

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	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 WSDC 2017 Bali	5
2.2 Ionic <i>Framework</i>	6
2.2.1 Native API	7
2.2.2 UI Component	8
2.2.3 Migrasi Ionic 3 ke Ionic 5	12
DAFTAR REFERENSI	19
A KODE PROGRAM	21
B HASIL EKSPERIMEN	23

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
B.1 Hasil 1	23
B.2 Hasil 2	23
B.3 Hasil 3	23
B.4 Hasil 4	23

¹

BAB 1

²

PENDAHULUAN

³ 1.1 Latar Belakang

⁴ *World Schools Debating Championships* (WSDC) merupakan sebuah turnamen debat bahasa Inggris
⁵ tahunan untuk tim-tim tingkat sekolah menengah yang mewakili berbagai negara ¹. Pada awalnya,
⁶ kompetisi universitas dunia akan diselenggarakan di Sydney pada bulan Juli 1988. Anggota Federasi
⁷ Debat Australia menyadari bahwa tidak ada acara serupa untuk siswa sekolah menengah. Namun
⁸ kejuaraan universitas dunia ini menunjukkan potensi yang sangat besar untuk kompetisi debat
⁹ internasional yang melibatkan siswa dari seluruh dunia. Pada tahun 1991, kejuaraan diadakan
¹⁰ di Edinburgh. Dan sejak saat itu nama World Schools Debating Championships digunakan dan
¹¹ berlangsung hingga saat ini.

¹² Ionic merupakan sebuah kerangka kerja *open source* lintas platform yang memungkinkan untuk
¹³ mengembangkan aplikasi hibrida yang bekerja pada berbagai macam platform seluler seperti *android*,
¹⁴ *iOS*, dan *Windows* [1]. Ionic memiliki berbagai macam *front-end library* dan *User Interface*(UI)
¹⁵ *Components* yang digunakan untuk perancangan aplikasi menggunakan teknologi web seperti *HTML*,
¹⁶ *Cascading Style Sheets* *CSS*, dan *Javascript*.

¹⁷ Pada Ionic 5, terdapat beberapa kerangka *Javascript* yang dapat diimplementasikan menggu-
¹⁸ nakan *framework* Ionic, seperti *Angular*, *React*, dan *Vue*. *Angular* pada awalnya diciptakan oleh
¹⁹ karyawan Google, Misko Hevert dan Adam Abrons pada tahun 2008, yang masih bernama *AngularJS*
²⁰ dan dikembangkan dalam *JavaScript* [2]. Pada saat itu sebagian besar situs web menggunakan
²¹ aplikasi multi-halaman, yaitu ketika pengguna mengklik tautan, maka browser harus mengambil
²² dokumen *HTML* yang diminta dari server. *React* adalah *library* *JavaScript* *open source* untuk
²³ membangun antarmuka pengguna, dikelola oleh Facebook, dapat digunakan dalam berbagai skenario
²⁴ termasuk aplikasi *iOS* dan *Android* [2]. Sedangkan *Vue* merupakan *framework* progresif untuk
²⁵ membangun antarmuka pengguna untuk web, yang dapat digunakan baik untuk projek kecil dan
²⁶ untuk *Single-Page Applications* (SPAs) [2].

²⁷ WSDC yang diselenggarakan di Bali, Indonesia pada tahun 2017 memiliki sebuah aplikasi
²⁸ bernama WSDC 2017 Bali yang dikembangkan oleh PT DNArtworks menggunakan *framework*
²⁹ Ionic 3 untuk menunjang acara tersebut. Terdapat beberapa fungsi penting di dalam aplikasi
³⁰ ini, diantaranya adalah jadwal untuk kegiatan peserta, berita tentang acara WSDC yang sedang
³¹ berlangsung, pemberitahuan mengenai kegiatan acara kepada peserta, informasi lokasi dan penunjuk
³² arah ke lokasi kegiatan acara yang sedang berlangsung, dan notifikasi untuk peserta.

¹ ‘WSDC’ <https://wsdcdebate.org/history>, Diakses pada 8 Juli 2021.

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

7 **1.2 Rumusan Masalah**

8 Rumusan masalah yang akan dibahas pada skripsi ini adalah sebagai berikut :
9 • Fitur-fitur apa yang akan tersedia di aplikasi WSDC terbaru?
10 • Bagaimana membangun aplikasi *android* WSDC menggunakan *framework* Ionic versi 5?
11 • Bagaimana melakukan migrasi Ionic versi 3 ke Ionic versi 5?

12 **1.3 Tujuan**

13 Tujuan yang ingin dicapai dari penulisan skripsi ini adalah sebagai berikut :
14 • Mendefinisikan fitur-fitur yang akan tersedia di aplikasi WSDC terbaru.
15 • Membangun aplikasi *android* WSDC menggunakan *framework* Ionic versi 5.
16 • Melakukan migrasi Ionic versi 3 ke Ionic versi 5.

17 **1.4 Batasan Masalah**

18 Dalam skripsi ini dibuat batasan-batasan masalah dalam pembuatan perangkat lunak. Batasan-
19 batasan masalah yang ditetapkan adalah sebagai berikut:
20 1. Aplikasi ini tidak akan memiliki fitur notifikasi, karena acara WSDC 2017 Bali sudah selesai
21 dan tidak diperlukan kembali fitur notifikasi.
22 2. Aplikasi hanya akan berjalan pada *platform mobile* berbasis android.

23 **1.5 Metodologi**

24 Langkah-langkah yang dilakukan dalam skripsi ini adalah:
25 1. Melakukan studi mengenai *framework* Ionic versi 3 dan versi 5.
26 2. Menganalisis aplikasi WSDC 2017 Bali.
27 3. Mempelajari bagaimana cara melakukan migrasi Ionic versi 3 ke versi 5.
28 4. Mendesain kelas aplikasi.
29 5. Membangun aplikasi WSDC dengan *framework* Ionic versi 5.
30 6. Melakukan pengujian dan eksperimen.
31 7. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

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

3 1. Bab Pendahuluan

4 Bab 1 berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan
5 sistematika pembahasan yang digunakan untuk menyusun skripsi ini.

6 2. Bab Dasar Teori

7 Bab 2 berisi teori-teori yang digunakan dalam pembuatan skripsi ini. Teori-teori tersebut
8 yaitu WSDC, Ionic, Capacitor, Cordova, UI Components, dan Migrasi Ionic.

9 3. Bab Analisis

10 Bab 3 berisi analisis yang dilakukan pada skripsi ini, meliputi analisis sistem, analisis kebutuhan
11 aplikasi WSDC.

12 4. Bab Perancangan

13 Bab 4 berisi perancangan aplikasi, meliputi

14 5. Bab Implementasi dan Pengujian

15 Bab 5 berisi implementasi dan pengujian aplikasi, meliputi

16 6. Bab Kesimpulan dan Saran Bab 6 berisi kesimpulan dari hasil pembangunan aplikasi beserta
17 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

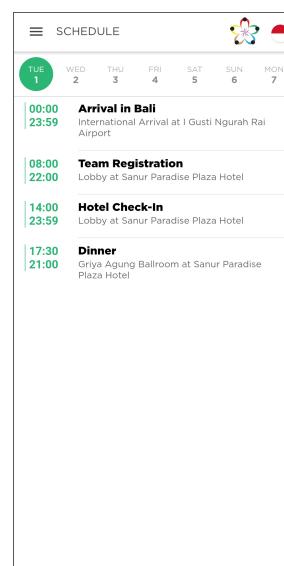
- 7 Aplikasi WSDC 2017 Bali digunakan untuk menunjang keberlangsungan acara WSDC 2017 yang
8 diselenggarakan di Bali, Indonesia (Gambar 2.1a). Aplikasi WSDC 2017 Bali dapat diunduh
9 untuk sistem operasi *android* melalui URL <https://play.google.com/store/apps/details?id=org.wsdc2017indonesia.app&hl=en&gl=US>. Aplikasi ini dibangun dan dikembangkan oleh PT
10 DNArtworks Komunikasi Visual yang rilis di Play Store pada tanggal 30 Juli 2017, dengan versi
11 terakhir adalah versi 1.1.2 yang rilis pada 1 Agustus 2017. Selain rilis pada perangkat *android*,
12 aplikasi ini juga rilis untuk perangkat bergerak berbasis sistem operasi iOS. Namun saat ini aplikasi
13 tersebut sudah diturunkan dari App Store pada perangkat berbasis sistem opearsi iOS. Untuk
14 membuka dan memakai aplikasi WSDC 2017 Bali saat ini, pengguna tidak diperlukan *login* agar
15 dapat mengakses seluruh fitur yang tersedia. Lalu, untuk kepentingan skripsi ini, peneliti memiliki
16 akses ke dalam kode program aplikasi WSDC 2017 Bali.
17



(a) Halaman Utama



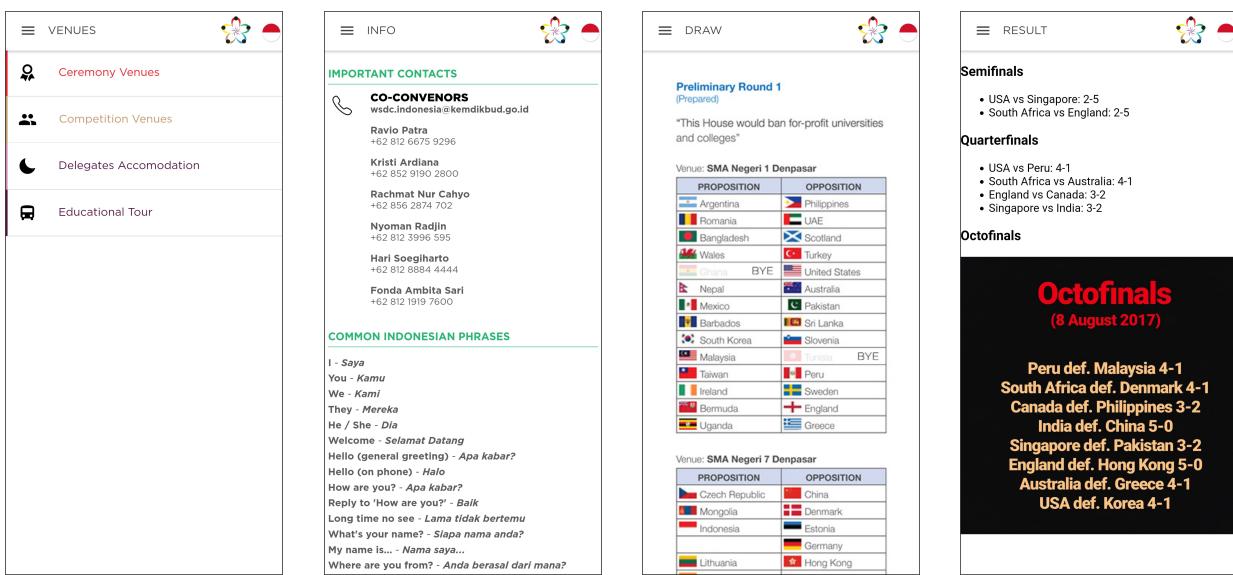
(b) Halaman *Announcements*



(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. *Announcements* : Pengguna dapat melihat pemberitahuan tentang berjalannya acara WSDC 2017 Bali (Gambar 2.1b).
 2. *Schedule* : Pengguna atau peserta WSDC 2017 Bali dapat melihat jadwal acara yang sudah diadakan (Gambar 2.1c).
 3. *Venues* : Pengguna atau peserta WSDC 2017 Bali dapat melihat berbagai macam lokasi acara, 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).
 4. *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).
 5. *Draw* : Pengguna dapat melihat melihat pembagian *venue* dan kubu proposisi atau oposisi dari hasil pengundian untuk para negara peserta (Gambar 2.2c).
 6. *Result* : Pengguna dapat melihat informasi terkait hasil dari pertandingan pada semi final, perempat final, dan perdelapan final (Gambar 2.2d).



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

2.2 Ionic Framework

Ionic Framework merupakan sebuah kerangka kerja *open source* lintas platform yang memungkinkan untuk mengembangkan aplikasi hibrida yang bekerja pada berbagai macam platform seluler seperti *android*, *iOS*, dan *Windows* [1]. Ionic memiliki berbagai macam *front-end library* dan *User Interface(UI)* *Components* yang digunakan untuk perancangan aplikasi menggunakan teknologi web seperti *HTML*, *CSS*, dan *Javascript*, dengan integrasi untuk berbagai *framework* seperti *Angular*, *React*, dan *Vue*. Saat pertama kali dibuat, Ionic menggunakan *AngularJS*. Namun, seiring saat *Angular* versi 2 yang menggunakan *TypeScript* dirilis, Ionic versi 2 dan selanjutnya menggunakan *Angular*.

1 Lalu, pada tahun 2019, Ionic mendukung penggunaan *framework* lain selain Angular, yaitu React
2 dan Vue. Di dalam Ionic, Angular digunakan untuk membangun aplikasi dan perutean, sehingga
3 aplikasi dapat sejalan dengan ekosistem Angular lainnya. Ionic menyediakan *toolkit* Angular untuk
4 membangun aplikasi dan terintegrasi dengan Angular CLI resmi yang menyediakan fitur khusus
5 untuk aplikasi Ionic Angular. Pada saat skripsi ini dibuat, Ionic versi terbaru adalah Ionic versi 5,
6 sedangkan Angular yang digunakan adalah Angular versi 12.

7 2.2.1 Native API

8 Native API memungkinkan pengembangan aplikasi langsung terintegrasi ke dalam *platform*. Pe-
9 ngembang dapat membuat aplikasi pada perangkat *mobile* untuk dapat diimplementasikan ke
10 berbagai *platform*, seperti IOS, dan Android, setelah pengembangan selesai di dalam *framework*
11 *native* tanpa perubahan, dan tidak mempengaruhi peforma dari aplikasi tersebut [3].

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

16 2.2.1.1 Capacitor

17 Penjelasan singkat mengenai capacitor di ionic

18 2.2.1.2 Cordova

19 Cordova merupakan *framework open source* yang dapat membuat pengembang untuk menggunakan
20 teknologi seperti HTML, JavaScript, dan CSS untuk membangun aplikasi untuk perangkat bergerak
21 yang dapat berjalan pada beberapa sistem operasi *mobile* [4]. Cordova menyediakan antarmuka
22 antara WebView dan lapisan *native* pada perangkat [3]. Selain dapat bekerja pada dua platform
23 seluler Android dan IOS, Cordova juga dapat digunakan pada platform seluler seperti Windows
24 Phone, Blackberry, dan FireOS.

25 Untuk mengonfigurasi proyek Cordova, saat ini dapat menggunakan *Command Line Tool* (CLI).
26 CLI membuat proyek dasar dan mengonfigurasinya agar berfungsi dengan platform seluler apa pun
27 yang didukung yang dapat digunakan. Cordova CLI juga dapat membuat pengembang memiliki
28 integrasi dan pengelolaan *plug in*. Selain itu, CLI juga dapat mengkompilasi aplikasi untuk berjalan
29 pada simulator atau pada perangkat yang sebenarnya. Serupa dengan Capacitor, Cordova membuat
30 pengembang dapat mengakses fitur *native* dari sebuah perangkat, seperti kamera, papan ketik, dan
31 geolokasi, menggunakan *plugin*. Ionic *Framework* telah terdapat berbagai macam TypeScript *wrapper*
32 untuk *plugins* Cordova. Untuk dapat menggunakan Cordova Plugins, yaitu dengan memasang
33 Cordova Plugins terlebih dahulu (Kode 2.1), dan memperbaruiinya ke versi terakhir (Kode 2.2) yang
34 dapat dilakukan melalui CLI. Setiap *plugins* memiliki dua komponen, yaitu kode *native* (Cordova),
35 dan kode TypeScript (Ionic Native). *Plugins* Cordova juga dibungkus di dalam Promise atau
36 Observable untuk menyediakan antarmuka plugin.

Kode 2.1: Kode untuk Memasang Cordova Plugins

37

38 1 | npm install cordova-plugin-name

2 | npx cap sync

Kode 2.2: Kode untuk Memperbarui Cordova Plugins

3
4 | npm install cordova-plugin-name@2
5 | npx cap update

7 2.2.2 UI Component

8 Framework Ionic menggunakan kemampuan Angular memperluas kosakata HTML yaitu menyertakan
9 *tag* khusus untuk menciptakan seluruh rangkaian komponen [3]. Semua komponen memiliki awalan
10 ion, sehingga dapat dikenali dalam markup. Sama seperti *tag* HTML standar, komponen Ionic juga
11 dapat menerima berbagai macam atribut sebagai pengaturan dari *tag* tersebut, seperti mengatur id
12 atau mendefinisikan kelas CSS tambahan. Terdapat beberapa komponen yang ada pada *framework*
13 Ionic yaitu :

- 14 • Action Sheet

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

- 19 • Alert

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

- 24 • Badge

25 Merupakan elemen *inline block* yang biasanya muncul di dekat elemen lain, berisi angka atau
26 karakter lain, yang digunakan sebagai pemberitahuan bahwa ada item tambahan yang terkait
27 dengan suatu elemen dan menunjukkan berapa banyak item yang ada. Penggunaan Badge
28 dengan menggunakan *tag* <ion-badge> (Kode 2.3).

Kode 2.3: Potongan Kode Program dari Badge Component

29
30 | 1 <ion-badge>99</ion-badge>

- 32 • Button

33 Merupakan elemen yang dapat diklik, biasanya digunakan dalam formulir atau dimanapun
34 yang membutuhkan fungsionalitas tombol. Button biasanya menampilkan teks, ikon, atau
35 bisa juga keduanya. Button dapat pula menggunakan atribut untuk menampilkannya dengan
36 penampilan tertentu. Penggunaan Button dengan menggunakan *tag* <ion-button> (Kode 2.4).

Kode 2.4: Potongan Kode Program dari Button Component

37
38 | 1 <ion-button>Default</ion-button>

- 40 • Card

41 Merupakan bagian standar dari tampilan antarmuka yang berfungsi sebagai titik masuk
42 ke dalam informasi yang lebih detail. Card dapat menjadi satu komponen, tetapi sering

1 kali terdiri dari beberapa header, judul, sub judul, dan konten. Penggunaan Card dengan
2 menggunakan tag `<ion-card>` yang dapat berisi `header`, `subtitle`, `title`, dan `content` (Kode 2.5).

Kode 2.5: Potongan Kode Program dari Card Component

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

- Content

16 Komponen content merupakan penyedia area konten yang bisa digunakan untuk mengontrol
17 area yang dapat digulir. Dalam satu tampilan, setidaknya terdapat satu buah content. Content
18 juga dapat dimodifikasi padding, margin, dan lainnya menggunakan *global style* yang berada
19 di CSS Utilites atau mengubahnya secara individual dengan menggunakan CSS. Penggunaan
20 Content dengan menggunakan tag `<ion-content>` (Kode 2.6).

Kode 2.6: Potongan Kode Program dari Content Component

```
21
22      <ion-content
23          [scrollEvents]="true"
24          (ionScrollStart)="logScrollStart()"
25          (ionScroll)="logScrolling($event)"
26          (ionScrollEnd)="logScrollEnd()">
27          <h1>Main Content</h1>
28
29          <div slot="fixed">
30              <h1>Fixed Content</h1>
31          </div>
32      </ion-content>
```

- Date and Time Pickers

35 Datetime merupakan penampil antarmuka untuk pengguna memilih tanggal dan waktu.
36 Terdapat kolom yang dapat digulir yang dapat digunakan untuk memilih tahun, bulan, hari,
37 jam, dan menit secara individual. Komponen ini menampilkan nilai di dua tempat, yaitu di
38 komponen `<ion-datetime>` (Kode 2.13), dan di antarmuka pemilih yang ditampilkan dari
39 bawah layar.

Kode 2.7: Kode Program dari Datetime Component dengan Format Bulan-Hari-Tahun

```
40
41      <ion-datetime displayFormat="MM_DD_YY" placeholder="Select Date"></ion-datetime>
```

- Infinite Scroll

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

Kode 2.8: Potongan Kode Program dari Infinite Scroll Component

```
47
48      <ion-infinite-scroll threshold="100px" (ionInfinite)="loadData($event)">
49          <ion-infinite-scroll-content
50              loadingSpinner="bubbles"
51              loadingText="Loading more data...">
52          </ion-infinite-scroll-content>
53      </ion-infinite-scroll>
```

1 • Item

2 Item merupakan elemen yang dapat berisi teks, ikon, avatar, gambar, masukan, dan elemen
3 asli atau kustom lainnya. Biasanya, item ditempatkan di dalam sebuah *list* bersamaan dengan
4 item lainnya dengan tag `<ion-item>` (Kode 2.9). Dapat dilakukan *swipe*, dihapus, disusun
5 ulang, diedit, dan lainnya.

Kode 2.9: Potongan Kode Program dari Item Component

```
6
7   1 <ion-item>
8     2   <ion-label>
9       3     Item
10      4   </ion-label>
11    5 </ion-item>
```

13 • Menu

14 Komponen menu merupakan panel navigasi samping yang dapat dilakukan *slides* dari sisi
15 pada tampilan halaman saat ini menggunakan tag `<ion-menu>` (Kode 2.10). Pada dasarnya,
16 Menu muncul dari kiri, tetapi sisi kemunculan menu dapat diganti.

Kode 2.10: 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-item>Menu Item</ion-item>
15      </ion-list>
16    </ion-content>
17  </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 memfilter isi di dalam daftar, serta beberapa kasus serupa lainnya (Kode 2.11).

Kode 2.11: Kode Program dari Modal

```

24
25   import { Component, Input } from '@angular/core';
26
27   @Component({
28     selector: 'modal-page',
29   })
30   export class ModalPage {
31
32     constructor() {}
33
34   }

```

- Navigation

Navigation adalah komponen mandiri yang digunakan untuk membuat komponen baru ke dalam *stack*. Navigation tidak terikat kepada router tertentu, menakibatkan 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.

- Tabs

Tabs merupakan navigasi *top-level* yang mengimplementasi sebuah *tab-based navigation*. Tabs dapat digunakan dengan tag `<ion-tabs>` (Kode 2.12) yang tidak memiliki *styling* apapun dan bekerja sebagai *router outlet* untuk menangani navigasi.

Kode 2.12: Kode Program dari Tabs

```

47
48   <ion-tabs>
49     <ion-tab-bar slot="bottom">
50       <ion-tab-button tab="schedule">
51         <ion-icon name="calendar"></ion-icon>
52         <ion-label>Schedule</ion-label>
53         <ion-badge>6</ion-badge>
54       </ion-tab-button>
55
56       <ion-tab-button tab="speakers">
57         <ion-icon name="person-circle"></ion-icon>
58         <ion-label>Speakers</ion-label>
59       </ion-tab-button>
60
61     </ion-tab-bar>
62   </ion-tabs>

```

1 • Toolbar

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

Kode 2.13: Kode Program dari Toolbar dengan Button di Dalamnya

```
6
7   <ion-toolbar>
8     <ion-buttons slot="start">
9       <ion-back-button></ion-back-button>
10      </ion-buttons>
11      <ion-title>Back Button</ion-title>
12    </ion-toolbar>
```

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

18 **2.2.3 Migrasi Ionic 3 ke Ionic 5**

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

21 1. Migrasi Ionic 3 ke Ionic 4

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

23 (a) Membuat Proyek Ionic Baru

24 Untuk membuat projek Ionic baru tanpa *template* apapun dengan menggunakan perintah
 25 **ionic start myApp blank** dan memilih Angular sebagai *frameworknya* 2.14.

Kode 2.14: Perintah Membuat Proyek Ionic Baru

```
26
27   1  ionic start myApp blank
```

29 (b) Menyalin Angular *services*

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

32 (c) Menyalin *Root-level Items*

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

36 • *Imports*

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

39 – Component Imports

40 Untuk kepentingan konsistensi dengan *framework* lain, maka untuk mengimpor
 41 komponen Ionic diawali dengan ion 2.15.

Kode 2.15: Impor Komponen pada Ionic 4

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

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

4 – Package Name

5 Terdapat perubahan pada Ionic 4, dimana nama *package* diubah menjadi @ionic/6
 6 angular. Untuk dapat menggunakannya dengan cara mencopot pemasangan
 7 Ionic 3 dan memasang Ionic 4 dengan nama *package* yang baru (Kode 2.16).

Kode 2.16: Penggunaan Button pada Ionic 4

```
8
9
10 npm uninstall ionic-angular
11 npm install @ionic/angular>
```

12 • Penamaan Berkas

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

15 – *Page*

16 Contoh perbedaan nama *file* pada folder Pages adalah sebagai berikut :

17 Pada Ionic 3 : home.html

18 Terdapat perubahan pada Ionic 4 menjad : home.page.html

19 – *App*

20 Contoh perbedaan nama *file* pada direktori App adalah sebagai berikut :

21 Pada Ionic 3 : app.html

22 Terdapat perubahan pada Ionic 4 menjad : app-component.html

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

25 (e) Menyalin Bagian-bagian Aplikasi

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

28 • Shadow DOM sudah aktif secara *default*.

29 • Halaman atau komponen Sass tidak lagi dibungkus dengan tag halaman / komponen
 30 dan harus menggunakan opsi styleUrls milik Angular dari dekorator @Component.

31 • RxJS

32 Pada Ionic 3, RxJS yang digunakan adalah versi 5. Sedangkan pada Ionic 4, RxJS
 33 yang digunakan adalah versi 6. Terdapat beberapa perbedaan diantara kedua versi
 34 tersebut, diantaranya yaitu :

35 – *Operator*

36 Dalam pemanggilan operator, sejak RxJS versi 6 perlu menambahkan .pipe
 37 untuk menggunakan operator apapun.

38 • *Lifecycle Hooks* tertentu harus digantikan dengan Angular Hooks.

39 • Perubahan markup yang mungkin saja dibutuhkan.

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

1 – *Button*

2 Terdapat perbedaan pada *tag* untuk membuat *Button*, yang semula pada Ionic 3
 3 adalah <button> menjadi <ion-button> pada Ionic 4 [2.17](#).

Kode 2.17: Penggunaan *Button* pada Ionic 4

```
4
5           <ion-button (click)="doSomething()">
6            Default Button
7          </ion-button>
```

9 – Floating Action Button (FAB)

10 Terdapat perbedaan pada *tag* di dalam <ion-fab>, yang semula pada Ionic 3
 11 adalah <button> menjadi <ion-fab-button> pada Ionic 4 [2.18](#).

Kode 2.18: Penggunaan Floating Action Button pada Ionic 4

```
12
13           <ion-fab>
14            <ion-fab-button>
15            <ion-icon name="add"></ion-icon>
16            </ion-fab-button>
17            <ion-fab-list>
18            <ion-fab-button>
19            <ion-icon name="logo-facebook"></ion-icon>
20            </ion-fab-button>
21            </ion-fab-list>
22
23          </ion-fab>
```

24 – Label

25 Pada Ionic 4, atribut untuk mengatur posisi dari label digabungkan dengan
 26 atribut *position* [2.19](#).

Kode 2.19: Penggunaan Atribut *Position* pada Ionic 4

```
27
28           <ion-item>
29            <ion-label position="floating">Floating Label</ion-label>
30            <!-- input -->
31          </ion-item>
```

33 – Menu

34 Terdapat beberapa perubahan nama pada Ionic 4, yaitu :

35 * Perubahan Nama Properti

- 36 · swipeEnable

37 Pada Ionic 3 : swipeEnabled

38 Sedangkan pada Ionic 4 menjadi : swipeGesture

- 39 · content

40 Pada Ionic 3 : content

41 Sedangkan pada Ionic 4 menjadi : contentId

42 * Perubahan Nama Events

- 43 · ionClose

44 Pada Ionic 3 : ionClose

45 Sedangkan pada Ionic 4 menjadi : ionDidClose

- 46 · ionOpen

47 Pada Ionic 3 : ionOpen

48 Sedangkan pada Ionic 4 menjadi : ionDidOpen

```

1      – Nav
2      Terdapat beberapa perubahan Nav pada Ionic 4, yaitu :
3      * Perubahan Nama Method
4          · remove
5              Pada Ionic 3 : remove
6              Sedangkan pada Ionic 4 untuk menghindari konflik dengan HTML, berubah
7              menjadi : removeIndex
8          · getActiveChildNavs
9              Pada Ionic 3 : getActiveChildNavs
10             Sedangkan pada Ionic 4 menjadi : getChildNavs
11     * Perubahan Nama Prop
12         Pada Ionic 3 : swipeBackEnabled
13         Sedangkan pada Ionic 4 menjadi : swipeGesture
14     – Navbar
15         Pada Ionic 4, terdapat penghapusan terhadap komponen <ion-navbar> karena
16         untuk menjaga agar selalu menggunakan <ion-toolbar> dengan back button yang
17         eksplisit 2.20.

```

Kode 2.20: Penggunaan Navbar pada Ionic 4 dengan *Back Button*

```

18
19          <ion-toolbar>
20              <ion-buttons slot="start">
21                  <ion-back-button></ion-back-button>
22              </ion-buttons>
23              <ion-title>My Navigation Bar</ion-title>
24          </ion-toolbar>

```

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

2. Migrasi Ionic 4 ke Ionic 5

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

- CSS

- CSS *Utilities*

Karena pada versi sebelumnya, yaitu Ionic versi 4, terdapat masalah dengan menggunakan atribut CSS dengan *framework* yang menggunakan JSX dan TypeScript, Ionic *Framework* menambahkan dukungan untuk beberapa *framework*, dan pada Ionic 5 menambahkan kelas CSS. Ionic versi 5 menghapus atribut CSS dan mendukung konsistensi. Selain itu, Ionic versi 5 juga mengubah ke kelas dengan diawali dengan ion untuk menghindari konflik dengan atribut asli dan CSS dari pengguna (Kode [2.21](#)).

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

Kode 2.21: 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 class="ion-wrap"></ion-item>
```

– *Display Classes*

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

– *Activated, Focused, Hover States*

Kelas .activated secara otomatis ditambahkan ke komponen yang dapat diklik, mengalami perubahan nama menjadi .ion-activated. Selain itu juga memperbarui komponen Action Sheet sehingga variabel akan diawali dengan *button*. Hal ini dapat memungkinkan aplikasi tetap memiliki kontrol atas *opacity* jika diinginkan, tetapi saat memperbarui status, hanya perlu mengatur variabel utama, yaitu -background-activated, -background-focused, -background-hover. Hal tersebut penting saat mengubah tema global, karena memperbarui warna *toolbar* akan secara otomatis memperbarui *hover states* untuk semua *buttons* di *toolbar* (Kode 2.22).

Kode 2.22: Contoh Kode *Hover States* pada Ionic 5

```

1 /* Setting the button background on hover to solid red */
2 ion-button {
3     --background-hover: red;
4     --background-hover-opacity: 1;
5 }
6
7 /* Setting the action sheet button background on focus to an opaque green */
8 ion-action-sheet {
9     --button-background-focus: green;
10    --button-background-focus-opacity: 0.5;
11 }
12
13 /*
14 * Setting the fab button background on hover to match the text color with
15 * the default --background-hover-opacity on md
16 */
17 .md ion-fab-button {
18     --color: #222;
19     --background-hover: #222;
20 }
```

– *Distributed Sass*

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

- *Components*

Terdapat perubahan beberapa komponen pada Ionic 5, yaitu :

- *Back Button* dan *Button*

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

- *Controllers*

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

- *Header / Footer*

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

- 3 – *List Header* Konten teks apa pun di dalam `<ion-list-header>` harus dibungkus dengan
 4 `<ion-label>` sesuai dengan gaya desain yang baru (Kode 2.23). Jika label tidak ada,
 5 maka perataan tombol di header bisa saja terlihat tidak aktif.

Kode 2.23: Kode Program untuk List Header

```
6
7           <ion-list-header>
8             <ion-label>New This Week</ion-label>
9               <ion-button>See All</ion-button>
10            </ion-list-header>
```

- 12 – *Menu*

13 Fungsi `swipeEnable()` telah dihapus di Angular, sebagai gantinya menggunakan
 14 `swipeGesture()`. Lalu nilai *left* dan *right* telah dihapus, gunakan *start* dan *end*
 15 sebagai gantinya. Selain itu ada penghapusan atribut utama, sebagai gantinya
 16 yaitu dengan menggunakan `content-id` (untuk vanila JS/Vue) dan `contentId` (untuk
 17 Angular/React) (Kode 2.24).

Kode 2.24: Kode Program untuk Menu

```
18
19           <ion-menu content-id="main"></ion-menu>
20           <ion-content id="main">...</ion-content>
```

- 22 – *Select Option*

23 Properti *selected* telah dihapus. Sebagai gantinya harus mengatur properti nilai pada
 24 `ion-select` induk agar sesuai dengan opsi terpilih yang diinginkan (Kode 2.25).

Kode 2.25: Kode Program untuk Select Option

```
25
26           <ion-select value="two">
27             <ion-select-option value="one">One</ion-select-option>
28             <ion-select-option value="two">Two</ion-select-option>
29           </ion-select>
```

- 31 – *Toast*

32 Properti *close button* seperti `showCloseButton` dan `closeButtonText` telah dihapus.
 33 Sebagai gantinya, gunakan *buttons array* untuk fungsi batal (Kode 2.26).

Kode 2.26: Kode Program untuk Toast

```
34
35           async presentToast() {
36             const toast = await this.toastController.create({
37               message: 'Your settings have been saved.',
38               buttons: [
39                 {
40                   text: 'Close',
41                   role: 'cancel',
42                   handler: () => {
43                     console.log('Close clicked');
44                   }
45                 }
46               ]
47             });
48             toast.present();
49           }
```

51 Selain yang sudah disebutkan, terdapat beberapa komponen lain yang mendapat perubahan
 52 di Ionic 5, namun tidak ditulis di dalam dokumen skripsi ini. Komponen tersebut

1 antara lain Action Sheet, Anchor, Card, FAB, Item, Menu Button, Nav Link, Radio,
 2 Segment, Segment Button, Skeleton Text, Split Pane, dan Tabs ³.

3 • *Colors*

4 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 Perubahan Warna Bawaan di Ionic 5

5 • *Events*

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

8 • *Package dan Dependencies*

9 Untuk memasang *package* dan *dependencies* pada Angular, dapat memanfaatkan npm
 10 pada CLI, dengan menjalankan pemasangan pada *package* ionic-angular (Kode 2.27).
 11 Namun jika ingin membuat proyek baru, dapat dibuat dari CLI dan aplikasi yang ada
 12 dapat dimigrasikan secara manual.

Kode 2.27: Kode untuk Memasang *Package* dan *Dependencies* pada Angular

13 14 `npm install @ionic/angular@latest @ionic/angular-toolkit@latest --save`

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

DAFTAR REFERENSI

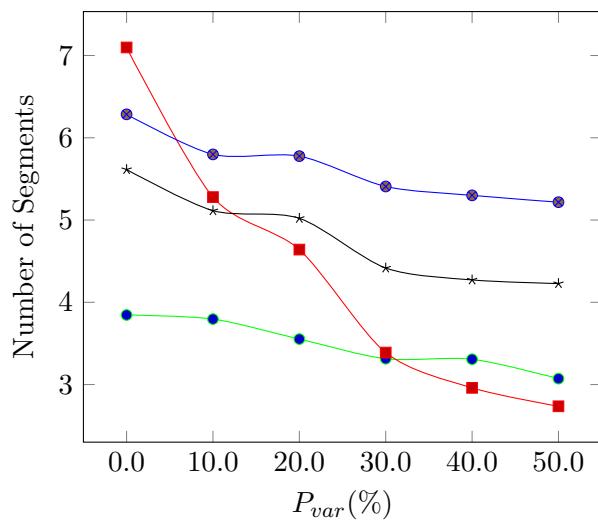
- [1] Waranashiwar, J. dan Ukey, M. (2018) Ionic framework with angular for hybrid app development. *International Journal of New Technology and Research*, **4**, 01–02.
- [2] Wohlgethan, E. (2018) Supporting web development decisions by comparing three major javascript frameworks: Angular, react and vue.js. Thesis. Hochschule für angewandte Wissenschaften Hamburg, Germany.
- [3] Griffith, C. (2017) *Mobile App Development with Ionic : Cross-Platform Apps with Ionic, Angular and Cordova*, 1st edition. O'Reilly Media, Inc., California, USA.
- [4] 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.

LAMPIRAN A
KODE PROGRAM

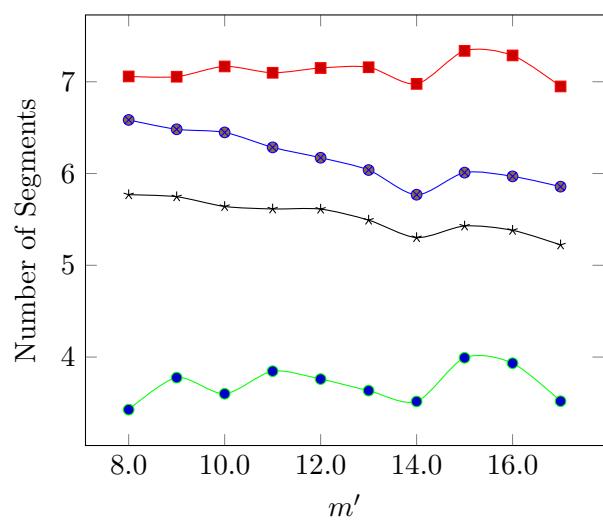
LAMPIRAN B

HASIL EKSPERIMENT

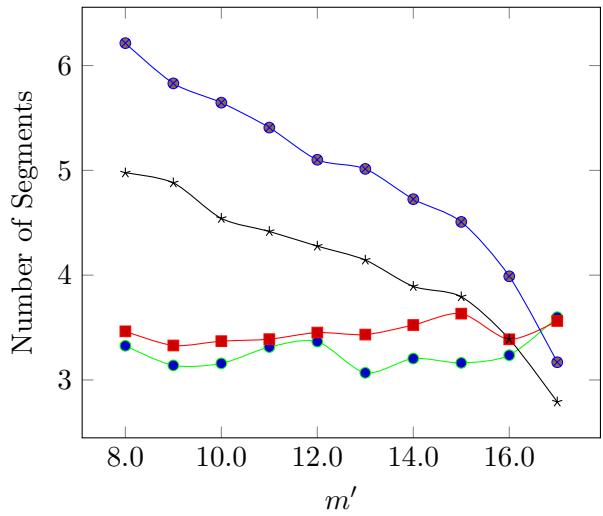
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



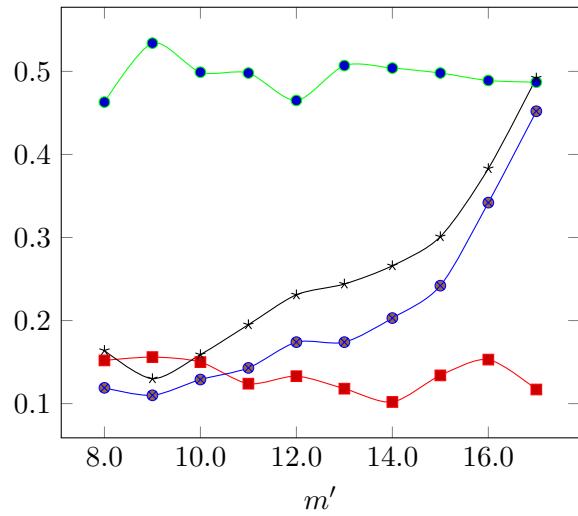
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4