

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

DAFTAR ISI	iii
DAFTAR GAMBAR	v
1 PENDAHULUAN	1
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
2 LANDASAN TEORI	5
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	14
3 ANALISIS	21
3.1 Analisis Sistem Kini	21
3.1.1 Skenario Pengguna	22
3.1.2 Struktur Ionic 3	24
3.2 Analisis Sistem Usulan	42
3.2.1 Analisis Kebutuhan	42
3.2.2 Permasalahan Pengembangan Sistem Usulan	46
DAFTAR REFERENSI	47

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	22
3.2 Wireframe Aplikasi WSDC 2017 Bali	26
3.3 Wireframe Aplikasi WSDC 2017 Bali	33

¹

BAB 1

²

PENDAHULUAN

³ 1.1 Latar Belakang

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

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

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

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

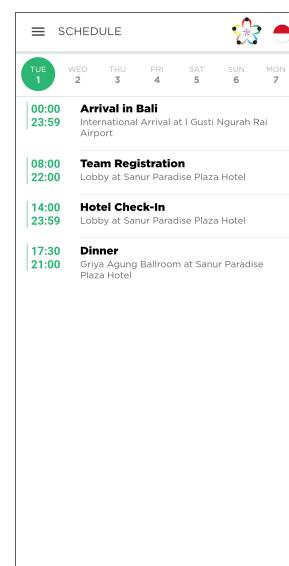
- 7 Aplikasi WSDC 2017 Bali digunakan untuk menunjang keberlangsungan acara WSDC 2017 yang dise-
8 lenggarakan di Bali, Indonesia. Aplikasi WSDC 2017 Bali dapat diunduh untuk sistem operasi *andro-*
9 *id* melalui URL <https://play.google.com/store/apps/details?id=org.wsdc2017indonesia.app>. Aplikasi 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



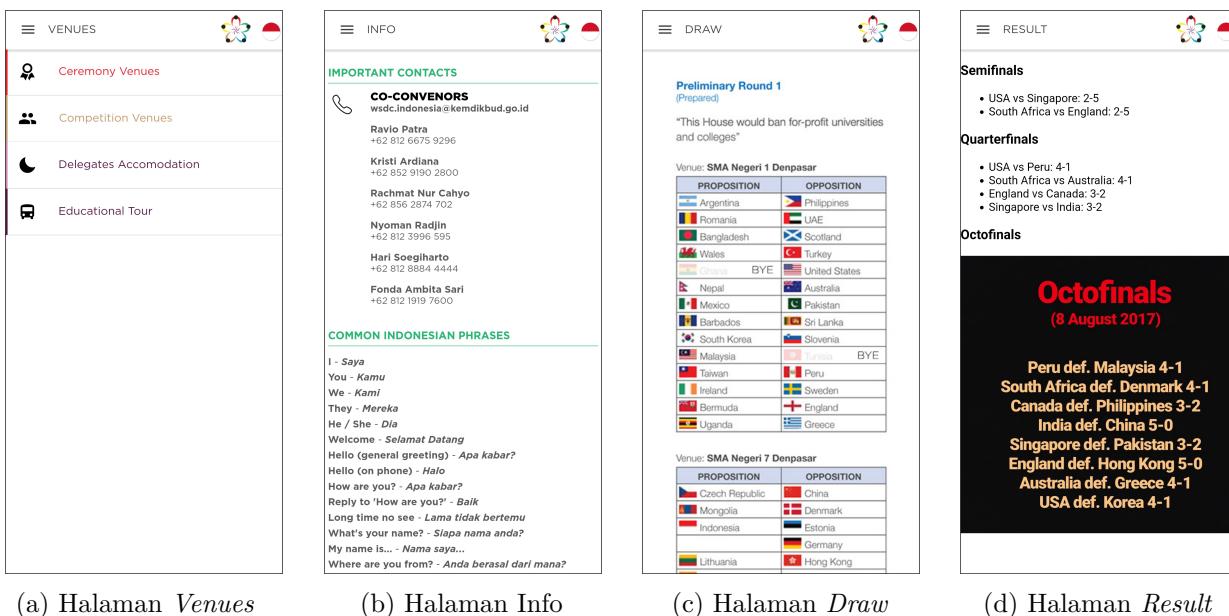
(b) Halaman *Announcements*



(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 Mereferensikan 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*-n 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 2.18), 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

```

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

```

1 <ion-item>
2   <ion-label>
3     Item
4   </ion-label>
5 </ion-item>
```

- Menu

Komponen menu merupakan panel navigasi samping yang dapat dilakukan *slides* dari sisi pada tampilan halaman saat ini menggunakan tag `<ion-menu>` (Kode 2.14). Pada dasarnya, Menu muncul dari kiri, tetapi sisi kemunculan menu dapat diganti.

Kode 2.14: Potongan Kode Program dari Menu Component

```

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

- Modal

Modal merupakan kotak dialog yang muncul diatas konten aplikasi lain, dan harus diutup secara manual oleh pengguna sebelum pengguna dapat melanjutkan menggunakan aplikasi. Modal berguna sebagai komponen pilihan ketika ada banyak opsi untuk dipilih, atau melakukan penyaringan isi di dalam daftar, serta beberapa kasus serupa lainnya (Kode 2.15).

Kode 2.15: Kode Program dari Modal

```

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

- Navigation

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

- Segment

Segment berfungsi untuk menampilkan pilihan tombol bagi pengguna untuk beralih di antara tampilan berbeda di dalam satu halaman yang sama. Segment menampilkan sekelompok tombol-tombol yang dapat diklik, dalam baris horizontal. Penggunaan Segment dengan menggunakan tag `<ion-segment>` (Kode 2.16).

Kode 2.16: Kode Program dari Segment

```

23 <ion-segment (ionChange)="segmentChanged($event)">
24   <ion-segment-button value="friends">
25     <ion-label>Friends</ion-label>
26   </ion-segment-button>
27   <ion-segment-button value="enemies">
28     <ion-label>Enemies</ion-label>
29   </ion-segment-button>
30 </ion-segment>
```

- Tabs

Tabs merupakan navigasi *top-level* yang mengimplementasi sebuah *tab-based navigation*. Tabs dapat digunakan dengan tag `<ion-tabs>` (Kode 2.17) yang tidak memiliki *styling* apapun dan bekerja sebagai *router outlet* untuk menangani navigasi.

Kode 2.17: Kode Program dari Tabs

```

37 <ion-tabs>
38   <ion-tab-bar slot="bottom">
39     <ion-tab-button tab="schedule">
40       <ion-icon name="calendar"></ion-icon>
41       <ion-label>Schedule</ion-label>
42       <ion-badge>6</ion-badge>
43     </ion-tab-button>
44
45     <ion-tab-button tab="speakers">
46       <ion-icon name="person-circle"></ion-icon>
47       <ion-label>Speakers</ion-label>
48     </ion-tab-button>
49   </ion-tab-bar>
50 </ion-tabs>
```

- Toolbar

Toolbar dapat diposisikan di atas ataupun di bawah konten. Ketika toolbar ditempatkan di header `<ion-header>` akan muncul di bagian atas konten, sedangkan ketika ditempatkan di footer `<ion-footer>` akan muncul tetap di bagian bawah. Toolbar menggunakan tag `<ion-toolbar>`, yang di dalamnya dapat berisi button, dan dapat menggunakan border (Kode 2.18).

Kode 2.18: Kode Program dari Toolbar dengan Button di Dalamnya

```

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

```

Selain komponen-komponen yang telah disebutkan, tertapat beberapa komponen lainnya yang tidak disebutkan disini. Komponen-komponen tersebut yaitu Checkbox, Chip, Floating Action Button, Grid, Icon, Input, List, Popover, Progress Indicator, Radio, Refresher, Reorder, Routing, Searchbar, Segment, Select, Slides, Toast, dan Toggle ¹.

2.3.3 Migrasi Ionic 3 ke Ionic 5

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

1. Migrasi Ionic 3 ke Ionic 4

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

- (a) Membuat Proyek Ionic Baru

Untuk membuat projek Ionic baru tanpa *template* apapun dengan menggunakan perintah **ionic start myApp blank** dan memilih Angular sebagai *frameworknya* 2.19.

Kode 2.19: Perintah Membuat Proyek Ionic Baru

```

21 1 ionic start myApp blank
22

```

- (b) Menyalin Angular Services

Menyalin Angular Services yang pada Ionic 3 berada di **src/providers**, menjadi **src/app/services** pada Ionic 4.

- (c) Menyalin *Root-level Items*

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

- i. *Imports*

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

- A. Component Imports

Untuk kepentingan konsistensi dengan *framework* lain, maka untuk mengimpor komponen Ionic diawali dengan ion 2.20.

Kode 2.20: Impor Komponen pada Ionic 4

```

37 1 import { IonInput, IonList, IonSlides } from '@ionic/angular';
38

```

- B. Package Name

Terdapat perubahan pada Ionic 4, dimana nama *package* diubah menjadi @ionic/angular. Untuk dapat menggunakan dengan cara mencopot pemasangan Ionic 3 dan memasang Ionic 4 dengan nama *package* yang baru (Kode 2.21).

¹ ‘UI Components’ <https://ionicframework.com/docs/components>, Diakses pada 14 November 2021.

Kode 2.21: Pencopotan Ionic 3 dan Pemasangan Ionic 4 dengan nama *package* baru

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

5 ii. Penamaan Berkas

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

8 A. *Page*

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

11 Pada Ionic 3 : home.html

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

13 B. *App*

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

16 Pada Ionic 3 : app.html

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

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

- 20 (e) Menyalin Bagian-bagian Aplikasi

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

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

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

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

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

35 – *Button*

36 Terdapat perbedaan pada *tag* untuk membuat Button, yang semula pada Ionic 3
37 adalah <button> menjadi <ion-button> pada Ionic 4 [2.22](#).

Kode 2.22: Penggunaan Button pada Ionic 4

```
38 1 <ion-button (click)="doSomething()">
39 2   Default Button
40 3 </ion-button>
```

43 – Floating Action Button (FAB)

44 Terdapat perbedaan pada *tag* di dalam <ion-fab>, yang semula pada Ionic 3
45 adalah <button> menjadi <ion-fab-button> pada Ionic 4 [2.23](#).

Kode 2.23: Penggunaan Floating Action Button pada Ionic 4

```

1 <ion-fab>
2   <ion-fab-button>
3     <ion-icon name="add"></ion-icon>
4   </ion-fab-button>
5   <ion-fab-list>
6     <ion-fab-button>
7       <ion-icon name="logo-facebook"></ion-icon>
8     </ion-fab-button>
9   </ion-fab-list>
10 </ion-fab>
11
12

```

– Label

Pada Ionic 4, atribut untuk mengatur posisi dari label digabungkan dengan atribut *position* [2.24](#).

Kode 2.24: Penggunaan Atribut *Position* pada Ionic 4

```

1 <ion-item>
2   <ion-label position="floating">Floating Label</ion-label>
3   <!-- input -->
4 </ion-item>

```

– 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

- content

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

Pada Ionic 3 : content

Sedangkan pada Ionic 4 menjadi : contentId

* Perubahan Nama Events Terdapat perubahan nama *events* pada Ionic 4.

Perubahan-perubahan tersebut adalah sebagai berikut :

- ionClose

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

Pada Ionic 3 : ionClose

Sedangkan pada Ionic 4 menjadi : ionDidClose

- ionOpen

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

Pada Ionic 3 : ionOpen

Sedangkan pada Ionic 4 menjadi : ionDidOpen

1 – Nav

2 Terdapat perubahan Nav pada Ionic 4. Perubahan-perubahan tersebut adalah
3 sebagai berikut :

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

- 6 · remove

7 Terdapat perubahan remove pada Ionic 4. Perubahan tersebut adalah
8 sebagai berikut :

9 Pada Ionic 3 : remove

10 Sedangkan pada Ionic 4 untuk menghindari konflik dengan HTML, berubah
11 menjadi : removeIndex

- 12 · getActiveChildNavs

13 Terdapat perubahan getActiveChildNavs pada Ionic 4. Perubahan tersebut
14 adalah sebagai berikut :

15 Pada Ionic 3 : getActiveChildNavs

16 Sedangkan pada Ionic 4 menjadi : getChildNavs

17 * Perubahan Nama Prop

18 Terdapat perubahan nama prop pada Ionic 4. Perubahan tersebut adalah
19 sebagai berikut :

20 Pada Ionic 3 : swipeBackEnabled

21 Sedangkan pada Ionic 4 menjadi : swipeGesture

22 – Navbar

23 Pada Ionic 4, terdapat penghapusan terhadap komponen <ion-navbar> karena
24 untuk menjaga agar selalu menggunakan <ion-toolbar> dengan *back button* yang
25 eksplisit [2.25](#).

Kode 2.25: Penggunaan Navbar pada Ionic 4 dengan *Back Button*

```
26 1 <ion-toolbar>
27 2   <ion-buttons slot="start">
28 3     <ion-back-button></ion-back-button>
29 4   </ion-buttons>
30 5   <ion-title>My Navigation Bar</ion-title>
31 6 </ion-toolbar>
```

34 Selain yang telah disebutkan, terdapat beberapa perubahan lainnya yang tidak
35 ditulis seperti Action Sheet, Alert, Colors, Content, Datetime, Dynamic Mode, Fixed
36 Content, Grid, Icon, Infinite Scroll, Item, Item Divider, Item Options, Item Sliding,
37 List Header, Loading, Modal, Option, Overlays, Popover, Radio, Range, Refresher,
38 Scroll, Segment Button, Select, Show When, Hide When, Spinner, Tabs, Typography,
39 Therming, Toast, dan Toolbar [2](#).

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

1 2. Migrasi Ionic 4 ke Ionic 5

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

4 • CSS

5 – CSS *Utilities*

6 Karena pada versi sebelumnya, yaitu Ionic versi 4, terdapat masalah dengan menggunakan atribut CSS dengan *framework* yang menggunakan JSX dan TypeScript, Ionic
7 *Framework* menambahkan dukungan untuk beberapa *framework*, dan pada Ionic
8 5 menambahkan kelas CSS. Ionic versi 5 menghapus atribut CSS dan mendukung
9 konsistensi. Selain itu, Ionic versi 5 juga mengubah ke kelas dengan diawali ion
10 untuk menghindari konflik dengan atribut asli dan CSS dari pengguna (Kode 2.26).

Kode 2.26: Contoh Kode Kelas CSS *Utility* pada Ionic 5

```
1 <ion-header class="ion-text-center"></ion-header>
2 <ion-content class="ion-padding"></ion-content>
3 <ion-label class="ion-text-wrap"></ion-label>
4 <ion-item classe="ion-wrap"></ion-item>
```

18 – *Display Classes*

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

21 – *Activated, Focused, Hover States*

22 Kelas .activated secara otomatis ditambahkan ke komponen yang dapat diklik,
23 mengalami perubahan nama menjadi .ion-activated. Selain itu juga memperbarui
24 komponen Action Sheet sehingga variabel akan diawali dengan *button*. Hal ini
25 dapat memungkinkan aplikasi tetap memiliki kontrol atas *opacity* jika diinginkan,
26 tetapi saat memperbarui status, hanya perlu mengatur variabel utama, yaitu -
27 background-activated, -background-focused, -background-hover. Hal tersebut penting
28 saat mengubah tema global, karena memperbarui warna *toolbar* akan secara otomatis
29 memperbarui *hover states* untuk semua *buttons* di *toolbar* (Kode 2.27).

Kode 2.27: Contoh Kode *Hover States* pada Ionic 5

```
30    /* Setting the button background on hover to solid red */
31    ion-button {
32     --background-hover: red;
33     --background-hover-opacity: 1;
34   }
35  
36   /* Setting the action sheet button background on focus to an opaque green */
37   ion-action-sheet {
38     --button-background-focus: green;
39     --button-background-focus-opacity: 0.5;
40   }
41  
42   /*
43   * Setting the fab button background on hover to match the text color with
44   * the default --background-hover-opacity on md
45   */
46   .md ion-fab-button {
47     --color: #222;
48     --background-hover: #222;
49   }
```

52 – *Distributed Scss*

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

1 • Komponen

2 Terdapat perubahan beberapa komponen pada Ionic 5, yaitu :

- 3 – Back Button dan Button

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

- 6 – Controllers

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

- 11 – Header dan Footer

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

- 14 – List Header

15 Konten berupa teks apa pun di dalam `<ion-list-header>` harus dibungkus dengan
16 `<ion-label>` sesuai dengan gaya desain yang baru (Kode 2.28). Jika label tidak ada,
17 maka perataan tombol di header bisa saja terlihat tidak aktif.

Kode 2.28: Kode Program untuk List Header

```
18
19 <ion-list-header>
20   <ion-label>New This Week</ion-label>
21   <ion-button>See All</ion-button>
22 </ion-list-header>
```

- 24 – Menu

25 Fungsi `swipeEnable()` telah dihapus di Angular, sebagai gantinya menggunakan
26 `swipeGesture()`. Lalu nilai `left` dan `right` telah dihapus, gunakan `start` dan `end`
27 sebagai gantinya. Selain itu ada penghapusan atribut utama, sebagai gantinya yaitu
28 dengan menggunakan `content-id` (untuk vanila JS atau Vue) dan `contentId` (untuk
29 Angular atau React) (Kode 2.29).

Kode 2.29: Kode Program untuk Menu

```
30
31 <ion-menu content-id="main"></ion-menu>
32 <ion-content id="main">...</ion-content>
```

- 34 – Select Option

35 Properti `selected` telah dihapus. Sebagai gantinya harus mengatur properti nilai
36 pada `ion-select` induk agar sesuai dengan opsi terpilih yang diinginkan (Kode 2.30).

Kode 2.30: Kode Program untuk Select Option

```
37
38 <ion-select value="two">
39   <ion-select-option value="one">One</ion-select-option>
40   <ion-select-option value="two">Two</ion-select-option>
41 </ion-select>
```

1 – Toast

2 Properti close button seperti showCloseButton dan closeButtonText telah dihapus.
 3 Sebagai gantinya, gunakan buttons array untuk fungsi batal (Kode 2.31).

Kode 2.31: Kode Program untuk Toast

```

4
5   async presentToast() {
6     const toast = await this.toastController.create({
7       message: 'Your settings have been saved.',
8       buttons: [
9         {
10           text: 'Close',
11           role: 'cancel',
12           handler: () => {
13             console.log('Close clicked');
14           }
15         }
16       ],
17     });
18     toast.present();
19   }

```

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

25 • Warna

26 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

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 2.32).
 33 Namun jika ingin membuat proyek baru, dapat dibuat dari CLI dan aplikasi yang ada
 34 dapat dimigrasikan secara manual.

Kode 2.32: Kode untuk Memasang *Package* dan *Dependencies* pada Angular

```

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

```

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

¹

BAB 3

²

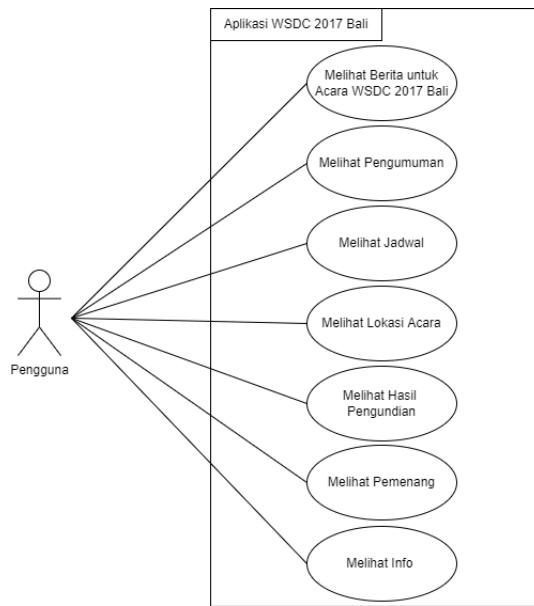
ANALISIS

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

⁶ 3.1 Analisis Sistem Kini

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

¹³ Aplikasi WSDC 2017 Bali dibangun menggunakan *framework* Ionic versi 3, dan Angular versi
¹⁴ 4.1.3. Dengan digunakannya Ionic Framework, maka memungkinkan aplikasi WSDC 2017 Bali
¹⁵ menggunakan teknologi web seperti HTML, dan CSS. Lalu untuk membangun aplikasi WSDC 2017
¹⁶ Bali agar dapat berjalan secara *native*, digunakanlah Cordova. Penggunaan Cordova memungkinkan
¹⁷ aplikasi WSDC 2017 Bali kompatibel dengan perangkat berbasis Android dan IOS, tanpa perlu
¹⁸ 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 card Announcements	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 <i>sidebar</i>	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 <i>sidebar</i>	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 <i>side bar</i>
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
15 1 @NgModule({
16 2   declarations: [
17 3     MyApp, HomePage, AnnouncementsPage, SchedulePage, VenuesPage, VenuesMapPage, DrawPage, ResultPage, InfoPage
18 4   ],
19 5   imports: [
20 6     BrowserModule, HttpModule, IonicModule.forRoot(MyApp), IonicModule.forRoot(), CloudModule.forRoot(cloudSettings)
21 7   ],
22 8   bootstrap: [IonicApp],
23 9   entryComponents: [
24 10     MyApp, HomePage, AnnouncementsPage, SchedulePage, VenuesPage, VenuesMapPage, DrawPage, ResultPage, InfoPage
25 11   ],
26 12   providers: [
27 13     StatusBar, SplashScreen, InAppBrowser, {provide: ErrorHandler, useClass: IonicErrorHandler}, Geolocation,
28 14   ]
29 15 }
30 16 )
31 17 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

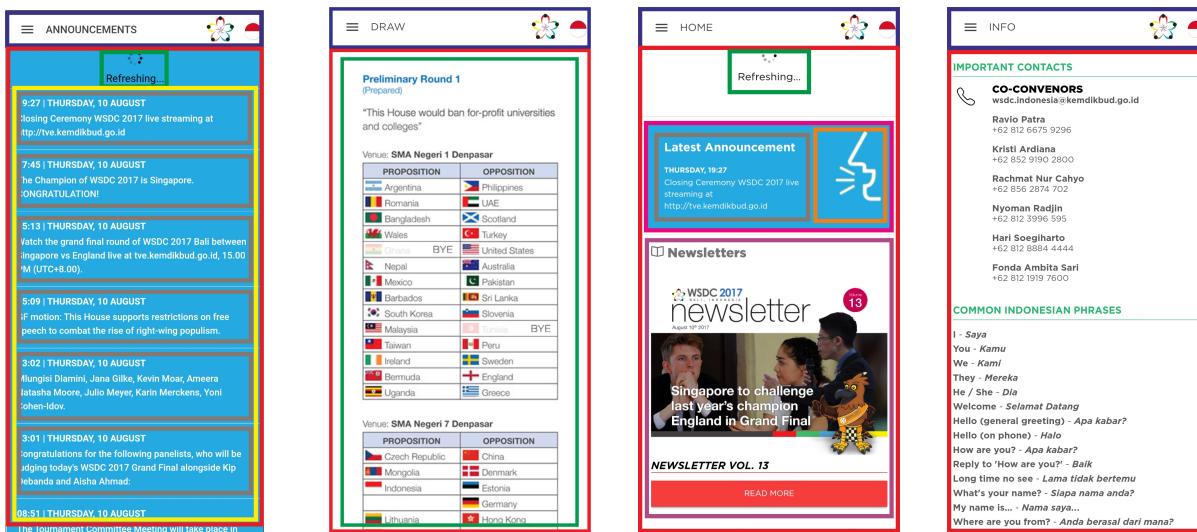
```

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 **export class** yang digunakan pada *import* di dalam app.module.ts. Kelas ini
16 berisi beberapa *method* yang akan digunakan di dalam aplikasi, diantaranya adalah sebagai
17 berikut:

18 – ionViewDidLoad()

19 *Method* ini merupakan *method* yang dipanggil ketika tampilan halaman *announcement*
20 pertama kali dimuat. *Method* berfungsi untuk memuat data *announcement* yang sudah
21 disimpan di dalam penyimpanan Ionic.

22 – doRefresh()

23 *Method* ini berfungsi untuk melakukan penyegaran ulang pada halaman *announcement* un-
24 tuk mendapatkan data *announcement* terbaru di dalam server, kemudian menyimpannya
25 ke dalam penyimpanan Ionic.

26 – presentConnectionAlert()

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

29 – formatDate()

30 *Method* ini berfungsi untuk membuat format tanggal.

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

34 – *Header*

35 Pada *header* dari halaman *announcement*, digunakan beberapa *tag* dari komponen yang
36 disediakan oleh Ionic Framework (Kode 3.4). Yaitu *tag* <ion-header> yang merupakan
37 komponen *parent* yang menampung komponen *toolbar* yang ditandai dengan warna biru
38 pada gambar 3.2a. Di dalam *tag* tersebut terdapat *tag* pendukung, seperti <ion-navbar>,
39 <button> sebagai tombol untuk membuka *sidebar*, <ion-icon> untuk menampilkan icon,
40 dan <ion-title> untuk menampilkan judul dari halaman, yaitu *Announcement*, pada
41 *navbar*.

Kode 3.4: *Header* pada halaman *Announcement*

```

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

```

– *Content*

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

Kode 3.5: *Content* pada halaman *annoncement*

```

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-list>
8     <ion-item text-wrap *ngFor="let announcement of announcements">
9       <h3>{{formatDatetime(announcement.localtime)}}</h3>
10      <p>{{announcement.message}}</p>
11    </ion-item>
12  </ion-list>
13 </ion-content>

```

- Komponen *Draw*

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

Kode 3.6: `@Component` pada *draw.ts*

```

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

```

Kode 3.7: `@ViewChild` pada *draw.ts*

```

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

```

1 Lalu, terdapat **export class** yang akan digunakan pada *import* di dalam app.module.ts. Kelas
 2 ini berisi beberapa *method* yang akan digunakan di dalam aplikasi, diantaranya adalah sebagai
 3 berikut:

4 – ionViewDidLoad()

5 *Method* ini merupakan *method* yang dipanggil ketika tampilan halaman *draw* pertama
 6 kali dimuat. *Method* ini berfungsi untuk mengambil data *draw* yang ada di dalam
 7 penyimpanan. Kemudian data yang didapatkan tersebut disimpan ke dalam *child*
 8 drawIFrame. Sampai seluruh halaman dimuat, akan dipanggil method presentLoading().

9 – presentLoading()

10 *Method* ini berfungsi untuk menampilkan teks saat halaman sedang dimuat. *Method* ini
 11 dipanggil di dalam method ionViewDidLoad().

12 – onDrawIframeLoad()

13 *Method* ini dipanggil di dalam tag <iframe> pada draw.html. *Method* ini berfungsi untuk
 14 menampilkan data yang telah diambil yang disimpan di dalam *child* drawIFrame.

15 *File* draw.html digunakan untuk menampilkan tata letak dari halaman *draw*. Terdapat
 16 beberapa komponen yang disediakan oleh Ionic Framework, yang diimplementasikan ke dalam
 17 halaman *draw*. Diantaranya adalah sebagai berikut:

18 – Header

19 *Header* dari halaman *draw* seperti pada gambar 3.2b menggunakan tag <ion-header>
 20 (Kode 3.8). Tag tersebut merupakan komponen *parent* yang menampung komponen
 21 navbar yang ditandai dengan kotak berwarna biru. Di dalam navbar tersebut, terdapat
 22 sebuah tag <button> untuk memunculkan *sidebar*. Lalu terdapat tag <ion-title> sebagai
 23 judul dari halaman.

Kode 3.8: *Header* pada draw.html

```
24<ion-header>
25  <ion-navbar>
26    <button ion-button menuToggle>
27      <ion-icon name="menu"></ion-icon>
28    </button>
29    <ion-title>Draw</ion-title>
30  </ion-navbar>
31</ion-header>
```

34 – Content

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

Kode 3.9: *Content* pada draw.html

```
40<ion-content>
41  <iframe #drawIFrame (load)="onDrawIframeLoad()" class="iframe-fullscreen"></iframe>
42</ion-content>
```

1 • Komponen *Home*

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

Kode 3.10: @Component pada home.ts

```
8
9   1 @Component({
10     2   selector: 'page-home',
11     3   templateUrl: 'home.html'
12   4 })
```

14 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 beberapa *method*, diantaranya adalah sebagai berikut:

19 – ionViewDidLoad()

20 *Method* ini merupakan *method* yang dipanggil ketika tampilan halaman *home* pertama kali dimuat. *Method* ini berfungsi untuk mengambil keseluruhan data aplikasi yang ada di dalam penyimpanan Kode 3.11). Karena halaman *home* merupakan halaman pertama ditampilkan ke dalam aplikasi, maka method ini yang akan dijalankan pertama kali. Karena itu, dengan memanfaatkan fitur *storage* yang dimiliki oleh Ionic Framework, *method* ini akan mengecek apakah sudah ada data keseluruhan aplikasi di dalam penyimpanan. Jika data tidak ditemukan, maka data akan diambil dari *file* wsdc-data.json yang kemudian dimasukan ke dalam penyimpanan internal.

Kode 3.11: Storage pada ionViewDidLoad()

```
28
29   1 this.storage.get('wsdcDataStorage').then((data) => {
30     2   if (data == null) {
31       // Default data, load from asset
32       this.http.get('assets/json/wsdc-data.json')
33         .subscribe(data => {
34           this.wsdcData = data.json();
35           this.storage.set('wsdcDataStorage', data.json());
36         },
37         err => {
38           // Error
39         });
40     } else { // Storage Data
41       this.wsdcData = data;
42     }
43   })
```

45 Lalu, *method* ini akan mengambil data dari server dengan batas waktu maksimal untuk
 46 terhubung dengan server. Jika sudah melewati batas waktu, dan aplikasi belum terhubung
 47 dengan server, maka *method* ini akan *method* showToast() yang akan menampilkan Toast
 48 yang berisi teks ‘Failed to refresh information’ (Kode 3.12). Data yang didapatkan dari
 49 server akan dimasukan ke dalam penyimpanan internal.

Kode 3.12: Perintah setTimeout() pada Home

```

1  setTimeout(() => {
2      this.http.get('http://wsdc.dnartworks.com/wsdc_data.json')
3          .timeout(7000)
4          .map(res => res.json()).subscribe(data => {
5              this.storage.set('wsdcDataStorage', data);
6              this.wsdcData = data;
7          },
8          err => {
9              this.showToast('Failed to refresh information');
10         });
11     }, 1000);

```

– launch()

Method ini digunakan untuk membuka berita yang berada di dalam url, dengan memanfaatkan *plugin* InAppBrowser yang disediakan oleh Ionic. *Method* ini dipanggil di dalam file home.html pada tag <button>. Ketika pengguna menekan tombol dengan tag <button>, tag ini mengirimkan url yang akan diterima oleh *method* launch().

– formatDatetime()

Method ini berfungsi untuk mengembalikan tanggal dengan format ‘hari, tanggal’.

– doRefresh()

Method ini berfungsi untuk memuat ulang data yang tersimpan di dalam server. *Method* ini akan melakukan pemanggilan kembali kepada server, dalam batas waktu tertentu. Jika batas waktu maksimal telah tercapai, sedangkan server belum juga memberi tanggapan, maka akan memanggil *method* showToast() yang akan menampilkan sebuah Toast yang berisi teks ‘Failed to refresh information’ (Kode 3.13). Setelah berhasil untuk terhubung dengan server, *method* ini akan menghapus data yang berada di penyimpanan *internal*, dan digantikan dengan data yang telah didapatkan dari server.

Kode 3.13: *Method* doRefresh() pada Home

```

29 // Begin async operation
30 this.http.get('http://wsdc.dnartworks.com/wsdc_data.json')
31     .timeout(30500)
32     .map(res => res.json()).subscribe(data => {
33         this.storage.clear();
34         this.storage.set('wsdcDataStorage', data);
35         this.wsdcData = data;
36         refresher.complete();
37     },
38     err => {
39         // Timeout or no connection
40         this.showToast('Failed to refresh information');
41         refresher.complete();
42     });
43
44
45 // Timeout refresher
46 setTimeout(() => {
47     refresher.complete();
48 }, 30000);

```

– showToast()

Method ini berfungsi untuk memunculkan sebuah Toast. *Method* ini menerima parameter berupa sebuah string, yang berisi pesan yang akan dimunculkan ke dalam sebuah Toast.

– onAnnouncementClick()

Method ini berfungsi untuk berpindah halaman menjadi halaman *announcement*.

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

1 – *Header*

2 Halaman *home* memiliki *header* dengan tag `<ion-header>` (Kode 3.14) seperti pada gam-
3 bar 3.2c yang ditandai dengan kotak berwarna biru. Tag tersebut merupakan komponen
4 *parent* yang menampung komponen navbar yang ditandai dengan kotak berwarna biru.
5 Di dalam navbar tersebut, terdapat sebuah tag `<button>` untuk memunculkan *sidebar*.
6 Lalu terdapat tag `<ion-title>` sebagai judul dari halaman.

Kode 3.14: *Header* pada home.html

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

17 – *Content*

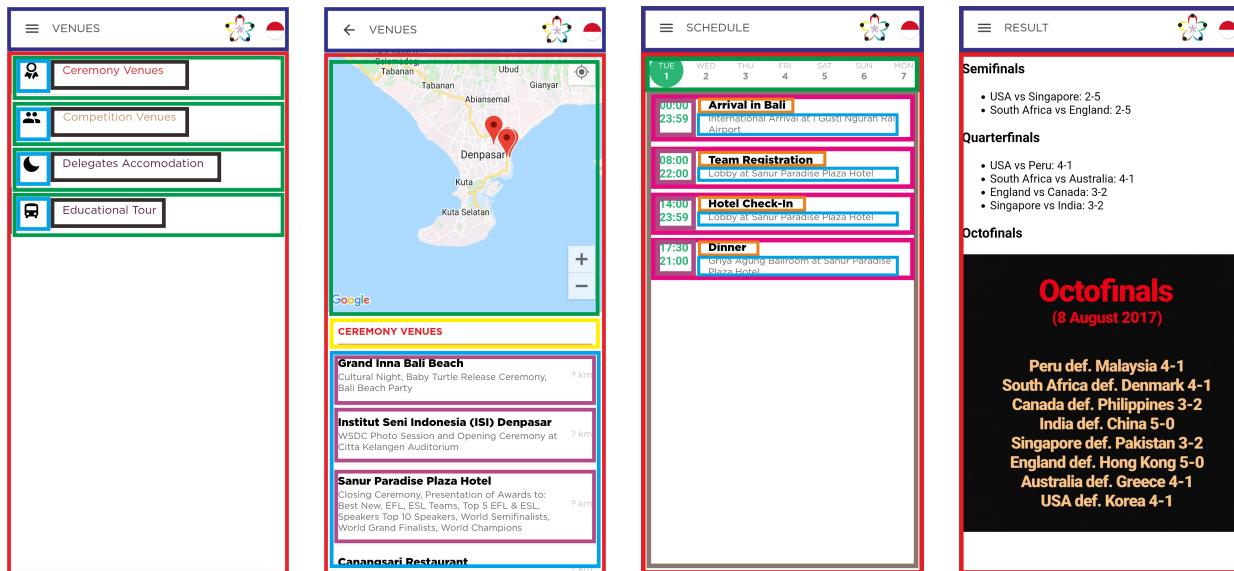
18 *Content* pada halaman *home* dengan tag `<ion-content>` (Kode 3.15 pada gambar 3.2c)
19 ditandai dengan kotak berwarna merah. Di dalam tag `<ion-content>` terdapat beberapa
20 tag lainnya. Pertama yaitu sebuah tag `<ion-refresher>` yang digunakan untuk menam-
21 pilkan simbol *refresh* saat pengguna menyegarkan halaman dengan cara melakukan *swipe*
22 dari atas ke bawah layar, ditandai dengan kotak berwarna hijau.

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

30 Lalu, terdapat sebuah tag `<ion-list>` untuk menyimpan berita-berita terkait acara WSDC
31 2017 Bali, yang ditandai dengan warna ungu. Di dalam *list* tersebut terdapat tag `<ion-
32 list-header>` sebagai judul dari *list*, dan tag `<ion-item>` untuk menyimpan berita-berita
33 terkait acara WSDC 2017 Bali. Di dalam tag `<ion-item>` terdapat tag `<button>` yang
34 apabila ditekan oleh pengguna, maka akan mengarahkan pengguna untuk melihat berita
35 tertentu sesuai dengan *item* yang dipilih dengan memanggil *method* `launch()` yang ada di
36 *home.ts*.

Kode 3.15: *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 wsdcNews of wsdcData?.newsletters">
31   <br/>
33   <h2 text-wrap>{{wsdcNews.title}}</h2>
34   <button ion-button full block color="danger" (click)="launch(wsdcNews.url)">Read More</button>
35 </ion-item>
36 </ion-list>
37 </ion-content>
```



Gambar 3.3: Wireframe Aplikasi WSDC 2017 Bali

- Komponen Info

Komponen ini digunakan untuk menampilkan halaman Info pada aplikasi. Komponen ini memiliki sebuah *file* TypeScript untuk mengatur keseluruhan halaman. Di dalam *file* info.ts terdapat sebuah *decorator* @Component untuk komponen (Kode 3.16). Di dalam decorator 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 3.16: @Component pada info.ts

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

Terdapat **export class** yang digunakan pada *import* di dalam app.module.ts. Kelas ini hanya berisi *constructor* yang digunakan untuk menginisialisai halaman yang akan digunakan. *Constructor* sendiri berfungsi untuk mengambil data info dari penyimpanan internal 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 3.17) 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*. Lalu terdapat tag <ion-title> sebagai judul dari halaman, yaitu “Info”.

Kode 3.17: *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 3.18) 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 3.18: *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>
```

- Komponen *Result* Komponen ini digunakan untuk menampilkan halaman *Result* pada aplikasi. Komponen ini memiliki sebuah *file* TypeScript untuk mengatur keseluruhan halaman. Di dalam *file* result.ts terdapat sebuah *decorator* @Component untuk komponen (Kode 3.19) dan *decorator* @ViewChild (Kode 3.24). 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* info.html. Lalu, **@ViewChild** digunakan untuk memanggil elemen dari DOM untuk meamanggil komponen API ke dalam TypeScript, yaitu pada komponen *result* adalah resultIFrame yang berada di *file* result.html.

Kode 3.19: @Component pada result.ts

```
6 1 @Component({
7 2   selector: 'page-result',
8 3   templateUrl: 'result.html'
9 4 })
```

Kode 3.20: @ViewChild pada result.ts

```
12 1 @ViewChild('resultIFrame') resultIFrame: ElementRef;
```

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

- ionViewDidLoad()

Method ini merupakan *method* yang dipanggil ketika tampilan halaman *result* pertama kali dimuat. *Method* ini berfungsi untuk memuat data *result* yang sudah disimpan di dalam penyimpanan internal. Setelah berhasil memuat data *result*, data tersebut akan dimasukan ke dalam *child* resultIFrame. Sampai seluruh halaman dimuat, akan dipanggil method presentLoading().

- presentLoading()

Method ini berfungsi untuk menampilkan teks ‘Please wait...’ saat aplikasi menunggu halaman *result* selesai dimuat.

- onResultIframeLoad()

Method ini berfungsi untuk menampilkan data *result* ke dalam halaman pada tag <iframe> yang ada di *file* result.html.

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

- *Header*

Halaman *result* memiliki *header* dengan tag <ion-header> (Kode 3.21) seperti pada gambar 3.3d. 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 “Result”.

Kode 3.21: Header pada result.html

```
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>Result</ion-title>
43 7   </ion-navbar>
44 8 </ion-header>
```

1 – *Content*

2 *Content* pada halaman result memiliki tag `<ion-content>` (Kode 3.22) yang pada gam-
 3 bar 3.3d dengan kotak berwarna merah. Di dalam tag `<ion-content>` terdapat tag
 4 `<iframe>`. Tag tersebut berisi informasi mengenai daftar pemenang acara WSDC 2017
 5 bali yang di dapatkan pada method `onResultIframeLoad()` di kelas ResultPage pada file
 6 `result.ts`.

Kode 3.22: *Content* pada `result.html`

```
7 <ion-content>
8   <iframe #resultIFrame (load)="onResultIframeLoad()" class="iframe-fullscreen"></iframe>
9 </ion-content>
```

12 • Komponen *Schedule*

13 Komponen ini digunakan untuk menampilkan halaman *Schedule* pada aplikasi. Komponen
 14 ini memiliki sebuah file TypeScript untuk mengatur keseluruhan halaman. Di dalam file
 15 `schedule.ts` terdapat sebuah *decorator* `@Component` untuk komponen (Kode 3.23) dan dua
 16 *decorator* `@ViewChild` (Kode 3.24). Di dalam *decorator* ini terdapat CSS *selector* untuk
 17 memilih CSS yang akan digunakan, serta `templateUrl` untuk mendefinisikan ekxternal HTML
 18 *template* yang akan digunakan. *Template* HTML yang digunakan adalah file `info.html`. Lalu,
 19 `@ViewChild` digunakan untuk memanggil elemen dari DOM untuk meamanggil komponen API
 20 ke dalam TypeScript, yaitu pada komponen *result* adalah `scheduleSlider` dan `segmentContainer`
 21 yang berada di file `result.html`. `scheduleSlider` berfungsi untuk menyimpan konten dari sebuah
 22 *slide*. Sedangkan `segmentContainer` berfungsi untuk menyimpan konten dari sebuah *segment*.

Kode 3.23: `@Component` pada `schedule.ts`

```
23 @Component({
24   selector: 'page-schedule',
25   templateUrl: 'schedule.html'
26 })
27 }
```

Kode 3.24: `@ViewChild` pada `schedule.ts`

```
29 @ViewChild('scheduleSlider') slider: Slides;
30 @ViewChild('segmentContainer') segmentContainer: ElementRef;
```

33 Lalu, terdapat **export class** pada `schedule.ts` yang digunakan pada `import` di dalam `app.module.ts`.
 34 Kelas ini memiliki satu *constructor*. *Constructor* kelas ini berfungsi untuk mengambil data
 35 *result* yang berada di dalam penyimpanan internal. Data tersebut kemudian disimpan ke
 36 dalam variabel lokal `schedules`. Kemudian akan mengatur *segment* yang berfungsi pada saat
 37 pertama kali halaman *result* dibuka, yaitu *segment* yang pertama dan menampilkan *slide*
 38 pertama yang berisi jadwal pada hari pertama. Selain itu, terdapat beberapa method yang
 39 digunakan, yaitu:

40 – `onSegmentChanged()`

41 *Method* ini digunakan ketika pengguna memilih *segment* pada tag `<ion-segment>` di
 42 dalam file `result.html`, lalu *method* ini akan mengubah *slide* yang aktif pada tag `<ion-
 43 slide>` sesuai dengan hari pada *segment* yang sedang aktif.

44 – `onSlideChanged()`

45 *Method* ini berfungsi untuk berpindah antar *slides* yang ada pada tag `<ion-slides>` di file
 46 `schedule.html`. Tag tersebut akan berisi sebuah jadwal yang ada di dalam suatu hari.

- 1 – getDayName()
- 2 *Method* ini berfungsi untuk mengembalikan nama hari.
- 3 – getDate()
- 4 *Method* ini bergunsi untuk mengembalikan tanggal.

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

- 8 – *Header*

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

Kode 3.25: *Header* pada schedule.html

```
14 <ion-header>
15   <ion-navbar>
16     <button ion-button menuToggle>
17       <ion-icon name="menu"></ion-icon>
18     </button>
19     <ion-title>Schedule</ion-title>
20   </ion-navbar>
21 </ion-header>
```

- 24 – *Content*

25 *Content* pada halaman result memiliki *tag* <ion-content> (Kode 3.26) yang pada gam-
 26 bar 3.3c dengan kotak berwarna merah. Di dalam *tag* <ion-content> terdapat dua buah
 27 *tag* <div> yang masing masing berisi *tag* <ion-segment> dan *tag* <ion-slides>. *Tag*
 28 <ion-segment> digunakan untuk tampilan hari, seperti pada gambar 3.3c yang ditandai
 29 dengan kotak berwarna hijau. Lalu, *tag* <ion-slides> digunakan untuk tampilan jadwal
 30 di dalam satu hari, seperti yang ditandai dengan kotak berwarna coklat.

31 Setiap jadwal yang berada di *tag* <ion-slides> dibungkus dengan *tag* <ion-list> seperti
 32 pada kotak berwarna muda di gambar 3.3c. Dalam satu *tag* <ion-list> terdapat
 33 *tag* <ion-note> yang berisi waktu mulai dan waktu selesai dari satu jadwal seperti yang
 34 ditandai dengan kotak berwarna ungu, *tag* <h3> berisi nama acara seperti yang ditandai
 35 dengan kotak berwarna oranye, dan *tag* <p> yang berisi tempat acara tersebut diadakan
 36 ditandai dengan kotak berwarna biru muda.

Kode 3.26: Content pada schedule.html

```

1 <ion-content>
2   <div id="schedulesContainer">
3     <div id="schedulesSegments">
4       <ion-segment #segmentContainer *ngIf="schedules" [(ngModel)]="selectedSegmentIdx" (ionChange)="onSegmentChanged($event)">
5         <ion-segment-button *ngFor="let _schedule_of_schedules; let i = index" [value]="i">
6           <div class="day">{{getDayName(schedule.date)}}</div>
7           <div class="date">{{getDate(schedule.date)}}</div>
8         </ion-segment-button>
9       </ion-segment>
10      </div>
11    <div id="schedulesSlides">
12      <ion-slides #scheduleSlider (ionSlideDidChange)="onSlideChanged()">
13        <ion-slide *ngFor="let _schedule_of_schedules">
14          <ion-list>
15            <ion-item text-wrap *ngFor="let agenda_of_schedule.agenda">
16              <ion-note item-start>
17                {{agenda.start}}<br/>
18                {{agenda.end}}
19              </ion-note>
20              <h3>{{agenda.title}}</h3>
21              <p>{{agenda.subtitle}}</p>
22            </ion-item>
23          </ion-list>
24        </ion-slide>
25      </ion-slides>
26    </div>
27  </div>
28 </ion-content>

```

- Komponen *Venues*

Komponen ini digunakan untuk menampilkan halaman *Venues* pada aplikasi. Komponen ini memiliki sebuah *file* TypeScript untuk mengatur keseluruhan halaman. Di dalam *file* venues.ts terdapat sebuah *decorator* @Component untuk komponen (Kode 3.27). 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* venues.html

Kode 3.27: @Component pada venues.ts

```

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

```

Lalu, terdapat **export class** pada venues.ts yang digunakan pada *import* di dalam app.module.ts. Kelas ini 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:

1 – *Header*

2 Halaman *venues* memiliki *header* dengan tag `<ion-header>` (Kode 3.28) seperti pada
 3 gambar 3.3a. 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*. Lalu terdapat tag `<ion-title>` sebagai
 6 judul dari halaman, yaitu “Venues”.

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

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

17 – *Content*

18 *Content* pada halaman *venues* memiliki tag `<ion-content>` (Kode 3.29) yang pada
 19 gambar 3.3a dengan kotak berwarna merah. Di dalam tag `<ion-content>` terdapat
 20 sebuah tag `<ion-grid>` dan sebuah tag `<ion-row>`. Di dalam tag `<ion-row>` terdapat
 21 sebuah tag `<ion-list>` yang berisi tag `<ion-button>` yang ditandai dengan kotak berwarna
 22 hijau pada gambar 3.3a. Masing-masing tag `<ion-button>` berisi tag `<ion-icon>` yang
 23 ditandai dengan kotak berwarna biru muda, dan tag `` berisi nama *venues* yang
 24 ditandai dengan kotak berwarna hitam.

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

```
25 <ion-content>
26   <ion-grid>
27     <ion-row>
28       <ion-list style="width: 100%;" no-lines>
29         <button ion-item id="{{wsdcVenue.id}}" *ngFor="let wsdcVenue of venuesData" (click)="itemTapped($event, wsdcVenue)">
30           <ion-icon ios="ios-{{wsdcVenue.icon}}-outline" md="md-{{wsdcVenue.icon}}" item-start></ion-icon>
31           <span>{{wsdcVenue.name}}</span>
32         </button>
33       </ion-list>
34     </ion-row>
35   </ion-grid>
36 </ion-content>
```

40 • Komponen *Venues Map*

41 Komponen ini digunakan untuk menampilkan halaman *Venues Map* pada aplikasi. Berbeda
 42 dengan komponen *venues*, komponen *Venues Map* menampilkan sebuah peta yang berisi lokasi
 43 dari acara WSDC 2017 Bali. Komponen ini memiliki sebuah file TypeScript untuk mengatur
 44 keseluruhan halaman. Di dalam file *venues_map.ts* terdapat sebuah *decorator* `@Component`
 45 untuk komponen (Kode 3.30). Di dalam *decorator* ini terdapat CSS *selector* untuk memilih CSS
 46 yang akan digunakan, serta `templateUrl` untuk mendefinisikan eksternal HTML *template*
 47 yang akan digunakan. Template HTML yang digunakan adalah file *venues_map.html*

Kode 3.30: `@Component` pada *venues_map.ts*

```
48 @Component({
49   selector: 'page-venuesmap',
50   templateUrl: 'venues_map.html',
51 })
52 })
```

1 Lalu, terdapat **export class** pada *venues_map.ts* yang digunakan pada *import* di dalam
2 *app.module.ts*. Kelas ini memiliki satu *constructor*. *Constructor* kelas ini berfungsi untuk
3 mengambil data *venues* yang berada di dalam penyimpanan internal. Data tersebut kemudian
4 disimpan ke dalam variabel lokal *venuesData*.

5 Pada komponen ini, terdapat sebuah *plugin* Google Maps, yang digunakan untuk menampilkan
6 peta yang berisi lokasi dari kegiatan WSDC 2017 Bali. *Plugin* tersebut diinisialisasikan di
7 dalam *constructor*. Selain itu, terdapat pula sebuah *plugin* Geolocation, yang berfungsi untuk
8 menerima masukan posisi dari suatu lokasi yang berisi *latitude* dan *longitude*, yang kemudian
9 keseluruhan lokasi tersebut pada *constructor* akan dihitung jaraknya dari lokasi pengguna
10 saat ini. Lalu terdapat beberapa *method* yang digunakan, diantaranya yaitu:

11 – *ngAfterViewInit()*

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

15 – *loadMap()*

16 *Method* ini digunakan saat pertama kali halaman *venues map* dibuka. *Method* ini
17 dipanggil di dalam *constructor*, dan berfungsi untuk menampilkan peta dengan bantuan
18 *plugin* Google Maps. Pada *method* ini, peta akan menyimpan lokasi spesifik pada *latitude*
19 dan *longitude* tertentu dari suatu lokasi acara WSDC 2017 Bali. Lalu dengan bantuan
20 *plugin* Google Maps, peta akan menampilkan peta pulau bali, dengan beberapa lokasi
21 *venues* yang ada. Lokasi *venues* tersebut, akan ditandai dengan sebuah *marker* yang
22 akan dibuat oleh *method* *loadMarkers()*.

23 *Plugin* Google Maps sendiri akan memanfaatkan fitur-fifur *native* dari suatu perangkat.
24 Fitur-fitur tersebut adalah untuk melakukan *gesture* seperti *scroll*, *tilt*, *rotate*, dan *zoom*.
25 Lalu fitur untuk menagkses kontrol pada Google Maps, seperti mengakses kompas, lokasi
26 pengguna saat ini, dan melihat peta di dalam ruangan.

27 – *loadMarkers()*

28 *Method* ini dipanggil oleh *method* *loadMap()*. *Method* ini digunakan untuk menampilkan
29 *marker* dari setiap lokasi acara WSDC 2017 Bali yang sudah tersimpan di dalam peta
30 pada *plugin* Google Maps.

31 – *featTapped()*

32 *Method* ini digunakan untuk melakukan *zoom* pada peta terhadap satu lokasi yang dipilih
33 oleh pengguna.

34 – *computeDistance()*

35 *Method* ini digunakan untuk menghitung jarak antara pengguna ke lokasi *venues*. *Method*
36 ini memanfaatkan fitur dari *plugin* Google Maps, yaitu *computeDistanceBetween* dengan
37 parameter lokasi *venues* dan lokasi perangkat pengguna.

38 Selain itu, terdapat *file* *venues_map.html* yang digunakan untuk menampilkan halaman
39 *venues map*. Terdapat beberapa komponen yang disediakan oleh Ionic Framework, yang
40 diimplementasikan ke dalam halaman. Diantaranya adalah sebagai berikut:

1 – *Header*

2 Halaman *venues* memiliki *header* dengan tag `<ion-header>` (Kode 3.31) seperti pada
 3 gambar 3.3b. Tag tersebut merupakan komponen *parent* yang menampung komponen
 4 navbar seperti yang ditandai dengan kotak berwarna biru. Di dalam navbar tersebut,
 5 terdapat sebuah tag `<button>` untuk memunculkan *sidebar*. Lalu terdapat tag `<ion-title>`
 6 sebagai judul dari halaman, yaitu “Venues”.

Kode 3.31: *Header* pada venues_map.html

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

17 – *Content*

18 *Content* pada halaman *venues* dibungkus oleh tag `<ion-content>` (Kode 3.32) yang pada
 19 gambar 3.3a dengan kotak berwarna merah. Di dalam tag `<ion-content>` terdapat sebuah
 20 tag `<div>` dengan id bernilai *map*, untuk menampilkan peta lokasi dari *venues* seperti
 21 yang ditandai dengan kotak berwarna hijau pada gambar 3.3b. Lalu untuk judul dari
 22 *venues* ditandai dengan kotak berwarna kuning dengan menggunakan tag `<h3>`. Selain
 23 itu terdapat sebuah tag `<ion-scroll>` seperti yang ditandai dengan kotak berwarna biru
 24 muda, berfungsi untuk menampilkan sebuah konten yang dapat digulir. Di dalam tag
 25 `<ion-scroll>` terdapat sebuah tag `<ion-list>` dan `<ion-item>` seperti yang ditandai dengan
 26 kotak berwarna ungu, berisi nama, deskripsi, serta jarak pengguna ke tempat *venues*
 27 berada. Tag `<ion-item>` akan melakukan perulangan dengan menggunakan `*ngFor` yang
 28 tersedia pada Angular. Dengan melakukan perulangan ini, akan menampilkan daftar
 29 *venues* yang tersedia, sesuai dengan yang ada pada server.

Kode 3.32: *Content* pada venues_map.html

```
30 <ion-content>
31   <div #map id="map"></div>
32   <h3 #pagetitle>
33     {{selectedItem.name}}
34   </h3>
35   <ion-scroll scrollY="true">
36     <ion-list>
37       <ion-item text-wrap *ngFor="let_feature_of_items; let_i=index" (click)="featTapped($event,i)">
38         <h2>{{feature.name}}</h2>
39         <p>{{feature.description}}</p>
40         <ion-note item-end>
41           {{feature.distance}}
42         </ion-note>
43       </ion-item>
44     </ion-list>
45   </ion-scroll>
46 </ion-content>
```

3.2 Analisis Sistem Usulan

Aplikasi yang ada pada saat ini menggunakan Ionic Framework versi 3, yang sudah tidak lagi didukung oleh Ionic. Maka dari itu, aplikasi WSDC 2017 Bali akan dibangun ulang menggunakan Ionic Framework versi terbaru saat ini, yaitu Ionic Framework versi 5. Proses untuk melakukan pembangunan ulang aplikasi dari Ionic Framework versi 3 ke Ionic Framework versi 5 telah dijelaskan pada sub bab [2.3.3](#). Pada sub bab ini akan dijelaskan analisis untuk pengembangan kebutuhan apilksi WSDC 2017 Bali agar aplikasi tersebut dapat berjalan menggunakan Ionic Framework versi 5.

3.2.1 Analisis Kebutuhan

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

Struktur yang akan dibuat, menggunakan struktur Ionic Framework versi 5, namun menggunakan komponen-komponen yang sama dengan Ionic Framework versi 3. Tapi, karena terdapat beberapa perubahan, maka perubahan 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, yang didalamnya terdapat tag `<ion-buttons>` 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>`.

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

- *Announcement*

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

- *Content*

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

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

- *Refresher*

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

1 – *List*

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

6 • *Draw*

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

12 • *Home*

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

14 – *Content*

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

19 – *Refresher*

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

23 – *Card*

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

28 – *Grid*

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

33 – *List*

34 *List* dengan tag `<ion-list>` akan terdiri dari beberapa baris item `<ion-item>` dan akan
35 memiliki sebuah *header*. UI Component *List* dengan tag `<ion-list>`, `<ion-item>`, `<ion-`
36 `<ion-label>` dan `<ion-list-header>` pada Ionic Framework versi 5 tidak mengalami perubahan
37 dari Ionic Framework versi 3.

38 – *Icon*

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

1 – *Button*

2 Di dalam halaman *home*, komponen ini merupakan sebuah komponen yang dapat diklik
3 untuk mengarahkan pengguna ke URL yang berisi berita terkait WSDC 2017 Bali. Pada
4 Ionic Framework versi 3, komponen ini ditulis menggunakan *tag* <button>, lalu pada
5 Ionic Framework versi 5, terjadi perubahan dengan mengganti *tag* tersebut menjadi
6 <ion-button>.

7 • Info

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

10 – Content

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

15 – Grid

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

19 • Result

20 Pada komponen *Result*, terdapat sebuah UI Component, yaitu *Content*. Komponen *content*
21 akan digunakan sebagai penyedia area konten yang digunakan untuk mengontrol area konten
22 dan menampilkan isi konten dari halaman *result*. UI Component *Content* dengan *tag* <ion-
23 content> pada Ionic Framework versi 5 tidak mengalami perubahan dari Ionic Framework
24 versi 3.

25 • Schedule

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

28 – Content

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

33 – Segment

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

1 – *Slides*

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

7 – *List*

8 *List* berfungsi untuk menyimpan konten yang terdiri dari beberapa baris. *List* dengan
9 tag <ion-list> akan terdiri dari beberapa baris item <ion-item> dan akan memiliki
10 sebuah *header*. UI Component *List* dengan tag <ion-list>, <ion-item>, <ion-label> dan
11 <ion-list-header> pada Ionic Framework versi 5 tidak mengalami perubahan dari Ionic
12 Framework versi 3.

13 • *Venues*

14 Pada komponen *venues*, 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
18 mengontrol area yang dapat digulir dan menampilkan isi konten dari halaman *venues*.
19 UI Component *Content* dengan tag <ion-content> pada Ionic Framework versi 5 tidak
20 mengalami perubahan dari Ionic Framework versi 3.

21 – *Grid*

22 Komponen ini akan digunakan untuk membangun tata letak kustom pada halaman info.
23 UI Component *Grid* dengan tag <ion-card> dan <ion-row> pada Ionic Framework versi
24 5 tidak mengalami perubahan dari Ionic Framework versi 3

25 – *List*

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

29 – *Button*

30 Di dalam halaman *venues*, komponen ini merupakan sebuah komponen yang dapat diklik
31 untuk mengarahkan pengguna ke halaman *venues map* sesuai dari kategori *venues* yang
32 dipilih. Pada Ionic Framework versi 3, komponen ini ditulis menggunakan tag <button>,
33 lalu pada Ionic Framework versi 5, terjadi perubahan dengan mengganti tag tersebut
34 menjadi <ion-button>.

35 – *Icon*

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

1 • *Venues Map*

2 Pada komponen *venues_map*, 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 meng-
6 ontrol area yang dapat digulir dan menampilkan isi konten dari halaman *venues_map*.

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

9 – *List*

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

14 Selain itu, akan digunakan sebuah *plugin* yang disediakan oleh Capacitor, yaitu Capacitor
15 Google Maps. *Plugin* ini berfungsi untuk menampilkan peta Google Maps secara *native*
16 pada perangkat pengguna. Lalu, untuk mengetahui posisi dari pengguna, akan digunakan
17 sebuah *plugin* yang disediakan oleh Capacitor, yaitu Capacitor Geolocation. *Plugin* ini akan
18 mendapatkan dan melacak posisi perangkat saat ini saat menggunakan Global Positioning
19 System (GPS).

20 **3.2.2 Permasalahan Pengembangan Sistem Usulan**

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

- 24 • Seperti yang disebutkan pada landasan teori (Sub Bab 2.3.3) sebelum melakukan migrasi
25 dari Ionic Framework versi 3 ke Ionic Framework versi 5 terlebih dahulu melakukan migrasi
26 dari Ionic Framework versi 3 ke Ionic Framework versi 4. Namun karena tidak tersedianya
27 perintah untuk membuat aplikasi dengan menggunakan Ionic Framework versi 4, maka penulis
28 langsung melakukan migrasi dari Ionic Framework versi 3 ke Ionic Framework versi 5. Dalam
29 melakukan hal ini, penulis berlandaskan bahwa susunan kelas Ionic Framework versi 4 dan
30 versi 5 tidaklah berubah sama sekali. Yang mengalami perubahan hanyalah pembaruan
31 properti mengenai API, CSS, dan *package dependencies* yang terpasang, yang telah dijelaskan
32 pada landasan teori (Sub Bab 2.3.3).
33 • Pada awal pengerjaan skripsi, halaman Draw dan Result pada aplikasi WSDC 2017 Bali tidak
34 dapat diakses karena terjadi kesalahan konfigurasi pada server. Lalu setelah menghubungi dan
35 dibantu oleh pembuat dari aplikasi WSDC 2017 Bali, maka masalah ini telah terselesaikan,
36 yaitu halaman Draw dan Result pada aplikasi WSDC 2017 Bali dapat diakses kembali
37 sebagaimana mestinya.

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.