ion-nav> sebagai area koten utama, dengan  
 38 properti [root] = "rootPage". Properti tersebut yang nantinya akan diisi oleh halaman *root* dari  
 39 aplikasi WSDC 2017 Bali, yaitu Home Page. Variabel rootPage telah diatur di file app.component.ts  
 40 secara spesifik mengarah ke HomePage, yang akan menjadi halaman petama yang ditampilkan di  
 41 nav controller.

Kode 3.2: Source Code File app.html

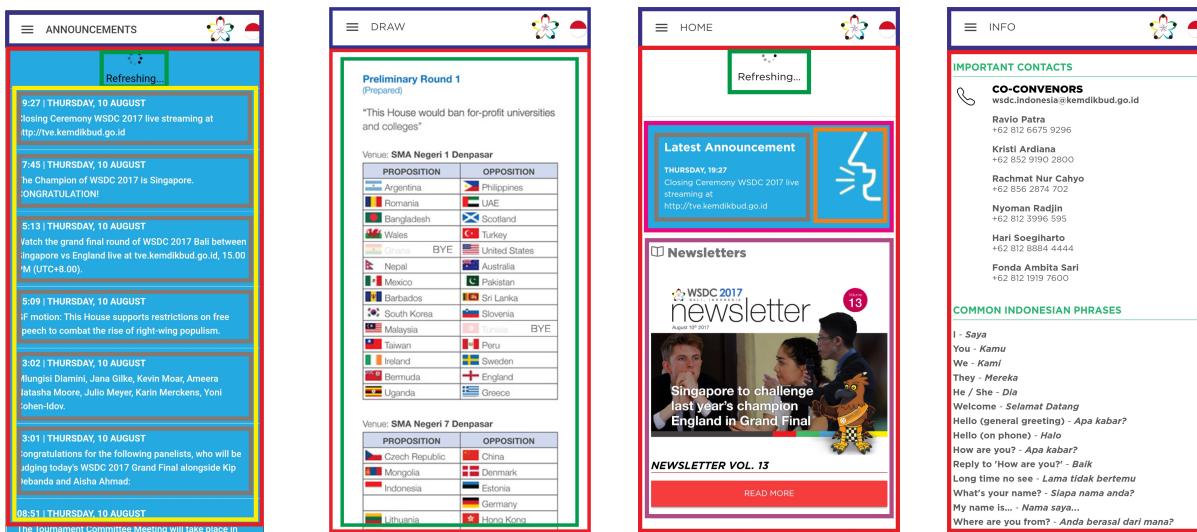
```

1  <ion-menu [content]="content">
2    <ion-header>
3      <ion-toolbar>
4        <ion-title>
5          <button menuClose id="menu-close-btn">
6            <ion-icon menu-close ios="ios-close-circle-outline" md="md-close-circle"></ion-icon>
7          </button>
8          <span class="text">Menu</span>
9        </ion-title>
10       </ion-toolbar>
11     </ion-header>
12
13   <ion-content>
14     <ion-list>
15       <button class="title-sidemenu" menuClose ion-item *ngFor="let p of pages" (click)="openPage(p)">
16         <ion-icon [ios]=p.iosicon [md]=p.mdicon></ion-icon>
17         <span class="text">{{p.title}}</span>
18       </button>
19     </ion-list>
20   </ion-content>
21
22 </ion-menu>
23
24 <!-- Disable swipe-to-go-back because it's poor UX to combine STGB with side menus -->
25 <ion-nav [root]="rootPage" #content swipeBackEnabled="false"></ion-nav>

```

Selain komponen *root*, terdapat pula beberapa komponen lain yang berisi halaman-halaman yang ada di aplikasi WSDC 2017 Bali. Masing-masing komponen akan mengimpor Component dari @angular/core, NavController dari ionic-angular, dan Storage dari @ionic/storage. Mengimpor Component dari @angular/core berfungsi untuk menambahkan sebuah komponen ke dalam *module*. Dengan begitu, komponen tersebut bisa terlihat di seluruh aplikasi, dan dapat digunakan oleh komponen lain. Lalu, NavController merupakan *base class* untuk mengatur komponen navigasi. Ini berguna agar aplikasi dapat berpindah antar halaman. Sedangkan Storage berfungsi untuk menyimpan pasangan *key/value* dan sebuah objek JSON.

Setiap komponen memiliki tiga buah *file* utama, yaitu *file* HTML, CSS, dan TypeScript. *File* HTML digunakan untuk menampilkan sebuah halaman ke dalam aplikasi dengan susunan kode HTML. Lalu *file* CSS digunakan untuk mengatur desain, bentuk, dan tampilan dari sebuah halaman. Sedangkan *file* TypeScript digunakan untuk mengontrol jalannya sebuah komponen.



(a) Announcement Page

(b) Draw Page

(c) Home Page

(d) Info Page

Gambar 3.2: Wireframe Aplikasi WSDC 2017 Bali

1 Komponen-komponen yang ada pada aplikasi WSDC 2017 Bali adalah sebagai berikut:

2 • Komponen *Announcement*

3 Komponen ini digunakan untuk menampilkan halaman *Announcement* pada aplikasi. Kom-  
4 ponen ini memiliki sebuah *file* TypeScript untuk mengatur keseluruhan halaman. Di dalam  
5 *file* announcement.ts terdapat sebuah *decorator* @Component untuk komponen (Kode 3.3).  
6 Di dalam decorator ini terdapat CSS *selector* untuk memilih CSS yang akan digunakan,  
7 serta **templateUrl** untuk mendefinisikan eksternal HTML *template* yang akan digunakan.  
8 *Template* HTML yang digunakan adalah *file* announcement.html.

Kode 3.3: @Component pada announcement.ts

```
9
10 1 @Component({
11 2   selector: 'page-announcements',
12 3   templateUrl: 'announcements.html'
13 4 })
```

15 Lalu, terdapat sebuah kelas AnnouncementsPage pada komponen ini yang berisi beberapa  
16 *method* yang akan digunakan di dalam aplikasi. *Method* pada kelas ini diantaranya adalah  
17 sebagai berikut:

18 – ionViewDidLoad()

19 *Method* ini merupakan sebuah *lifecycle event* yang dijalankan saat halaman *announcement*  
20 telah dimuat. *Event* ini hanya berjalan satu kali, jadi jika sudah melakukan cache  
21 terhadap halaman ini, maka ionViewDidLoad() tidak akan berjalan lagi. *Lifecycle event*  
22 ini berguna untuk melakukan pengaturan awal, yaitu untuk memuat data *announcement*  
23 dari *storage*, dan menyimpannya di dalam variabel lokal.

24 – doRefresh(refresher)

25 *Method* ini berfungsi untuk melakukan penyegaran ulang pada halaman *announcement*  
26 untuk mendapatkan data *announcement* terbaru di dalam server, kemudian menyimpan-  
27 nya ke dalam penyimpanan Ionic. *Method* ini memiliki sebuah parameter refresher, yang  
28 berisi sebuah CustomEvent dari penyegaran ulang yang dilakukan.

29 – presentConnectionAlert()

30 *Method* ini digunakan ketika method doRefresh() mengalami *error*, yang kemudian  
31 memunculkan *toast*.

32 – formatDate(sqlDatetime: string)

33 *Method* ini berfungsi untuk membuat format tanggal dan waktu. *Method* ini memiliki  
34 sebuah parameter, yaitu sqlDatetime yang bertipe *string*, yang merupakan sebuah  
35 *string* tanggal dengan format "tahun-bulan-hari jam-menit-detik". *Method* ini akan  
36 mengembalikan sebuah teks yang berisi waktu, tanggal dan bulan.

37 *File* announcement.html digunakan untuk menampilkan halaman *announcemnet*. Terdapat  
38 beberapa komponen yang disediakan oleh Ionic Framework, yang diimplementasikan ke dalam  
39 halaman *announcement*. Diantaranya adalah sebagai berikut:

40 – *Header*

41 Pada *header* dari halaman *announcement*, digunakan beberapa *tag* dari komponen yang  
42 disediakan oleh Ionic Framework (Kode 3.4). Yaitu *tag* <ion-header> yang merupakan  
43 komponen *parent* yang menampung komponen *toolbar* yang ditandai dengan warna biru  
44 pada gambar 3.2a. Di dalam *tag* tersebut terdapat *tag* pendukung, seperti <ion-navbar>,

1 <button> sebagai tombol untuk membuka *sidebar*, <ion-icon> untuk menampilkan icon  
 2 dari tombol pada *tag button*, dan <ion-title> untuk menampilkan judul dari halaman,  
 3 yaitu *Announcement*, pada *navbar*.

Kode 3.4: *Header* pada halaman *Announcement*

```
4
5   <ion-header>
6     <ion-navbar>
7       <button ion-button menuToggle>
8         <ion-icon name="menu"></ion-icon>
9       </button>
10      <ion-title>Announcements</ion-title>
11    </ion-navbar>
12  </ion-header>
```

#### 14 – Content

15 Konten pada halaman *announcement* yang ditandai dengan kotak berwarna merah  
 16 pada gambar 3.2a disusun menggunakan *tag* <ion-content> (Kode 3.5). *Tag* ini berisi  
 17 beberapa *tag* lain, yaitu *tag* <ion-refresher>, yang ditandai dengan kotak hijau, yang akan  
 18 menampilkan simbol *refresh* saat pengguna menyegarkan halaman dengan cara melakukan  
 19 *swipe* dari atas ke bawah layar. Kemudian terdapat *tag* <ion-list> yang ditandai  
 20 dengan kotak kuning, berfungsi untuk menampilkan baris. Baris-baris tersebut diisi  
 21 menggunakan *tag* <ion-item> yang ditandai dengan kotak berwarna hitam, digunakan  
 22 untuk menyimpan teks yang berisi tanggal, dan pesan pengumuman.

Kode 3.5: *Content* pada halaman *annoncement*

```
23
24   <ion-content>
25     <ion-refresher (ionRefresh)="doRefresh($event)">
26       <ion-refresher-content pullingIcon="arrow-dropdown" pullingText="Pull_to_refresh" refreshingSpinner="circles"
27         refreshingText="Refreshing...">
28       </ion-refresher-content>
29     </ion-refresher>
30     <ion-list>
31       <ion-item text-wrap *ngFor="let_announcement_of_announcements">
32         <h3>{{formatDatetime(announcement.localtime)}}</h3>
33         <p>{{announcement.message}}</p>
34       </ion-item>
35     </ion-list>
36   </ion-content>
```

- 38 • Komponen *Draw*

39 Komponen *draw* digunakan untuk menampilkan halaman *Draw* pada aplikasi. Terdapat *file*  
 40 *TypeScript*, draw.ts, yang berfungsi untuk mengatur keseluruhan halaman. Di dalam *file*  
 41 tersebut terdapat *decorator* @Component (Kode 3.6) dan *decorator* @ViewChild (Kode 3.7).  
 42 Pada *decorator* @Component, terdapat CSS *selector* untuk memilih CSS mana yang akan  
 43 digunakan, serta **templateUrl** untuk mendefinisikan eksternal HTML *template* halaman  
 44 *Draw* yang akan digunakan, yaitu draw.html. Lalu, @ViewChild digunakan untuk memanggil  
 45 elemen dari DOM untuk memanggil komponen API ke dalam *TypeScript*, yaitu pada  
 46 komponen *draw* adalah drawIFrame yang berada di *file* draw.html.

Kode 3.6: @Component pada draw.ts

```
47
48   @Component({
49     selector: 'page-draw',
50     templateUrl: 'draw.html'
51   })
```

## Kode 3.7: @ViewChild pada draw.ts

```
1 3 @ViewChild('drawIFrame') drawIFrame: ElementRef;
```

4 Lalu, terdapat kelas DrawPage yang berisi beberapa *method* yang akan digunakan di dalam  
5 aplikasi, diantaranya adalah sebagai berikut:

- 6 – ionViewDidLoad()

7 *Method* ini merupakan sebuah *lifecycle event* yang dijalankan saat halaman *draw* telah  
8 dimuat. *Event* ini hanya berjalan satu kali, jadi jika sudah melakukan chache terhadap  
9 halaman ini, maka ionViewDidLoad() tidak akan berjalan lagi. *Lifecycle event* ini  
10 berguna untuk melakukan pengaturan awal, yaitu untuk memuat data *draw* dari *storage*,  
11 kemudian data tersebut dimasukkan ke dalam *child* drawIFrame. Terakhir, method ini  
12 akan memanggil method presentLoading().

- 13 – presentLoading()

14 *Method* ini berfungsi untuk menampilkan sebuah *overlay* yang menunjukkan sebuah pesan  
15 dan indikator pemuatan saat pertama kali halaman *draw* dimuat. Karena *overlay* ini  
16 muncul di atas konten aplikasi, maka aktivitas pengguna akan diblokir untuk sementara  
17 selama 3 detik.

- 18 – onDrawIframeLoad()

19 *Method* ini merupakan sebuah *template statement* dipanggil oleh *event* di dalam tag  
20 <iframe> pada draw.html yaitu *event* (load). *Method* ini berfungsi untuk menampilkan  
21 data yang telah diambil yang disimpan di dalam *child* drawIFrame.

22 Selain itu terdapat file draw.html yang digunakan untuk menampilkan tata letak dari  
23 halaman *draw*. Terdapat beberapa komponen yang disediakan oleh Ionic Framework, yang  
24 diimplementasikan ke dalam halaman *draw*. Diantaranya adalah sebagai berikut:

- 25 – *Header*

26 *Header* dari halaman *draw* seperti pada gambar 3.2b menggunakan tag <ion-header>  
27 (Kode 3.8). Tag tersebut merupakan komponen *parent* yang menampung komponen  
28 navbar yang ditandai dengan kotak berwarna biru pada gambar 3.2b. Di dalam navbar  
29 tersebut, terdapat sebuah tag <button> untuk memunculkan sidebar, <ion-icon> untuk  
30 menampilkan icon dari tombol pada tag button, dan tag <ion-title> sebagai judul dari  
31 halaman.

Kode 3.8: *Header* pada draw.html

```
32 1 <ion-header>
33 2   <ion-navbar>
34 3     <button ion-button menuToggle>
35 4       <ion-icon name="menu"></ion-icon>
36 5     </button>
37 6     <ion-title>Draw</ion-title>
38 7   </ion-navbar>
39 8 </ion-header>
```

- 42 – *Content*

43 *Content* dari halaman *draw* seperti pada gambar 3.2b menggunakan tag <ion-content>  
44 (Kode 3.9) yang ditandai menggunakan kotak berwarna merah. Di dalam tag ini terdapat  
45 sebuah tag <iframe> yang berisi hasil pengundian grup untuk peserta WSDC 2017 Bali,  
46 ditandai menggunakan kotak berwarna hijau. Tag <iframe> menampilkan hasil dari  
47 *method* onDrawIframeLoad() pada draw.ts.

Kode 3.9: Content pada draw.html

```

1 <ion-content>
2   <iframe #drawIFrame (load)="onDrawIframeLoad()" class="iframe-fullscreen"></iframe>
3 </ion-content>

```

- Komponen *Home*

Komponen ini digunakan untuk menampilkan halaman *Home* pada aplikasi. Komponen ini memiliki sebuah *file* TypeScript untuk mengatur keseluruhan halaman. Di dalam *file* home.ts terdapat sebuah *decorator* @Component untuk komponen (Kode 3.10). Di dalam decorator ini terdapat CSS *selector* untuk memilih CSS yang akan digunakan, serta templateUrl untuk mendefinisikan ekxternal HTML *template* yang akan digunakan. *Template* HTML yang digunakan adalah *file* home.html.

Kode 3.10: @Component pada home.ts

```

1 @Component({
2   selector: 'page-home',
3   templateUrl: 'home.html'
4 })

```

Komponen *Home* merupakan komponen yang menjadi *rootPage* dari aplikasi ini, yang dimasukan di dalam *file* app.component.ts. Maka dari itu, saat pertama kali aplikasi dijalankan, komponen *home*-lah yang pertama kali ditampilkan di dalam layar. *rootPage* di dalam *file* app.component.ts akan memanggil komponen *home*, yang kemudian *file* home.ts akan berjalan. Di dalam *file* ini terdapat sebuah kelas HomePage yang berisi beberapa *method*, diantaranya adalah sebagai berikut:

- ionViewDidLoad()

*Method* ini merupakan sebuah *lifecycle event* yang dijalankan saat halaman *home* telah dimuat. *Event* ini hanya berjalan satu kali, jadi jika sudah melakukan chache terhadap halaman ini, maka ionViewDidLoad() tidak akan berjalan lagi. *Lifecycle event* ini berguna untuk melakukan pengaturan awal, yaitu untuk untuk mengambil keseluruhan data aplikasi yang ada di dalam penyimpanan. Dengan memanfaatkan fitur *storage* yang dimiliki oleh Ionic Framework, *method* ini akan mengecek apakah sudah ada data dengan format .json yang berisi keseluruhan data aplikasi di dalam penyimpanan. Data tersebut berisi data *announcements*, *newsletters*, *schedule*, *venues*, *draws*, dan *info*. Jika data tersebut tidak ditemukan, maka akan diambil dari *file* wsdc-data.json yang kemudian dimasukan ke dalam penyimpanan.

Lalu, *method* ini akan melakukan penyegaran terhadap data yang sudah ada di dalam penyimpanan dengan mengambil data dari server dengan batas waktu maksimal untuk terhubung dengan server. Jika sudah melewati batas waktu, dan aplikasi belum terhubung dengan server, maka *method* ini akan memanggil *method* showToast() yang akan menampilkan Toast yang berisi teks ‘Failed to refresh information’. Jika penyegaran berhasil, maka data yang didapatkan dari server akan dimasukan ke dalam penyimpanan internal.

- launch(url: string)

*Method* ini merupakan sebuah *template statement* dipanggil oleh *event* di dalam tag <button> pada home.html yaitu *event* (click). *Method* ini memiliki sebuah parameter url yang bertipe stirng. Parameter tersebut berisi url dari berita yang akan dilihat oleh peng-

1 gunga. Untuk membuka berita pada url tersebut memanfaatkan *plugin* InAppBrowser  
2 yang disediakan oleh Ionic.

3 – formatDatetime(sqlDatetime: string)  
4 *Method* ini berfungsi untuk membuat format tanggal dan waktu. *Method* ini memiliki  
5 sebuah parameter, yaitu sqlDatetime yang bertipe *string*, yang merupakan sebuah  
6 *string* tanggal dengan format “tahun-bulan-hari jam-menit-detik”. *Method* ini akan  
7 mengembalikan sebuah teks yang berisi waktu, tanggal dan bulan.

8 – doRefresh(refresher)  
9 *Method* ini berfungsi untuk melakukan penyegaran ulang pada halaman *home* untuk  
10 mendapatkan data *home* terbaru di dalam server, kemudian menyimpannya ke dalam  
11 penyimpanan. *Method* ini memiliki sebuah parameter refresher, yang berisi sebuah  
12 CustomEvent dari penyegaran ulang yang dilakukan. *Method* ini akan melakukan  
13 pemanggilan kembali kepada server, dalam batas waktu tertentu. Jika batas waktu  
14 maksimal telah tercapai, sedangkan server belum juga memberi tanggapan, maka akan  
15 memanggil *method* showToast() yang akan menampilkan sebuah Toast yang berisi teks  
16 ‘Failed to refresh information’. Jika berhasil untuk terhubung dengan server, *method* ini  
17 akan menghapus data yang berada di penyimpanan, dan digantikan dengan data yang  
18 telah didapatkan dari server.

19 – showToast(message: string, duration: number = 3000)  
20 *Method* ini berfungsi untuk memunculkan sebuah native Toast, yaitu sebuah *popup* teks,  
21 dengan memanfaatkan UI Component milik Ionic Framework. *Method* ini menerima  
22 parameter berupa sebuah *string*, yang berisi pesan yang akan dimunculkan ke dalam  
23 sebuah Toast, dan memiliki sebuah parameter *duration* yang berisi lama waktu

24 – onAnnouncementClick()  
25 *Method* ini merupakan sebuah *template statement* dipanggil oleh *event* di dalam tag  
26 <ion-card> pada home.html yaitu *event* (click). *Method* ini berfungsi untuk berpindah  
27 halaman menjadi halaman *announcement*.

28 File home.html digunakan untuk menampilkan tata letak dari halaman *home*. Terdapat  
29 beberapa komponen yang disediakan oleh Ionic Framework, yang diimplementasikan ke dalam  
30 halaman *home*. Diantaranya adalah sebagai berikut:

31 – Header  
32 Halaman *home* memiliki *header* dengan tag <ion-header> (Kode 3.11) seperti pada gam-  
33 bar 3.2c yang ditandai dengan kotak berwarna biru. Tag tersebut merupakan komponen  
34 *parent* yang menampung komponen navbar yang ditandai dengan kotak berwarna biru.  
35 Di dalam navbar tersebut, terdapat sebuah tag <button> untuk memunculkan *sidebar*,  
36 <ion-icon> untuk menampilkan icon dari tombol pada tag *button*, dan tag <ion-title>  
37 sebagai judul dari halaman.

Kode 3.11: *Header* pada home.html

```

1 <ion-header>
2   <ion-navbar>
3     <button ion-button menuToggle>
4       <ion-icon name="menu"></ion-icon>
5     </button>
6     <ion-title>Home</ion-title>
7   </ion-navbar>
8 </ion-header>
```

– *Content*

*Content* pada halaman *home* dengan tag *<ion-content>* (Kode 3.12 pada gambar 3.2c ditandai dengan kotak berwarna merah. Di dalam tag *<ion-content>* terdapat beberapa tag lainnya. Pertama yaitu sebuah tag *<ion-refresher>* yang digunakan untuk menampilkan simbol *refresh* saat pengguna menyegarkan halaman dengan cara melakukan *swipe* dari atas ke bawah layar, ditandai dengan kotak berwarna hijau.

Lalu terdapat tag *<ion-card>* yang digunakan sebagai tempat untuk pengumuman terkait acara WSDC 2017 Bali disimpan. Penggunaan *card* ditantai dengan kotak berwarna merah muda. Di dalam tag *<ion-card>* terdapat tag *<ion-grid>* untuk mengatur tata letak dari penyusunan isi dari suatu *card*. Di dalam *grid* tersebut terdapat satu baris dengan tag *<ion-row>* dan dua kolom dengan tag *<ion-col>*. Kolom pertama ditandai dengan kotak berwarna coklat, berisi tanggal beserta pengumuman. Lalu kolom kedua ditandai dengan kotak berwarna oranye berisi gambar.

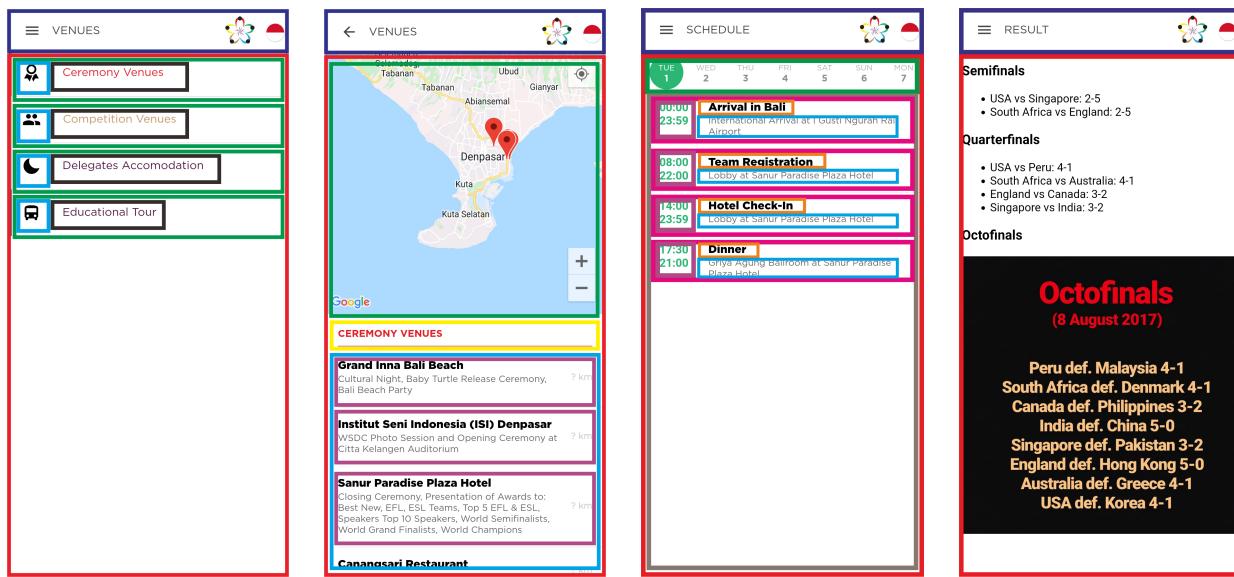
Selanjutnya terdapat sebuah tag *<ion-list>* untuk menyimpan berita-berita terkait acara WSDC 2017 Bali, yang ditandai dengan warna ungu. Di dalam *list* tersebut terdapat tag *<ion-list-header>* sebagai judul dari *list*, dan tag *<ion-item>* untuk menyimpan berita-berita terkait acara WSDC 2017 Bali. Di dalam tag *<ion-item>* terdapat tag *<button>* yang apabila ditekan oleh pengguna, maka akan mengarahkan pengguna untuk melihat berita tertentu sesuai dengan *item* yang dipilih dengan memanggil *method* *launch()* yang ada di *home.ts*.

Kode 3.12: Content pada home.html

```

1 <ion-content>
2   <ion-refresher (ionRefresh)="doRefresh($event)">
3     <ion-refresher-content pullingIcon="arrow-dropdown" pullingText="Pull_to_refresh" refreshingSpinner="circles"
4       refreshingText="Refreshing...">
5     </ion-refresher-content>
6   </ion-refresher>
7   <ion-card (click)="onAnnouncementClick()">
8     <ion-grid>
9       <ion-row>
10      <ion-col col-9>
11        <ion-card-header text-wrap>
12          Latest Announcement
13        </ion-card-header>
14        <ion-card-content>
15          <h3>{{formatDatetime(wsdcData?.announcements[0].localtime)}}</h3>
16          <p>{{wsdcData?.announcements[0].message}}</p>
17        </ion-card-content>
18      </ion-col>
19      <ion-col col-3>
20        
21      </ion-col>
22    </ion-row>
23  </ion-grid>
24 </ion-card>
25 <ion-list>
26   <ion-list-header>
27     <ion-icon ios="ios-book-outline" md="md-book"></ion-icon>
28     Newsletters
29   </ion-list-header>
30   <ion-item *ngFor="let_wsdNews_of_wsdData?.newsletters">
31     
33     <h2 text-wrap>{{wsdNews.title}}</h2>
34     <button ion-button full block color="danger" (click)="launch(wsdNews.url)">Read More</button>
35   </ion-item>
36 </ion-list>
37 </ion-content>
38

```



Gambar 3.3: Wireframe Aplikasi WSDC 2017 Bali

40 • Komponen Info

41 Komponen ini digunakan untuk menampilkan halaman Info pada aplikasi. Komponen ini  
42 memiliki sebuah file TypeScript untuk mengatur keseluruhan halaman. Di dalam file info.ts  
43 terdapat sebuah decorator @Component untuk komponen (Kode 3.13). Di dalam decorator  
44 ini terdapat CSS selector untuk memilih CSS yang akan digunakan, serta templateUrl

untuk mendefinisikan eksternal HTML *template* yang akan digunakan. *Template* HTML yang digunakan adalah *file* info.html.

Kode 3.13: @Component pada info.ts

```
3 @Component({
4   selector: 'page-info',
5   templateUrl: 'info.html'
6 })
7 }
```

Terdapat kelas InfoPage pada komponen info. Kelas ini hanya berisi *constructor* yang digunakan untuk menginisialisai halaman yang akan digunakan. *Constructor* sendiri berfungsi untuk memuat data info dari penyimpanan dan memasukannya ke dalam sebuah variabel lokal. Selain itu, terdapat *file* info.html yang digunakan untuk menampilkan tata letak dari halaman info. Terdapat beberapa komponen yang disediakan oleh Ionic Framework, yang diimplementasikan ke dalam halaman info. Diantaranya adalah sebagai berikut:

- *Header*

Halaman info memiliki *header* dengan tag <ion-header> (Kode 3.14) seperti pada gambar 3.2d. Tag tersebut merupakan komponen *parent* yang menampung komponen navbar yang ditandai dengan kotak berwarna biru. Di dalam navbar tersebut, terdapat sebuah tag <button> untuk memunculkan sidebar dan tag <ion-icon> untuk menampilkan icon dari tombol pada tag *button*. Lalu terdapat tag <ion-title> sebagai judul dari halaman, yaitu “Info”.

Kode 3.14: *Header* pada info.html

```
22 <ion-header>
23   <ion-navbar>
24     <button ion-button menuToggle>
25       <ion-icon name="menu"></ion-icon>
26     </button>
27     <ion-title>Info</ion-title>
28   </ion-navbar>
29 </ion-header>
```

- *Content*

*Content* pada halaman info memiliki tag <ion-content> (Kode 3.15) yang pada gambar 3.2d dengan kotak berwarna merah. Di dalam tag info terdapat tag <ion-grid> untuk mengatur *layout* dari *content*. Di dalam tag <ion-grid> terdapat sebuah tag <ion-row> yang berisi sebuah tag <div>. Tag tersebut berisi info yang di dapatkan pada *constructor* di *file* info.ts.

Kode 3.15: *Content* pada info.html

```
38 <ion-content>
39   <ion-grid>
40     <ion-row>
41       <div [innerHTML]=wsdcInfoData>
42         </div>
43       </ion-row>
44     </ion-grid>
45 </ion-content>
```

1     • Komponen *Result*

2       Komponen ini digunakan untuk menampilkan halaman *Result* pada aplikasi. Komponen ini  
 3       memiliki sebuah *file* TypeScript untuk mengatur keseluruhan halaman. Di dalam *file* result.ts  
 4       terdapat sebuah *decorator* @Component untuk komponen (Kode 3.16) dan *decorator* @View-  
 5       Child (Kode 3.21). Di dalam decorator ini terdapat CSS *selector* untuk memilih CSS yang  
 6       akan digunakan, serta **templateUrl** untuk mendefinisikan eksternal HTML *template* yang  
 7       akan digunakan. *Template* HTML yang digunakan adalah *file* info.html. Lalu, @ViewChild  
 8       digunakan untuk memanggil elemen dari DOM untuk memanggil komponen API ke dalam  
 9       TypeScript, yaitu pada komponen *result* adalah resultIFrame yang berada di *file* result.html.

Kode 3.16: @Component pada result.ts

```
10
11  @Component({
12    selector: 'page-result',
13    templateUrl: 'result.html'
14  })
```

Kode 3.17: @ViewChild pada result.ts

```
16
17  @ViewChild('resultIFrame') resultIFrame: ElementRef;
```

19       Di dalam *file* result.ts, terdapat beberapa *method*, yaitu:

20       – ionViewDidLoad()

21       *Method* ini merupakan sebuah *lifecycle event* yang dijalankan saat halaman *result* telah  
 22       dimuat. *Event* ini hanya berjalan satu kali, jadi jika sudah melakukan cache terhadap  
 23       halaman ini, maka ionViewDidLoad() tidak akan berjalan lagi. *Lifecycle event* ini  
 24       berguna untuk melakukan pengaturan awal, yaitu untuk memuat data *result* yang sudah  
 25       disimpan di dalam penyimpanan internal. Setelah berhasil memuat data *result*, data  
 26       tersebut akan dimasukan ke dalam *child* resultIFrame. Terakhir, akan dipanggil *method*  
 27       presentLoading().

28       – presentLoading()

29       *Method* ini berfungsi untuk menampilkan sebuah *overlay* yang menunjukkan sebuah pesan  
 30       dan indikator pemuatan saat pertama kali halaman *draw* dimuat. Karena *overlay* ini  
 31       muncul di atas konten aplikasi, maka aktivitas pengguna akan diblokir untuk sementara  
 32       selama 3 detik.

33       – onResultIframeLoad()

34       *Method* ini merupakan sebuah *template statement* dipanggil oleh *event* di dalam tag  
 35       <iframe> pada result.html yaitu *event* (load). *Method* ini berfungsi untuk menampilkan  
 36       data yang telah diambil yang disimpan di dalam *child* resultIFrame.

37       Selain itu, terdapat *file* result.html yang digunakan untuk menampilkan tata letak dari  
 38       halaman *result*. Terdapat beberapa komponen yang disediakan oleh Ionic Framework, yang  
 39       diimplementasikan ke dalam halaman *result*. Diantaranya adalah sebagai berikut:

1 – *Header*

2 Halaman *result* memiliki *header* dengan tag `<ion-header>` (Kode 3.18) seperti pada  
 3 gambar 3.3d. Tag tersebut merupakan komponen *parent* yang menampung komponen  
 4 navbar yang ditandai dengan kotak berwarna biru. Di dalam navbar tersebut, terdapat  
 5 sebuah tag `<button>` untuk memunculkan *sidebar*, dan `<ion-icon>` untuk menampilkan  
 6 icon dari tombol pada tag *button*. Lalu terdapat tag `<ion-title>` sebagai judul dari  
 7 halaman, yaitu “Result”.

Kode 3.18: *Header* pada result.html

```
8 <ion-header>
9   <ion-navbar>
10    <button ion-button menuToggle>
11      <ion-icon name="menu"></ion-icon>
12    </button>
13    <ion-title>Result</ion-title>
14  </ion-navbar>
15 </ion-header>
```

18 – *Content*

19 *Content* pada halaman *result* memiliki tag `<ion-content>` (Kode 3.19) yang pada gam-  
 20 bar 3.3d dengan kotak berwarna merah. Di dalam tag `<ion-content>` terdapat tag  
 21 `<iframe>`. Tag tersebut berisi informasi mengenai daftar pemenang acara WSDC 2017  
 22 bali yang di dapatkan pada method `onResultIframeLoad()` di kelas *ResultPage* pada file  
 23 *result.ts*.

Kode 3.19: *Content* pada result.html

```
24 <ion-content>
25   <iframe #resultIFrame (load)="onResultIframeLoad()" class="iframe-fullscreen"></iframe>
26 </ion-content>
```

- Komponen *Schedule*

30 Komponen ini digunakan untuk menampilkan halaman *Schedule* pada aplikasi. Komponen  
 31 ini memiliki sebuah file TypeScript untuk mengatur keseluruhan halaman. Di dalam file  
 32 *schedule.ts* terdapat sebuah *decorator* `@Component` untuk komponen (Kode 3.20) dan dua  
 33 *decorator* `@ViewChild` (Kode 3.21). Di dalam *decorator* ini terdapat CSS *selector* untuk  
 34 memilih CSS yang akan digunakan, serta `templateUrl` untuk mendefinisikan eksternal HTML  
 35 *template* yang akan digunakan. *Template* HTML yang digunakan adalah file *info.html*. Lalu,  
 36 `@ViewChild` digunakan untuk memanggil elemen dari DOM untuk memanggil komponen API  
 37 ke dalam TypeScript, yaitu pada komponen *result* adalah *scheduleSlider* dan *segmentContainer*  
 38 yang berada di file *result.html*. *scheduleSlider* berfungsi untuk menyimpan konten dari sebuah  
 39 *slide*. Sedangkan *segmentContainer* berfungsi untuk menyimpan konten dari sebuah *segment*.

Kode 3.20: `@Component` pada schedule.ts

```
40 @Component({
41   selector: 'page-schedule',
42   templateUrl: 'schedule.html'
43 })
44 )
```

Kode 3.21: `@ViewChild` pada schedule.ts

```
46 @ViewChild('scheduleSlider') slider: Slides;
47 @ViewChild('segmentContainer') segmentContainer: ElementRef;
```

1 Lalu, terdapat **export class** pada schedule.ts yang digunakan pada *import* di dalam app.module.ts.  
 2 Kelas ini memiliki satu *constructor*. *Constructor* kelas ini berfungsi untuk mengambil data  
 3 *result* yang berada di dalam penyimpanan internal. Data tersebut kemudian disimpan ke  
 4 dalam variabel lokal schedules. Kemudian akan mengatur *segment* yang aktif pada saat  
 5 pertama kali halaman *result* dibuka, yaitu *segment* yang pertama dan menampilkan *slide*  
 6 pertama yang berisi jadwal pada hari pertama. Selain itu, terdapat beberapa *method* yang  
 7 digunakan, yaitu:

8   – onSegmentChanged(segmentButton)

9   *Method* ini merupakan sebuah *template statement* dipanggil oleh *event* di dalam tag  
 10   <ion-segment> pada schedule.html yaitu *event* (ionChange). (ionChange) merupakan  
 11   *event* yang dimiliki oleh UI Component ion-segment milik Ionic Framework. *Method*  
 12   ini digunakan ketika pengguna memilih *segment* pada tag <ion-segment> di dalam file  
 13   schedule.html. *Method* ini memiliki sebuah parameter segmentButton yang berisi *event*  
 14   dari sebuah segment. *Child component* slides kemudian akan diubah sesuai dengan *value*  
 15   yang ada pada parameter, yaitu hari yang sedang aktif.

16   – onSlideChanged()

17   *Method* ini merupakan sebuah *template statement* dipanggil oleh *event* di dalam tag  
 18   <ion-slides> pada schedule.html yaitu *event* (ionSlideDidChange). (ionSlideDidChange)  
 19   merupakan *event* yang dimiliki oleh UI Component ion-slides milik Ionic Framework.  
 20   *Method* ini berfungsi untuk berpindah antar *slides* saat pengguna menggeser *slides*  
 21   tersebut ke kanan atau ke kiri layar.

22   – getDayName(sqlDate: string)

23   *Method* ini berfungsi untuk mengembalikan nama hari dari parameter.

24   – getDate(sqlDate: string)

25   *Method* ini bergunci untuk mengembalikan tanggal dari parameter.

26 Selain itu, terdapat file schedule.html yang digunakan untuk menampilkan halaman *schedule*.  
 27 Terdapat beberapa komponen yang disediakan oleh Ionic Framework, yang diimplementasikan  
 28 ke dalam halaman *schedule*. Diantaranya adalah sebagai berikut:

29   – *Header*

30 Halaman *schedule* memiliki *header* dengan tag <ion-header> (Kode 3.22) seperti pada  
 31 gambar 3.3c. Tag tersebut merupakan komponen *parent* yang menampung komponen  
 32 navbar yang ditandai dengan kotak berwarna biru. Di dalam navbar tersebut, terdapat  
 33 sebuah tag <button> untuk memunculkan *sidebar* dan <ion-icon> untuk menampilkan  
 34 icon dari tombol pada tag *button*. Lalu terdapat tag <ion-title> sebagai judul dari  
 35 halaman, yaitu “Schedule”.

Kode 3.22: *Header* pada schedule.html

```
36
37 1 <ion-header>
38 2   <ion-navbar>
39 3     <button ion-button menuToggle>
40 4       <ion-icon name="menu"></ion-icon>
41 5     </button>
42 6     <ion-title>Schedule</ion-title>
43 7   </ion-navbar>
44 8 </ion-header>
```

1       – *Content*

2       *Content* pada halaman result memiliki tag `<ion-content>` (Kode 3.23) yang pada gambar 3.3c dengan kotak berwarna merah. Di dalam tag `<ion-content>` terdapat dua buah tag `<div>` yang masing masing berisi tag `<ion-segment>` dan tag `<ion-slides>`. Tag `<ion-segment>` digunakan untuk tampilan hari, seperti pada gambar 3.3c yang ditandai dengan kotak berwarna hijau. Lalu, tag `<ion-slides>` digunakan untuk tampilan jadwal di dalam satu hari, seperti yang ditandai dengan kotak berwarna coklat.

3  
4  
5  
6  
7  
8       Setiap jadwal yang berada di tag `<ion-slides>` dibungkus dengan tag `<ion-list>` seperti  
9       pada kotak berwarna merah muda di gambar 3.3c. Dalam satu tag `<ion-list>` terdapat  
10      tag `<ion-note>` yang berisi waktu mulai dan waktu selesai dari satu jadwal seperti yang  
11      ditandai dengan kotak berwarna ungu, tag `<h3>` berisi nama acara seperti yang ditandai  
12      dengan kotak berwarna oranye, dan tag `<p>` yang berisi tempat acara tersebut diadakan  
13      ditandai dengan kotak berwarna biru muda.

Kode 3.23: *Content* pada schedule.html

```
14
15 <ion-content>
16   <div id="scheduleContainer">
17     <div id="scheduleSegments">
18       <ion-segment #segmentContainer *ngIf="schedules" [(ngModel)]="selectedSegmentIdx" (ionChange)="onSegmentChanged($event)">
19         <ion-segment-button *ngFor="let_schedule_of_schedules; let_i=index" [value]="i">
20           <div class="day">{{getDayName(schedule.date)}}</div>
21           <div class="date">{{getDate(schedule.date)}}</div>
22         </ion-segment-button>
23       </ion-segment>
24     </div>
25   <div id="scheduleSlides">
26     <ion-slides #scheduleSlider (ionSlideDidChange)="onSlideChanged()">
27       <ion-slide *ngFor="let_schedule_of_schedules">
28         <ion-list>
29           <ion-item text-wrap *ngFor="let_agenda_of_schedule.agenda">
30             <ion-note item-start>
31               {{agenda.start}}<br/>
32               {{agenda.end}}
33             </ion-note>
34             <h3>{{agenda.title}}</h3>
35             <p>{{agenda.subtitle}}</p>
36             </ion-item>
37           </ion-list>
38         </ion-slide>
39       </ion-slides>
40     </div>
41   </div>
42 </ion-content>
```

45       • Komponen *Venues*

46       Komponen ini digunakan untuk menampilkan halaman *Venues* pada aplikasi. Komponen ini  
47       memiliki sebuah file TypeScript untuk mengatur keseluruhan halaman. Di dalam file *venues.ts*  
48       terdapat sebuah *decorator* `@Component` untuk komponen (Kode 3.24). Di dalam decorator  
49       ini terdapat CSS *selector* untuk memilih CSS yang akan digunakan, serta `templateUrl`  
50       untuk mendefinisikan eksternal HTML *template* yang akan digunakan. *Template* HTML yang  
51       digunakan adalah file *venues.html*

Kode 3.24: `@Component` pada *venues.ts*

```
52
53   @Component({
54     selector: 'page-venues',
55     templateUrl: 'venues.html'
56   })
```

Lalu, terdapat kelas VenuesPage yang memiliki satu *constructor*. *Constructor* kelas ini berfungsi untuk mengambil data *venues* yang berada di dalam penyimpanan internal. Data tersebut kemudian disimpan ke dalam variabel lokal valVenues. Selain itu, terdapat sebuah *method* itemTapped() yang berfungsi untuk berpindah halaman ke halaman venues-map, yang akan menampilkan peta lokasi berlangsungnya acara, sesuai dengan *venues* yang dipilih. Selain itu, terdapat file venues.html yang digunakan untuk menampilkan halaman *venues*. Terdapat beberapa komponen yang disediakan oleh Ionic Framework, yang diimplementasikan ke dalam halaman *venues*. Diantaranya adalah sebagai berikut:

- *Header*

Halaman *venues* memiliki *header* dengan tag <ion-header> (Kode 3.25) seperti pada gambar 3.3a. Tag tersebut merupakan komponen *parent* yang menampung komponen navbar yang ditandai dengan kotak berwarna biru. Di dalam navbar tersebut, terdapat sebuah tag <button> untuk memunculkan *sidebar*. Lalu terdapat tag <ion-title> sebagai judul dari halaman, yaitu “Venues”.

Kode 3.25: *Header* pada venues.html

```

15
16 1<ion-header>
17 2  <ion-navbar>
18 3  <button ion-button menuToggle>
19 4    <ion-icon name="menu"></ion-icon>
20 5  </button>
21 6  <ion-title>Venues</ion-title>
22 7  </ion-navbar>
23 8 </ion-header>
```

- *Content*

*Content* pada halaman *venues* memiliki tag <ion-content> (Kode 3.26) yang pada gambar 3.3a dengan kotak berwarna merah. Di dalam tag <ion-content> terdapat sebuah tag <ion-grid> dan sebuah tag <ion-row>. Di dalam tag <ion-row> terdapat sebuah tag <ion-list> yang berisi tag <ion-button> yang ditandai dengan kotak berwarna hijau pada gambar 3.3a. Masing-masing tag <ion-button> berisi tag <ion-icon> yang ditandai dengan kotak berwarna biru muda, dan tag <span> berisi nama *venues* yang ditandai dengan kotak berwarna hitam.

Kode 3.26: *Content* pada venues.html

```

33
34 1<ion-content>
35 2  <ion-grid>
36 3  <ion-row>
37 4    <ion-list style="width: 100%;" no-lines>
38 5      <button ion-item id="{{wsdcVenue.id}}" *ngFor="let wsdcVenue of venuesData" (click)="itemTapped($event, wsdcVenue)">
39 6        <ion-icon ios="ios-{{wsdcVenue.icon}}-outline" md="md-{{wsdcVenue.icon}}" item-start></ion-icon>
40 7        <span>{{wsdcVenue.name}}</span>
41 8      </button>
42 9    </ion-list>
43 10   </ion-row>
44 11   </ion-grid>
45 12 </ion-content>
```

Lalu, terdapat kelas VenuesPage pada venues.ts. Kelas ini memiliki satu *constructor*. *Constructor* kelas ini berfungsi untuk mengambil data *venues* yang berada di dalam penyimpanan. Data tersebut kemudian disimpan ke dalam variabel lokal venuesData, yang berisi id, name, icon, geojson, dan colorIdx. Selain itu, terdapat sebuah *method* yaitu itemTapped(event, wsdcVenue). *Method* ini memiliki dua buah parameter, event yang berisi event pada tag button,

1 dan wsdcVenue yang merupakan data bertipe json yang berisi data lengkap sebuah venue yang  
 2 ada di penyimpanan sesuai dengan data venue pada *event* di dalam *tag button*. Kemudian,  
 3 dengan menggunakan NavController milik Ionic Framework, data wsdcVenue dikirimkan ke  
 4 halaman Venues Map. Setelah itu halaman akan berpindah ke halaman Venues Map.

5 • Komponen *Venues Map*

6 Komponen ini digunakan untuk menampilkan halaman *Venues Map* pada aplikasi. Berbeda  
 7 dengan komponen *venues*, komponen *Venues Map* menampilkan sebuah peta yang berisi lokasi  
 8 dari acara WSDC 2017 Bali. Komponen ini memiliki sebuah file TypeScript untuk mengatur  
 9 keseluruhan halaman. Di dalam file *venues\_map.ts* terdapat sebuah *decorator* @Component  
 10 untuk komponen (Kode 3.27). Di dalam decorator ini terdapat CSS *selector* untuk memilih CSS  
 11 yang akan digunakan, serta **templateUrl** untuk mendefinisikan eksternal HTML *template*  
 12 yang akan digunakan. *Template HTML* yang digunakan adalah file *venues\_map.html*

Kode 3.27: @Component pada *venues\_map.ts*

```
13
14
15 @Component({
16   selector: 'page-venuesmap',
17   templateUrl: 'venues_map.html',
18 })
```

19 Lalu, terdapat kelas *VenuesMapPage* di dalam *app.module.ts*. Kelas ini memiliki sebuah  
 20 *constructor*. *Constructor* kelas ini berfungsi untuk mengambil data *venues* yang berada di  
 21 dalam penyimpanan internal, kemudian disimpan ke dalam variabel lokal *venuesData*. Selain  
 22 itu, di dalam *constructor* juga mengambil data yang dikirimkan oleh halaman Venues. Data  
 23 tersebut kemudian dimasukkan ke dalam variabel lokal beranam *items*.

24 Pada komponen ini, terdapat sebuah *plugin* Google Maps, yang digunakan untuk menampilkan  
 25 peta yang berisi lokasi dari kegiatan WSDC 2017 Bali. *Plugin* tersebut diinisialisasikan di  
 26 dalam *constructor*. Selain itu, terdapat pula sebuah *plugin* Geolocation, yang berfungsi  
 27 untuk menerima masukan posisi dari lokasi pengguna yang berisi *latitude* dan *longitude*, yang  
 28 kemudian keseluruhan lokasi tersebut pada *constructor* akan dihitung jaraknya dari lokasi  
 29 pengguna saat ini.

30 Terdapat beberapa *method* yang digunakan, diantaranya yaitu:

31 – ngAfterViewInit()

32 *Method* ini dipanggil hanya sekali ketika Angular menyelesaikan inisialisasi tampilan  
 33 komponen. *Method* ini digunakan untuk menambahkan atribut ke dalam judul dari  
 34 halaman, yaitu menambahkan warna pada teks judul.

35 – loadMap()

36 *Method* ini dipanggil di dalam *constructor*, dan berfungsi untuk menampilkan peta dengan  
 37 bantuan *plugin* Google Maps. Pada *method* ini, peta pertama kali akan dibuat dengan  
 38 pengaturan kamera yang mengarah ke lokasi latitude dan longitude dari kota Kuta, Bali.  
 39 *Plugin* Google Maps sendiri akan memanfaatkan fitur-fitur *native* dari suatu perangkat.  
 40 Fitur-fitur tersebut adalah untuk melakukan *gesture* seperti *scroll*, *tilt*, *rotate*, dan *zoom*.  
 41 Lalu fitur untuk menagkses kontrol pada Google Maps, seperti mengakses kompas, tombol  
 42 lokasi pengguna saat ini, dan melihat peta di dalam ruangan. Saat Google Maps sudah  
 43 tersedia untuk digunakan, *method* ini akan memanggil *method* *loadMarkers()* untuk  
 44 membuat penanda pada peta.

1     – loadMarkers()

2       Method ini dipanggil oleh *method* loadMap(). *Method* ini digunakan untuk menampilkan  
3       *marker* dari setiap lokasi acara WSDC 2017 Bali yang sudah tersimpan di dalam variabel  
4       *items*. Google Maps menggunakan lokasi latitude dan longitude dari suatu lokasi yang  
5       berada di variabel *items* untuk membuat *marker*.

6     – featTapped(event, index)

7       Method ini merupakan sebuah *template statement* dipanggil oleh *event* di dalam *tag*  
8       <ion-item> pada *venues\_map.html* yaitu *event* (click). Saat pengguna melakukan klik  
9       di dalam *tag* <ion-item>, maka peta akan melakukan *zoom* sesuai dengan lokasi yang  
10      ada pada *tag* <ion-item>.

11    – computeDistance(p1, p2)

12      Method ini digunakan untuk menghitung jarak antara pengguna ke lokasi *venues*. Method  
13      ini memanfaatkan fitur dari *plugin* Google Maps, yaitu computeDistanceBetween dengan  
14      parameter lokasi *venues* dan lokasi perangkat pengguna yang didapatkan dari paramter  
15      *method* ini.

16      Selain itu, terdapat *file* *venues\_map.html* yang digunakan untuk menampilkan halaman  
17      *venues map*. Terdapat beberapa komponen yang disediakan oleh Ionic Framework, yang  
18      diimplementasikan ke dalam halaman. Diantaranya adalah sebagai berikut:

19     – *Header*

20      Halaman *venues* memiliki *header* dengan *tag* <ion-header> (Kode 3.28) seperti pada  
21      gambar 3.3b. *Tag* tersebut merupakan komponen *parent* yang menampung komponen  
22      navbar seperti yang ditandai dengan kotak berwarna biru. Di dalam navbar tersebut,  
23      terdapat sebuah *tag* <button> untuk memunculkan *sidebar* dan <ion-icon> untuk  
24      menampilkan icon dari tombol pada *tag* button. Lalu terdapat *tag* <ion-title> sebagai  
25      judul dari halaman, yaitu “Venues”.

Kode 3.28: *Header* pada *venues\_map.html*

```
26 1 <ion-header>
27 2   <ion-navbar>
28 3     <button ion-button menuToggle>
29 4       <ion-icon name="menu"></ion-icon>
30 5     </button>
31 6     <ion-title>Venues</ion-title>
32 7   </ion-navbar>
33 8 </ion-header>
```

36     – *Content*

37      *Content* pada halaman *venues* dibungkus oleh *tag* <ion-content> (Kode 3.29) yang pada  
38      gambar 3.3a dengan kotak berwarna merah. Di dalam *tag* <ion-content> terdapat sebuah  
39      *tag* <div> dengan id bernilai map, untuk menampilkan peta lokasi dari *venues* seperti  
40      yang ditandai dengan kotak berwarna hijau pada gambar 3.3b. Lalu untuk judul dari  
41      *venues* ditandai dengan kotak berwarna kuning dengan menggunakan *tag* <h3>. Selain  
42      itu terdapat sebuah *tag* <ion-scroll> seperti yang ditandai dengan kotak berwarna biru muda,  
43      berfungsi untuk menampilkan sebuah konten yang dapat digulir. Di dalam *tag*  
44      <ion-scroll> terdapat sebuah *tag* <ion-list> dan <ion-item> seperti yang ditandai dengan  
45      kotak berwarna ungu, berisi nama, deskripsi, serta jarak pengguna ke tempat *venues*  
46      berada. *Tag* <ion-item> akan melakukan perulangan dengan menggunakan \*ngFor yang

tersedia pada Angular. Dengan melakukan perulangan ini, akan menampilkan daftar *venues* yang tersedia, sesuai dengan yang ada pada server.

Kode 3.29: *Content* pada venues\_map.html

```
3 <ion-content>
4   <div #map id="map"></div>
5   <h3 #pagetitle>
6     {{selectedItem.name}}
7   </h3>
8   <ion-scroll scrollY="true">
9     <ion-list>
10       <ion-item text-wrap *ngFor="let feature of items; let i=index" (click)="featTapped($event,i)">
11         <h2>{{feature.name}}</h2>
12         <p>{{feature.description}}</p>
13         <ion-note item-end>
14           {{feature.distance}}
15         </ion-note>
16       </ion-item>
17     </ion-list>
18   </ion-scroll>
19 </ion-content>
```

## **3.2 Analisis Sistem Usulan**

23 Aplikasi yang ada pada saat ini menggunakan Ionic Framework versi 3, yang sudah tidak lagi  
24 didukung oleh Ionic. Maka dari itu, aplikasi WSDC 2017 Bali akan dibangun ulang menggunakan  
25 Ionic Framework versi terbaru saat ini, yaitu Ionic Framework versi 5. Proses untuk melakukan  
26 pembangunan ulang aplikasi dari Ionic Framework versi 3 ke Ionic Framework versi 5 telah dijelaskan  
27 pada sub bab 2.3.3. Pada sub bab ini akan dijelaskan analisis untuk pengembangan kebutuhan  
28 apilkasi WSDC 2017 Bali agar aplikasi tersebut dapat berjalan menggunakan Ionic Framework versi 5.

### **29 3.2.1 Analisis Kebutuhan**

30 Aplikasi WSDC 2017 Bali yang akan dibangun akan mengadopsi desain dan tata letak yang sama  
31 persis dengan aplikasi WSDC 2017 Bali saat ini. Namun dengan perubahan penggunaan Ionic  
32 Framework yang digunakan, yaitu versi 5, serta Angular versi 12. Pada Ionic Framework terbaru  
33 saat skripsi ini dibuat, aplikasi WSDC 2017 Bali yang akan dibangun akan memanfaatkan fasilitas  
34 yang disediakan oleh Ionic Framework, yaitu UI Component, dan CSS Utilities.

Struktur yang dibuat akan menggunakan struktur Ionic Framework versi 5, namun menggunakan komponen-komponen yang sama dengan Ionic Framework versi 3. Tapi, karena terdapat beberapa perubahan, maka perubahannya di dalam komponen seperti UI Component, dan CSS akan mengikuti Ionic Framework versi 5.

UI Component yang akan digunakan akan mengikuti perkembangan pada Ionic Framework versi 5. Pada setiap komponen, akan terdapat sebuah *header* dengan tag `<ion-header>`. Tag tersebut akan membungkus tag `<ion-toolbar>` sebagai *toolbar* dari aplikasi. Tag ini akan menggantikan tag `<ion-navbar>` yang telah dihapus sejak Ionic versi 4. Didalamn `<ion-toolbar>` terdapat tag `<ion-buttons>` sebagai pengganti tag `<button>` yang dihapus sejak Ionic versi 4, dan tag `<ion-title>`. Dibandingkan dengan aplikasi sistem kini, terdapat perubahan pada tag `<ion-toolbar>` yang semula bernama `<ion-navbar>` pada Ionic Framework versi 3. Selain itu ada tag `<ion-buttons>` yang semula bernama `<button>`. Di dalam tag `<ion-buttons>` akan ada sebuah tag `<ion-menu-button>`.

1 Selain itu, terdapat UI Component lain yang akan diterapkan ke dalam masing-masing komponen  
2 di dalam aplikasi yang akan dibangun, diantaranya adalah sebagai berikut :

3 • *Announcement*

4 Pada komponen *announcement*, terdapat beberapa UI Component yang akan diimplementa-  
5 sikan, diantaranya adalah sebagai berikut:

6 – *Content*

7 Komponen ini akan digunakan sebagai penyedia area konten yang digunakan untuk meng-  
8 ontrol area yang dapat digulir dan menampilkan isi konten dari halaman *announcement*.

9 UI Component *Content* dengan tag <ion-content> pada Ionic Framework versi 5 tidak  
10 mengalami perubahan dari Ionic Framework versi 3.

11 – *Refresher*

12 *Refresher* menyediakan fungsionalitas pull-to-refresh pada komponen *content*. UI Com-  
13 ponent *Refresher* dengan tag <ion-refresher> dan <ion-refresher-content> pada Ionic  
14 Framework versi 5 tidak mengalami perubahan dari Ionic Framework versi 3.

15 – *List*

16 *List* dengan tag <ion-list> akan terdiri dari beberapa baris item <ion-item> yang berisi  
17 label <ion-label>. UI Component *List* dengan tag <ion-list>, <ion-item> dan <ion-  
18 label> pada Ionic Framework versi 5 tidak mengalami perubahan dari Ionic Framework  
19 versi 3.

20 – *Item*

21 *Item* dengan tag <ion-item> sejak Ionic 4 mengalami perubahan dibandingkan pada  
22 Ionic 3, yaitu wajib menambahkan *label* dengan tag <ion-label>.

23 • *Draw*

24 Pada komponen *draw*, terdapat sebuah UI Component, yaitu *Content*. Komponen *content*  
25 akan digunakan sebagai penyedia area konten yang digunakan untuk mengontrol area yang  
26 dapat digulir dan menampilkan isi konten dari halaman *draw*. UI Component *Content* dengan  
27 tag <ion-content> pada Ionic Framework versi 5 tidak mengalami perubahan dari Ionic  
28 Framework versi 3.

29 • *Home*

30 Pada komponen *home*, terdapat beberapa file yaitu:

31 – *Content*

32 Komponen ini akan digunakan sebagai penyedia area konten yang digunakan untuk  
33 mengontrol area yang dapat digulir dan menampilkan isi konten dari halaman *home*.  
34 UI Component *Content* dengan tag <ion-content> pada Ionic Framework versi 5 tidak  
35 mengalami perubahan dari Ionic Framework versi 3.

36 – *Refresher*

37 *Refresher* menyediakan fungsionalitas pull-to-refresh pada komponen *content*. UI Com-  
38 ponent *Refresher* dengan tag <ion-refresher> dan <ion-refresher-content> pada Ionic  
39 Framework versi 5 tidak mengalami perubahan dari Ionic Framework versi 3.

1       – *Card*

2       Komponen ini akan digunakan sebagai tampilan antar muka, yang dapat menjadi titik  
3       masuk ke dalam informasi yang lebih detail. UI Component *Card* dengan tag <ion-card>,  
4       <ion-card-title> dan <ion-card-content> pada Ionic Framework versi 5 tidak mengalami  
5       perubahan dari Ionic Framework versi 3.

6       – *Grid*

7       Komponen ini akan digunakan untuk membangun tata letak kustom pada halaman *home*  
8       bagian *announcement*, yang terdiri dari baris dan kolom. UI Component *Grid* dengan tag  
9       <ion-card>, <ion-row> dan <ion-col> pada Ionic Framework versi 5 tidak mengalami  
10      perubahan dari Ionic Framework versi 3

11      – *List*

12      *List* dengan tag <ion-list> terdiri dari tag <ion-list> dan <ion-list-header> pada Ionic  
13      Framework versi 5 tidak mengalami perubahan dari Ionic Framework versi 3. Namun  
14      sejak Ionic Framework versi 4, diwajibkan untuk menambahkan tag label <ion-label> di  
15      dalam *list*.

16      – *Icon*

17      Komponen ini akan digunakan untuk menampilkan ikon pada halaman *home*. UI  
18      Component *List* dengan tag <ion-icon> pada Ionic Framework versi 5 tidak mengalami  
19      perubahan dari Ionic Framework versi 3.

20      – *Button*

21      Di dalam halaman *home*, komponen ini merupakan sebuah komponen yang dapat diklik  
22      untuk mengarahkan pengguna ke URL yang berisi berita terkait WSDC 2017 Bali. Pada  
23      Ionic Framework versi 3, komponen ini ditulis menggunakan tag <button>, lalu sejak  
24      Ionic Framework versi 4, terjadi perubahan dengan mengganti tag tersebut menjadi  
25      <ion-button>.

26      • Info

27      Pada komponen info, terdapat beberapa UI Component yang akan diimplementasikan, dian-  
28      taranya adalah sebagai berikut:

29      – *Content*

30      Komponen ini akan digunakan sebagai penyedia area konten yang digunakan untuk  
31      mengontrol area yang dapat digulir dan menampilkan isi konten dari halaman info. UI  
32      Component *Content* dengan tag <ion-content> pada Ionic Framework versi 5 tidak  
33      mengalami perubahan dari Ionic Framework versi 3.

34      – *Grid*

35      Komponen ini akan digunakan untuk membangun tata letak kustom pada halaman info,  
36      yang terdiri dari baris. UI Component *Grid* dengan tag <ion-card> dan <ion-row>  
37      pada Ionic Framework versi 5 tidak mengalami perubahan dari Ionic Framework versi 3

38      • *Result*

39      Pada komponen *Result*, terdapat sebuah UI Component, yaitu *Content*. Komponen *content* ak-  
40      an digunakan sebagai penyedia area konten yang digunakan untuk mengontrol area konten dan  
41      menampilkan isi konten dari halaman *result*. UI Component *Content* dengan tag <ion-content>  
42      pada Ionic Framework versi 5 tidak mengalami perubahan dari Ionic Framework versi 3.

1     • *Schedule*

2       Pada komponen *schedule*, terdapat beberapa UI Component yang akan diimplementasikan,  
3       diantaranya adalah sebagai berikut:

4       – *Content*

5       Komponen ini akan digunakan sebagai penyedia area konten yang digunakan untuk  
6       mengontrol area yang dapat digulir dan menampilkan isi konten dari halaman *schedule*.

7       UI Component *Content* dengan tag `<ion-content>` pada Ionic Framework versi 5 tidak  
8       mengalami perubahan dari Ionic Framework versi 3.

9       – *Item*

10      *Item* dengan tag `<ion-item>` sejak Ionic 4 mengalami perubahan dibandingkan pada  
11      Ionic 3, yaitu wajib menambahkan *label* dengan tag `<ion-label>`.

12      – *List*

13      *List* berfungsi untuk menyimpan konten yang terdiri dari beberapa baris. *List* dengan  
14      tag `<ion-list>` akan terdiri dari beberapa baris item `<ion-item>` dan akan memiliki  
15      sebuah *header*. UI Component *List* dengan tag `<ion-list>`, dan `<ion-item>` pada Ionic  
16      Framework versi 5 tidak mengalami perubahan dari Ionic Framework versi 3.

17      – *Segment*

18      Komponen ini akan digunakan untuk pengguna agar dapat berpindah tampilan di dalam  
19      halaman yang sama. Seperti pada tampilan halaman jadwal yang ada pada aplikasi  
20      WSDC 2017 Bali saat ini, dimana pengguna dapat berpindah hari untuk mengetahui  
21      jadwal kegiatan pada hari tertentu yang dipilih oleh pengguna, namun masih berada di  
22      halaman yang sama, yaitu halaman Schedule. UI Component *Segment* dengan tag `<ion-  
23      segment>` dan `<ion-segment-button>` pada Ionic Framework versi 5 tidak mengalami  
24      perubahan dari Ionic Framework versi 3.

25      – *Slides*

26      Komponen ini akan digunakan sebagai wadah dari *multi-section*. Penggunaan slide di  
27      halaman *schedule* yaitu untuk berpindah jadwal perhari dengan cara melakukan *swipe*  
28      dari kanan ke kiri layar atau sebaliknya. UI Component *Slides* dengan tag `<ion-slides>`  
29      dan `<ion-slide>` pada Ionic Framework versi 5 tidak mengalami perubahan dari Ionic  
30      Framework versi 3.

31     • *Venues*

32       Pada komponen *venues*, terdapat beberapa UI Component yang akan diimplementasikan,  
33       diantaranya adalah sebagai berikut:

34       – *Content*

35       Komponen ini akan digunakan sebagai penyedia area konten yang digunakan untuk  
36       mengontrol area yang dapat digulir dan menampilkan isi konten dari halaman *venues*.  
37       UI Component *Content* dengan tag `<ion-content>` pada Ionic Framework versi 5 tidak  
38       mengalami perubahan dari Ionic Framework versi 3.

39       – *Grid*

40       Komponen ini akan digunakan untuk membangun tata letak kustom pada halaman info.  
41       UI Component *Grid* dengan tag `<ion-card>` dan `<ion-row>` pada Ionic Framework versi  
42       5 tidak mengalami perubahan dari Ionic Framework versi 3

1       – *List*

2       *List* dengan tag `<ion-list>`, yang terdiri dari baris yang setiap barisnya berisi kategori  
3       *venues* yang disusun menggunakan *button*. UI Component *List* dengan tag `<ion-list>`  
4       pada Ionic Framework versi 5 tidak mengalami perubahan dari Ionic Framework versi 3.

5       – *Button*

6       Pada Ionic Framework versi 3, komponen ini ditulis menggunakan tag `<button>`, lalu  
7       sejak Ionic Framework versi 3, terjadi perubahan dengan mengganti tag tersebut menjadi  
8       `<ion-button>`.

9       – *Icon*

10      Komponen ini akan digunakan untuk menampilkan ikon pada halaman *venues*. UI  
11      Component *Icon* dengan tag `<ion-icon>` pada Ionic Framework versi 5 tidak mengalami  
12      perubahan dari Ionic Framework versi 3.

13      • *Venues Map*

14      Pada komponen *venues\_map*, terdapat beberapa UI Component yang akan diimplementasikan,  
15      diantaranya adalah sebagai berikut:

16       – *Content*

17       Komponen ini akan digunakan sebagai penyedia area konten yang digunakan untuk meng-  
18       ontrol area yang dapat digulir dan menampilkan isi konten dari halaman *venues\_map*.  
19       UI Component *Content* dengan tag `<ion-content>` pada Ionic Framework versi 5 tidak  
20       mengalami perubahan dari Ionic Framework versi 3.

21       – *Scroll*

22       Tag `<ion-scroll>` telah dihapus sejak Ionic Framework versi 4, dan digantikan pengguna-  
23       annya dengan menggunakan tag `<ion-content>` sejak Ionic Framework versi 4.

24       – *List*

25       *List* dengan tag `<ion-list>`, yang terdiri dari baris yang setiap barisnya berisi nama dan  
26       lokasi *venues* yang disusun menggunakan `<ion-item>`. UI Component *List* dengan tag  
27       `<ion-list>` dan `<ion-item>` pada Ionic Framework versi 5 tidak mengalami perubahan  
28       dari Ionic Framework versi 3.

29      Lalu sebagai penyedia *interface* untuk mengakses SDK *native* dan API *native* pada perangkat,  
30      pada skripsi ini akan menggunakan Capacitor dibandingkan dengan Cordova. Capacitor  
31      mengelola *plugin* dengan cara yang berbeda dibandingkan dengan Cordova, yaitu dengan cara  
32      membangun semua *plugin* sebagai sebuah *libraries* di Android, dan diinstal menggunakan  
33      *management tool* android, yaitu Gradle. Selain itu, Capacitor didukung langsung oleh Ionic,  
34      dengan pengembangan yang lebih baru dibandingkan Cordova, membuatnya dapat lebih  
35      mendukung untuk *Web Apps* modern untuk membuka fungsionalitas *native* dari platform  
36      melalui API.

Dengan digunakannya Capacitor, maka akan digunakan sebuah *plugin* yang disediakan oleh Capacitor, yaitu Capacitor Google Maps yang menggantikan *plugin* Google Maps sebelumnya yang berjalan menggunakan Cordova. *Plugin* ini berfungsi untuk menampilkan peta Google Maps secara *native* pada perangkat pengguna. Peta tersebut berisi lokasi dari seluruh acara WSDC 2017 Bali dilaksanakan yang akan ditandai dengan *marker*, dan juga akan menampilkan posisi dari perangkat pengguna. Lalu, untuk mengetahui posisi dari pengguna, akan digunakan sebuah *plugin* yang disediakan oleh Capacitor, yaitu Capacitor Geolocation. *Plugin* ini akan mendapatkan posisi *latitude* dan *longitude* perangkat saat ini dengan menggunakan Global Positioning System (GPS).

### 3.2.2 Permasalahan Pengembangan Sistem Usulan

Saat sedang melakukan proses migrasi aplikasi WSDC 2017 Bali dari Ionic Framework versi 3 ke Ionic Framework versi 5, terdapat beberapa kendala yang dialami. Kendala-kendala tersebut adalah sebagai berikut :

- Seperti yang disebutkan pada landasan teori (Sub Bab 2.3.3) sebelum melakukan migrasi dari Ionic Framework versi 3 ke Ionic Framework versi 5 terlebih dahulu melakukan migrasi dari Ionic Framework versi 3 ke Ionic Framework versi 4. Namun karena tidak tersedianya perintah untuk membuat aplikasi dengan menggunakan Ionic Framework versi 4, maka penulis langsung melakukan migrasi dari Ionic Framework versi 3 ke Ionic Framework versi 5. Dalam melakukan hal ini, penulis berlandaskan bahwa susunan kelas Ionic Framework versi 4 dan versi 5 tidaklah berubah sama sekali. Yang mengalami perubahan hanyalah pembaruan properti mengenai API, CSS, dan *package dependencies* yang terpasang, yang telah dijelaskan pada landasan teori (Sub Bab 2.3.3).
- Pada awal penggerjaan skripsi, halaman Draw dan Result pada aplikasi WSDC 2017 Bali tidak dapat diakses karena terjadi kesalahan konfigurasi pada server. Lalu setelah menghubungi dan dibantu oleh pembuat dari aplikasi WSDC 2017 Bali, maka masalah ini telah terselesaikan, yaitu halaman Draw dan Result pada aplikasi WSDC 2017 Bali dapat diakses kembali sebagaimana mestinya.
- Pada saat penggerjaan skripsi ini, pada Desember 2021 Ionic meluncurkan generasi terbaru dari Ionic Framework, yaitu Ionic versi 6. Sedangkan Ionic versi 5 pengembangannya berhenti didukung pada tanggal 8 Juni 2022. Maka dari itu, atas saran dan masukan dosen pembimbing dan dosen penguji, maka peneliti melakukan migrasi tidak lagi sampai Ionic veri 5, melainkan sampai dengan Ionic versi 6 untuk masa dukungan Ionic Framework yang lebih lama.



## DAFTAR REFERENSI

- [1] Yusuf, S. (2016) *Ionic Framework By Example*, 1st edition. Pact Publishing Ltd., Birmingham, UK.
- [2] Waranashiwar, J. dan Ukey, M. (2018) Ionic framework with angular for hybrid app development. *International Journal of New Technology and Research*, **4**, 01–02.
- [3] Wohlgethan, E. (2018) Supporting web development decisions by comparing three major javascript frameworks: Angular, react and vue.js. Thesis. Hochschule für angewandte Wissenschaften Hamburg, Germany.
- [4] Emmit A. Scott, J. (2015) *SPA Design and Architecture: Understanding single-page web applications*, 1st edition. Manning Publications, New York, USA.
- [5] World Schools Debate Championship (2021) WSDC. <https://wsdcdebate.org/history>. [Online; diakses 8-Juli-2021].
- [6] Moiseev, A. dan Fain, Y. (2018) *Angular Development with TypeScript*, 2nd edition. Manning Publications, New York, USA.
- [7] Griffith, C. (2017) *Mobile App Development with Ionic : Cross-Platform Apps with Ionic, Angular and Cordova*, 1st edition. O'Reilly Media, Inc., California, USA.
- [8] Grønli, T.-M., Biørn-Hansen, A., dan Majchrzak, T. A. (2019) Median trajectories using well-visited regions and shortest pathssoftware development for mobile computing the internet of things and wearable devices: Inspecting the past to understand the future. *Proceedings of the 52nd Hawaii International Conference on System Sciences*, Grand Wailea, Hawaii, 8–11 January, pp. 7451–7460. University of Hawaii, Manoa.
- [9] Gonsalves, M. (2018) Evaluating the mobile development frameworks apache cordova and flutter and their impact on the development process and application characteristics. Thesis. California State University, Chico, California, USA.