

SKRIPSI

PEMBUATAN ULANG APLIKASI RUGBY INDONESIA DENGAN IONIC 7 DAN CAPACITOR



Nathanael Adi Trianto

NPM: 6181901041

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2023

DAFTAR ISI

DAFTAR ISI	iii
DAFTAR GAMBAR	v
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	4
1.4 Batasan Masalah	4
1.5 Metodologi	4
2 LANDASAN TEORI	5
2.1 Aplikasi Rugby Indonesia	5
2.2 ReactJS [1]	5
2.2.1 Hooks	6
2.2.2 Components	7
2.3 Ionic 7 Framework [2]	9
2.3.1 UI Components	9
2.3.2 Capacitor Native	15
2.4 RSS	17
3 ANALISIS	19
3.1 Analisis Sistem Kini	19
3.1.1 Halaman Utama	20
3.1.2 Halaman Teammate Photos	20
3.1.3 Halaman Rugby Clubs	20
3.1.4 Halaman Fixture & Results	21
3.2 Analisis Masalah Aplikasi Rugby Indonesia	21
3.3 Analisis RSS Rugby Indonesia	22
3.4 Analisis Sistem Usulan	22
3.4.1 Halaman Utama	23
3.4.2 Halaman Teammate Photos	23
DAFTAR REFERENSI	25
A KODE PROGRAM	27
B HASIL EKSPERIMEN	29

DAFTAR GAMBAR

1.1	Halaman aplikasi Rugby Indonesia	3
2.1	Gambar Hasil Action Sheet	10
3.1	Use Case Diagram Rugby Indonesia App	19
3.2	Halaman Utama Aplikasi Rugby Indonesia	21
3.3	Tangkapan Layar RSS Rugby Indonesia	22
3.4	UCD Sistem Usulan Rugby Indonesia	22
3.5	Analisis dari Halaman Utama Rugby Indonesia	23
3.6	Analisis dari Halaman Teammate Photos	24
B.1	Hasil 1	29
B.2	Hasil 2	29
B.3	Hasil 3	29
B.4	Hasil 4	29

1

BAB 1

2

PENDAHULUAN

3 1.1 Latar Belakang

4 Dikutip dari buku Olahraga Rugby (2010) [3], *Rugby* merupakan olahraga yang berasal dari negara
5 Inggris, di mana olahraga ini memiliki kemiripan dengan sepak bola. *Rugby* berawal dari zaman
6 Yunani Kuno, di mana seorang yang bernama William Webb Ellis melanggar aturan bermain
7 bola pada tahun 1823 dengan membawa bola ke gawang lawan. Olahraga ini juga pertama kali
8 diperkenalkan pada tahun 1986 pada zaman penjajahan, dan dihidupkan kembali pada tahun 2004
9 dengan bantuan negara lain seperti Australia dan Selandia Baru sehingga terbentuklah Indonesian
10 Rugby Football Union (IRFU) yang bertujuan untuk mengembangkan olahraga ini.

11 Dikutip dari Asia Rugby¹, Persatuan Rugby Union Indonesia atau yang biasa disingkat dengan
12 PRUI merupakan Asosiasi Uni Rugbi Indonesia yang bertanggung jawab atas pengelolaan dan
13 pengembangan Rugby Union di Indonesia yang dibangun. PRUI dibentuk pada tahun 2004 dan
14 telah menjadi anggota penuh dari Komite Nasional Olahraga Indonesia atau yang biasa disingkat
15 dengan KONI dan Komite Olimpiade Indonesia. PRUI juga memiliki banyak sejarah, seperti
16 menjadi anggota penuh dari Asia Rugby dan World Rugby². PRUI telah mengembangkan Rugby
17 di berbagai sekolah di daerah Indonesia seperti Daerah Istimewa Yogyakarta, dan Jawa Barat.
18 Pada tahun 2023, terdapat 35 klub rugby, 700 pemain terdaftar, 45 pelatih terdaftar, dan 23 wasit
19 terdaftar di Indonesia³. Rugby Indonesia juga memiliki *website*, di mana *website* tersebut didirikan
20 oleh PT. Dnartworks Komunikasi Visual untuk memberikan info seputar kegiatan olah raga *rugby*
21 yang berada di Indonesia. *Website* ini didirikan menggunakan platform Wordpress.

22 Dikutip dari⁴, Wordpress merupakan sebuah platform Content Management System (CMS)
23 yang sangat populer untuk membangun *website* dan blog. Diluncurkan pada tahun 2004 oleh Matt
24 Mullenweg dan Mike Little, WordPress memungkinkan pengguna untuk membuat dan mengelola
25 *website* tanpa perlu memiliki pengetahuan dasar pemrograman. Content Management System
26 (CMS) yang dibagun WordPress menggunakan bahasa pemrograman PHP dan database MySQL.
27 Dengan WordPress, pengguna baru dapat mengelola website tanpa perlu memahami teknis bahasa
28 pemrograman, hal itu pun yang membuat para pemula dapat dengan mudah membuat website,
29 bahkan dalam hitungan menit.

30 Dikutip dari GitHub Ionic [4], Ionic Framework merupakan *toolkit UI open-source* untuk
31 membangun aplikasi modern, *cross-platform* yang berkualitas tinggi dari satu kode sumber dengan

¹<https://www.asiarugby.com/unions/indonesia/>

²<https://rugbyindonesia.or.id/tentang/>

³<https://rugbyindonesia.or.id/>

⁴<https://www.dicoding.com/blog/apa-itu-cms-wordpress/>

1 JavaScript dan web. Ionic menyediakan alat dan layanan untuk mengembangkan aplikasi *hybrid*
2 *mobile*, *desktop*, dan *progressive web* berdasarkan teknologi dan praktik pengembangan web modern,
3 menggunakan teknologi web seperti CSS, HTML5, dan Sass. Ionic 7 merupakan versi *stable release*
4 terbaru dari Ionic, yang memperkenalkan cara kerja yang lebih efisien dengan kontrol formulir
5 seperti *Toggle* atau *Input*. Komponen *Item* dan *Label* tidak lagi diperlukan, dan setiap kontrol
6 formulir menangani konten label secara langsung. Selain itu, fitur tertentu seperti teks bantuan
7 atau mode pengisian *input* telah dipindahkan dari *ion-item* ke kontrol formulir yang sesuai seperti
8 *ion-input*, *ion-textarea*, dan *ion-select*. Perubahan ini mengurangi *boilerplate* kode dengan
9 menghilangkan persyaratan *ion-item* dan *ion-label*. Komponen Ionic Framework secara otomatis
10 menyesuaikan tampilan dan nuansanya dengan platform di mana komponen tersebut dijalankan,
11 memungkinkan gestur dan perilaku *native* yang sama dengan yang biasa digunakan pengguna.
12 Ionic memiliki lebih dari 100 komponen *UI* yang telah dirancang sebelumnya, tipografi, dan tema
13 dasar yang menyesuaikan dengan setiap platform. Ini dioptimalkan untuk *mobile* dengan animasi
14 yang diakselerasi oleh *hardware*, *lazy loading*, dan *scrolling* 60FPS. Ionic CLI digunakan untuk
15 membuat, membangun, dan menguji aplikasi serta memanfaatkan *Live Reload*, *deployment*, dan
16 dokumen yang baik.

17 Dikutip dari dokumentasi Capacitor [5], Capacitor merupakan *runtime native open source* yang
18 dapat digunakan untuk membangun aplikasi Web Native. Capacitor memungkinkan pengembang
19 untuk membangun aplikasi *mobile* pada perangkat iOS dan Android. Selain itu, Capacitor juga
20 memungkinkan pengembang untuk membangun aplikasi web progresif. Aplikasi yang dibangun
21 akan menggunakan JavaScript, HTML, dan CSS. Capacitor juga menyediakan *cross-platform API*
22 yang dapat digunakan pada berbagai platform seperti iOS, Android, Web dengan satu kode sumber.
23 Capacitor ini sendiri dikembangkan oleh tim Ionic, di mana tim ini jugalah yang mengembangkan
24 *framework* Ionic. Capacitor dirancang untuk menggantikan Apache Cordova, yang merupakan
25 *runtime* yang digunakan oleh Ionic sebelumnya. Capacitor ini juga memiliki beberapa keunggulan
26 dibandingkan Apache Cordova, diantaranya adalah lebih ringan, lebih cepat, lebih modern, dan
27 lebih fleksibel.

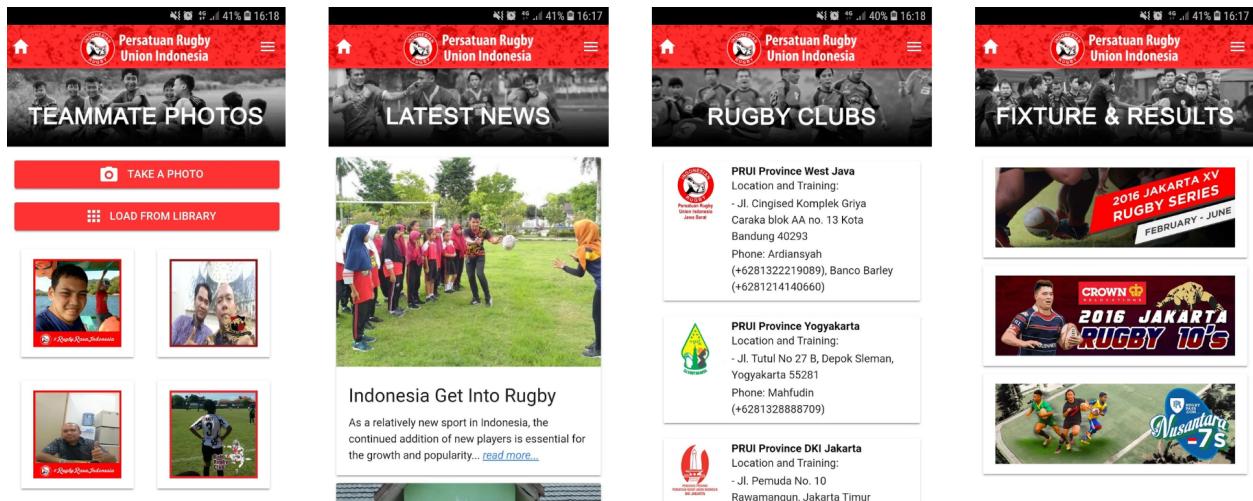
28 Dikutip dari dokumentasi RSS⁵, RSS atau Really Simple Syndication adalah format sindikasi
29 konten web yang merupakan dialek dari XML. Semua file RSS harus disesuaikan dengan spesifikasi
30 XML 1.0 yang dipublikasikan oleh World Wide Web Consortium (W3C). Dokumen RSS terdiri
31 dari elemen <*rss*> yang memiliki atribut wajib bernama version untuk menentukan versi RSS yang
32 digunakan. Di bawah elemen <*rss*> terdapat elemen <*channel*> yang berisi informasi tentang
33 saluran (metadata) dan isinya.

34 Pada sekitar tahun 2015, perusahaan PT. DNArtworks Komunikasi Visual membuat aplikasi
35 Rugby Indonesia yang memanfaatkan Apache Cordova. Aplikasi tersebut memiliki:

- 36 • Halaman *Latest News* yang diambil dari <https://rugbyindonesia.or.id/berita/> dengan
37 memanfaatkan protokol RSS. (Lihat gambar 1.1)
- 38 • Halaman *Fixture & Results*, namun sekarang sudah tidak ada. (Lihat gambar 1.1)
- 39 • Halaman *Teammate Photos* dengan fungsi:
 - 40 – Pengguna dapat langsung mengambil foto dari aplikasi tersebut.
 - 41 – Pengguna dapat langsung memberikan *frame* terhadap foto tersebut.

⁵<https://www.rssboard.org/rss-specification>

- 1 – Pengguna dapat langsung mengunggah foto tersebut ke dalam galeri publik.
- 2 • Halaman *Rugby Clubs* yang memiliki fungsi di mana pengguna dapat melihat klub *rugby* serta
- 3 info dari klub tersebut pada tiap daerah. (Lihat gambar 1.1)



Gambar 1.1: Halaman dari aplikasi Rugby Indonesia dari Google Play Store

4 Pada saat ini, aplikasi tersebut masih tersedia di Google Play Store⁶, namun aplikasi tersebut ti-
 5 dak dapat dipasang pada perangkat Android saat ini dikarenakan website <https://rugbyindonesia.>
 6 [or.id](http://id.or.id) sudah berubah dan juga Apache Cordova yang sudah tidak dapat digunakan kembali pada
 7 pengembangan aplikasi berbasis *framework* Ionic dikarenakan Ionic sendiri telah mengembangkan
 8 Native Capacitor sehingga Ionic dapat melepas ketergantungannya pada Angular dan Cordova⁷.
 9 Maka dari itu pada skripsi ini, akan dibuat ulang sebuah perangkat lunak Rugby Indonesia yang
 10 terbaru, sehingga perangkat lunak tersebut dapat *compatible* dengan perangkat android saat ini
 11 dengan memanfaatkan *backend* aplikasi yang sudah dibuat menggunakan Wordpress dan juga
 12 protokol RSS untuk melakukan komunikasi antar *backend* dan juga *frontend*. Perangkat lunak ini
 13 juga akan memanfaatkan Capacitor bawaan dari Ionic yang berfungsi untuk membangun antarmuka
 14 secara *native*.

15 Perangkat lunak ini akan dibuat dengan memanfaatkan bantuan *framework* Ionic 7 dan Native
 16 Capacitor dengan memiliki:

- 17 • Halaman *Latest News*, di mana pengguna dapat melihat berita terbaru dari Rugby Indonesia.
- 18 • Halaman *Teammate Photos* di mana pengguna dapat mengambil foto dan langsung mengung-
 gahnya ke dalam halaman *teammate photos*.
- 19 • Halaman *About Us* di mana pengguna dapat melihat tentang Rugby Indonesia.

21 1.2 Rumusan Masalah

22 Rumusan masalah yang akan dibahas pada tugas akhir ini adalah sebagai berikut:

- 23 1. Bagaimana membangun ulang serta mengembangkan perangkat lunak Rugby Indonesia dengan
 24 memanfaatkan *framework* Ionic 7?

⁶ <https://play.google.com/store/apps/details?id=id.or.rugbyindonesia.androidapp&hl=id>

⁷ <https://ionic.io/blog/ionic-isnt-cordova-anymore>

- 1 2. Bagaimana membangun ulang serta mengembangkan perangkat lunak Rugby Indonesia dengan
2 memanfaatkan Capacitor Native?

3 1.3 Tujuan

- 4 Tujuan yang ingin dicapai pada penulisan tugas akhir ini yaitu:
 - 5 1. Pembuatan ulang aplikasi Rugby Indonesia yang sudah memanfaatkan *framework* dari Ionic
6 terbaru yaitu Ionic 7.
 - 7 2. Pembuatan ulang aplikasi Rugby Indonesia yang sudah memanfaatkan Native Capacitor.

8 1.4 Batasan Masalah

- 9 Batasan masalah yang terdapat pada penggerjaan tugas akhir ini yaitu:
 - 10 1. Perangkat lunak ini dibuat untuk perangkat Android saja, tidak untuk iOS. Sehingga pengujian
11 dari perangkat lunak ini hanya dilakukan pada platform berbasis android saja. Perangkat
12 lunak ini hanya dibuat untuk perangkat lunak android dikarenakan peneliti tidak memiliki
13 perangkat lunak yang menggunakan sistem operasi iOS.
 - 14 2. Pengguna hanya bisa mengunggah foto dan melihat foto unggahan dari pengguna lain,
15 pengguna tidak dapat menghapus ataupun mengubah foto tersebut. Hal ini dikarenakan pada
16 aplikasi sebelumnya, pengguna hanya dapat melakukan hal tersebut.

17 1.5 Metodologi

- 18 Langkah-langkah yang dilakukan dalam penggerjaan tugas akhir ini yaitu:
 - 19 1. Melakukan studi literatur serta mendalami ReactJS sebagai salah satu perpustakaan JavaScript
20 untuk membangun tampilan antar muka.
 - 21 2. Melakukan studi literatur mengenai *framework* Ionic 7 dan juga Capacitor yang terdapat
22 pada Ionic Native.
 - 23 3. Melakukan analisis terhadap perangkat lunak yang ada dan melakukan perancangan terhadap
24 perangkat lunak yang akan dibuat.
 - 25 4. Membangun aplikasi Rugby Indonesia yang sudah menggunakan *framework* Ionic 7 serta
26 Capacitor.
 - 27 5. Melakukan pengujian dan eksperimen.
 - 28 6. Menulis dokumen tugas akhir.

¹

BAB 2

²

LANDASAN TEORI

³ 2.1 Aplikasi Rugby Indonesia

⁴ Aplikasi Rugby Indonesia merupakan aplikasi resmi dari Persatuan Rugby Union Indonesia yang
⁵ dapat memberikan informasi terbaru mengenai olahraga *rugby* di Indonesia. Aplikasi ini dapat
⁶ memberikan notifikasi langsung mengenai berita terakhir, turnamen yang akan datang, dan informasi
⁷ lainnya. Selain itu, aplikasi ini juga memungkinkan pengguna untuk mengambil gambar dan
⁸ menunjukkan dukungan mereka dengan rekan-rekan penggemar *rugby* lainnya dengan meng-*upload*
⁹ gambar ke dalam halaman *Teammate Photos*. Terdapat juga aplikasi multimedia pengenalan olahraga
¹⁰ *rugby* berbasis Android yang dapat memberikan informasi sejarah, peraturan, dan peralatan serta
¹¹ tempat-latihan *rugby* di beberapa Kabupaten/Kota. Aplikasi Rugby Indonesia tersedia di Google
¹² Play Store dan dapat diunduh secara gratis. Gambar dari halaman Rugby Indonesia dapat dilihat
¹³ pada Gambar 1.1.

¹⁴ Fitur-fitur yang ada pada aplikasi Rugby Indonesia saat ini yaitu:

- ¹⁵ 1. Halaman *Latest News* yang diambil dari <https://rugbyindonesia.or.id/berita/> dengan
¹⁶ memanfaatkan protokol RSS.
- ¹⁷ 2. Halaman *Fixture & Results*.
- ¹⁸ 3. Halaman *Teammate Photos* dengan fungsi:
 - ¹⁹ • Pengguna dapat langsung mengambil foto dari aplikasi tersebut.
 - ²⁰ • Pengguna dapat langsung memberikan *frame* terhadap foto tersebut.
 - ²¹ • Pengguna dapat langsung mengunggah foto tersebut ke dalam galeri publik.
- ²² 4. Halaman *Rugby Clubs* yang memiliki fungsi di mana pengguna dapat langsung mendaftar ke
²³ dalam *Rugby Clubs* yang berada di Indonesia.
- ²⁴ 5. Fungsi *Push Notifications*.

²⁵ 2.2 ReactJS [1]

²⁶ ReactJS atau React adalah sebuah *library* JavaScript yang digunakan untuk membangun *user*
²⁷ *interface* yang interaktif. ReactJS berisi kumpulan potongan kode JavaScript yang disebut “kompo-
²⁸ nen” yang bisa digunakan berulang kali untuk mendesain antarmuka pengguna. ReactJS bukanlah
²⁹ *framework* JavaScript, karena hanya bertugas untuk me-*render* komponen area tampilan aplikasi.
³⁰ ReactJS dapat digunakan untuk membuat aplikasi web dan *mobile*.

1 2.2.1 Hooks

2 Hooks adalah fitur yang memungkinkan penggunaan *state* dan fitur React lainnya tanpa membuat
 3 sebuah kelas. Hooks berfungsi untuk “mengaitkan” *state* dan fitur-fitur *lifecycle* React dari *function component*.
 4

5 State Hooks

6 State Hooks adalah salah satu jenis Hooks yang memungkinkan pengembang untuk menggunakan
 7 *state* pada *function component* tanpa harus membuat sebuah kelas. Cara untuk menggunakan State
 8 Hooks terdapat pada (Kode 2.1), di mana ‘useState’ digunakan untuk mendeklarasikan sebuah
 9 variabel *state* yang dapat langsung di *update*.

Kode 2.1: Contoh Potongan Kode State Hooks

```
10 11 function ImageGallery() {
12 12   const [index, setIndex] = useState(0);
13 13   // ...
```

15 Context Hooks

16 Context Hooks adalah salah satu jenis Hooks yang memungkinkan pengembang untuk menerima
 17 informasi dari *parent* yang jauh tanpa meneruskannya sebagai properti. Cara untuk menggunakan
 18 Context Hooks terdapat pada (Kode 2.2). Pada kode tersebut, ‘useContext’ digunakan untuk
 19 berlangganan pada sebuah *context* yang berada pada komponen.

Kode 2.2: Contoh Potongan Kode Context Hooks

```
20 21 function Button() {
22 22   const theme = useContext(ThemeContext);
23 23   // ...
```

25 Ref Hooks

26 Ref Hook pada React adalah salah satu dari beberapa Hooks bawaan yang memungkinkan peng-
 27 gunaan *refs* dalam komponen fungsional. *Ref* digunakan untuk mengakses DOM atau nilai dari
 28 elemen *child* dalam React. Cara untuk menggunakan Ref Hooks terdapat pada (Kode 2.3). Pada
 29 kode tersebut, ‘useRef’ digunakan untuk mereferensikan sebuah nilai yang tidak digunakan dalam
 30 *rendering*.

Kode 2.3: Contoh Potongan Kode Ref Hooks

```
31 32 function Form() {
33 33   const inputRef = useRef(null);
34 34   // ...
```

36 Effect Hooks

37 Effect Hooks adalah fungsi yang memungkinkan agar sebuah komponen dapat terhubung dan
 38 tersinkronisasi dengan sistem eksternal seperti pengambilan data, berlangganan data, atau mengubah
 39 DOM dari sebuah *function component*. Cara penggunaan dari Effect Hooks terdapat pada (Kode
 40 2.4). Pada kode tersebut, ‘useEffect’ berfungsi untuk menghubungkan komponen ke sistem eksternal.

Kode 2.4: Contoh Potongan Kode Effect Hooks

```
1  function ChatRoom({ roomId }) {
2    useEffect(() => {
3      const connection = createConnection(roomId);
4      connection.connect();
5      return () => connection.disconnect();
6    }, [roomId]);
7    // ...
8  }
```

10 Performance Hooks

11 Performance Hooks adalah fungsi yang mengoptimalkan performa *render* ulang dengan cara mele-
12 watkan pekerjaan yang tidak diperlukan. Pekerjaan ini biasanya berupa data yang tidak berubah
13 dari *render* sebelumnya. Cara penggunaan Performance Hooks terdapat pada (Kode 2.5). Pada
14 kode tersebut, ‘useMemo’ berfungsi untuk menyimpan hasil perhitungan yang besar ke dalam *cache*.

Kode 2.5: Contoh Potongan Kode Performance Hooks

```
15
16 1 function TodoList({ todos, tab, theme }) {
17 2   const visibleTodos = useMemo(() => filterTodos(todos, tab), [todos, tab]);
18 3   // ...
19 4 }
```

21 Resource Hooks

²² Resource Hooks dapat diakses oleh komponen tanpa menjadi bagian dari *state*. Cara penggunaan
²³ Resource Hooks terdapat pada (Kode 2.6).

Kode 2.6: Contoh Potongan Kode Resource Hooks

```
24
25  function MessageComponent({ messagePromise }) {
26    const message = use(messagePromise);
27    const theme = use(ThemeContext);
28    // ...
29 5 }
```

31 2.2.2 Components

32 Components pada React adalah bagian-bagian kecil yang memungkinkan pengembang untuk
33 membagi antarmuka pengguna menjadi bagian-bagian independen dan dapat digunakan kembali.
34 Components dapat berupa kelas atau fungsi. Komponen menerima masukan yang disebut “props”
35 dan mengembalikan elemen React yang mendefinisikan tampilan. Terdapat 4 komponen yang
36 terdapat pada React, yaitu:

- <Fragment>
 - <Profiler>
 - <Suspense>
 - <StrictMode>

41 Fragment

42 Fragment pada komponen React adalah fitur yang memungkinkan pengembang untuk mengelom-
43 pokkan sejumlah elemen anak tanpa perlu menambahkan *node* ekstra ke DOM. Hal ini berguna
44 ketika ingin mengembalikan beberapa elemen dari sebuah komponen tanpa harus membungkusnya

1 dalam sebuah elemen DOM tambahan seperti ‘`div`’. Fragment juga dapat digunakan untuk me-
 2 `render` beberapa elemen secara bersamaan tanpa harus menggunakan ‘`<div>`’. Ada dua cara untuk
 3 mendefinisikan Fragment, yaitu dengan menggunakan `<React.Fragment>` atau dengan menggunakan
 4 sintaksis singkat `<>...</>`. Cara untuk penggunaan komponen Fragment terdapat pada contoh
 5 (Kode 2.7).

Kode 2.7: Contoh Potongan Kode Fragment

```
6
7 1 function Post() {
8 2   return (
9 3     <>
10 4       <PostTitle />
11 5       <PostBody />
12 6     </>
13 7   );
14 8 }
```

16 Pada kode tersebut, Fragment akan mengelompokan dua elemen secara bersamaan menjadi satu
 17 grup dan akan mengembalikan grup yang berisi ‘`<PostTitle>`’ dan ‘`<PostBody>`’.

18 Profiler

19 Profiler adalah komponen yang digunakan untuk mengukur seberapa sering sebuah aplikasi React
 20 melakukan rendering dan seberapa besar biaya yang dikeluarkan untuk melakukan *rendering*
 21 tersebut. Cara penggunaan komponen Profiler terdapat pada contoh (Kode 2.8).

Kode 2.8: Contoh Potongan Kode Profiler

```
22
23 1 <Profiler id="App" onRender={onRender}>
24 2   <App />
25 3 </Profiler>
```

27 Suspense

28 Suspense adalah fitur yang memungkinkan penundaan *render* komponen sampai data yang diper-
 29 lukan tersedia. Fitur ini berguna untuk meningkatkan responsivitas aplikasi dengan membiarkan
 30 komponen me-*render* terlebih dahulu sebelum data siap. Cara penggunaan komponen Suspense
 31 terdapat pada contoh (Kode 2.9).

Kode 2.9: Contoh Potongan Kode Suspense

```
32
33 1 <Suspense fallback={<Loading />}>
34 2   <SomeComponent />
35 3 </Suspense>
```

37 StrictMode

38 StrictMode adalah sebuah komponen yang digunakan untuk menyoroti potensi masalah dalam
 39 sebuah aplikasi. Mode ini tidak berdampak dalam pembangunan produksi dan dapat diaktifkan
 40 untuk bagian-bagian tertentu dalam aplikasi. StrictMode membantu dalam mengidentifikasi
 41 komponen-komponen dengan siklus hidup yang tidak aman dan melakukan berbagai pemeriksaan
 42 tambahan untuk turunannya. Mode ini berguna saat mengembangkan kode baru atau melakukan
 43 *debugging*. StrictMode dapat diterapkan pada bagian mana pun dalam aplikasi, bukan hanya pada
 44 keseluruhan aplikasi. Mode ini membantu dalam menulis kode React dengan cara yang lebih baik

¹ dengan memberikan peringatan terkait praktik terbaik. Mode ini juga dapat digunakan baik pada
² komponen fungsional maupun kelas. Namun, StrictMode me-*render* setiap komponen dalam aplikasi
³ dua kali, sehingga sebaiknya hanya digunakan saat pengembangan atau *debugging*. Cara untuk
⁴ menggunakan StrictMode terdapat pada contoh (Kode 2.10). Pada kode tersebut, apabila terjadi
⁵ error pada ‘<App />’, maka React akan memunculkan pesan error sebelum ‘<App />’ di-*render*.

Kode 2.10: Contoh Potongan Kode StrictMode

```
6 1 import { StrictMode } from 'react';
 2 import { createRoot } from 'react-dom/client';
 3
 4 const root = createRoot(document.getElementById('root'));
 5 root.render(
 6   <StrictMode>
 7     <App />
 8   </StrictMode>
 9 );
10
```

¹⁷ 2.3 Ionic 7 Framework [2]

¹⁸ Ionic 7 adalah sebuah *framework* untuk membangun aplikasi *mobile hybrid* menggunakan HTML5,
¹⁹ CSS, dan JavaScript. Ionic 7 mendukung Angular 14+, React 17+, dan Vue 3.0.6+. *Framework* ini
²⁰ dapat digunakan secara gratis dan juga bersifat *open-source*, baik digunakan oleh pribadi maupun
²¹ komersial.

22 2.3.1 UI Components

UI Components adalah kumpulan komponen yang digunakan untuk membangun antarmuka pengguna aplikasi *mobile hybrid*. Komponen-komponen ini memungkinkan pengembang untuk membangun antarmuka pengguna yang menarik dan responsif dengan cepat. Beberapa komponen yang terdapat pada Ionic 7 diantaranya:

27 Action Sheet

28 Action Sheet merupakan sebuah komponen yang berguna untuk memunculkan dialog. Dialog
29 tersebut akan melakukan pemberhentian sementara terhadap aplikasi yang sedang dijalankannya
30 dan pengguna harus memilih pilihan yang berada di dalam dialog tersebut. Cara penggunaan dari
31 Action Sheet terdapat pada (Kode 2.11).

Kode 2.11: Contoh Potongan Kode Action Sheet

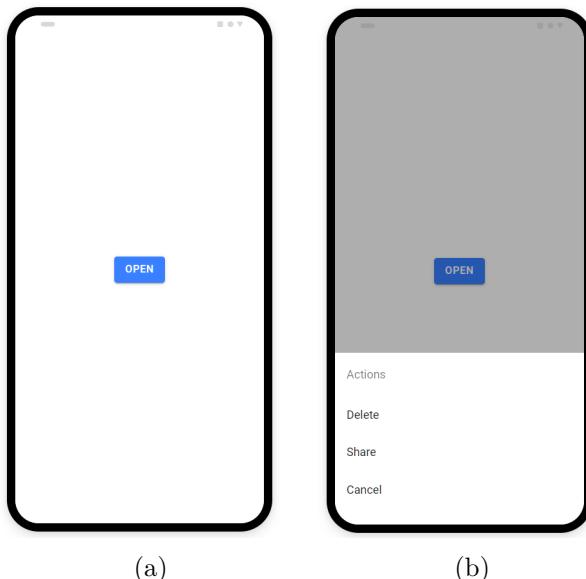
```
32
33 1 import React from 'react';
34 2 import { IonActionSheet, IonButton } from '@ionic/react'
35 3
36 4 function Example() {
37 5   return (
38 6     <>
39 7       <IonButton id="open-action-sheet">Open</IonButton>
40 8       <IonActionSheet
41 9         trigger="open-action-sheet"
42 0         header="Actions"
43 1         buttons={[
44 2           {
45 3             text: 'Delete',
46 4             role: 'destructive',
47 5             data: {
48 6               action: 'delete',
49 7             }
49 8           }
49 9         ]}
50 0       </IonActionSheet>
51 1     </>
52 2   )
53 3 }
54 4
```

```

18 },
19 {
20   text: 'Share',
21   data: {
22     action: 'share',
23   },
24 },
25 {
26   text: 'Cancel',
27   role: 'cancel',
28   data: {
29     action: 'cancel',
30   },
31 },
32 ],
33 ></IonActionSheet>
34 </>
35 );
36 }
37 export default Example;

```

22 Jika kode tersebut dijalankan, hasil yang ditampilkan dari Action Sheet yaitu (Gambar 2.1).



Gambar 2.1: (a) Halaman yang hanya berisi tombol ‘open’, dan (b) Halaman Action Sheet ketika tombol ‘open’ diklik

23 Properti ‘role’ pada Action Sheet adalah sebuah properti yang diberikan untuk memberikan *role*
24 ‘cancel’ atau ‘destructive’ pada tombol yang berada di dalam Action Sheet. Nilai ‘cancel’ digunakan
25 untuk tombol yang akan membatalkan aksi yang dilakukan, sedangkan nilai ‘destructive’ digunakan
26 untuk tombol yang akan menghapus atau mengubah data yang ada. Selain properti role, Action
27 Sheet memiliki properti seperti pada (Tabel 2.1).

Tabel 2.1: Tabel Properti dari Action Sheet

No.	Nama Properti	Deskripsi	Nilai Properti
1	animated	Memberikan animasi pada <i>action-sheet</i>	‘true’ atau ‘false’
2	header	Judul untuk <i>action-sheet</i>	String atau undefined
3	backdrop-dissmiss	Menutup <i>action-sheet</i> apabila backdrop diklik	‘true’ atau ‘false’

1 Button

- 2 Button merupakan elemen interaktif yang dapat digunakan dalam berbagai aplikasi untuk menyedi-
 3 akan fitur tombol standar. Cara penggunaan komponen Button terdapat pada (Kode 2.12).

Kode 2.12: Contoh Potongan Kode Button

```
4 1 <IonButton>Open</IonButton>
```

7 Pada komponen Button terdapat properti ‘expand’ dan ‘icons’. Secara *default*, komponen
 8 Button memiliki *style display: inline-block* dan tidak memiliki ikon. Namun dengan properti
 9 *expand*, *style display* pada komponen Button dapat diubah dengan cara memberikan properti ‘expand’ pada
 10 komponen Button seperti `<IonButton expand="block">Block</IonButton>`. Properti ikon juga
 11 dapat ditambahkan pada awal, akhir, ataupun hanya terdapat ikon pada Button tersebut, seperti
 12 pada (Kode 2.13). Saat menambahkan Icon, pengembang bisa mengikuti cara yang terdapat pada
 13 subbab 2.3.1.

Kode 2.13: Contoh Potongan Kode Button Menggunakan Icon

```
14 1 <IonButton>
15 2   <IonIcon slot="start" icon={star}></IonIcon>
16 3     Left Icon
17 4 </IonButton>
```

20 Nilai dari ‘slot’ dapat berupa ‘start’ untuk menempatkan ikon di awal Button, ‘end’ untuk
 21 menempatkan ikon di akhir button, ataupun ‘icon-only’ untuk memberikan ikon saja pada button.

22 Card

23 Card merupakan komponen yang digunakan untuk menampilkan konten seperti teks, gambar,
 24 tombol, dan daftar dalam sebuah kotak. Komponen ini biasanya terdiri dari *header*, judul, gambar,
 25 dan konten utama. Card dapat digunakan sebagai komponen tunggal atau digabungkan dengan
 26 komponen lain untuk membuat tampilan yang lebih kompleks. Card dapat disesuaikan dengan
 27 menggunakan properti CSS seperti ‘background’ dan ‘color’. Cara menggunakan Card terdapat
 28 pada contoh (Kode 2.14).

Kode 2.14: Contoh Potongan Kode Card

```
29 1 import React from 'react';
30 2 import { IonCard, IonCardContent, IonCardHeader, IonCardSubtitle, IonCardTitle } from '@ionic/react';
31 3
32 4 function Example() {
33 5   return (
34 6     <IonCard>
35 7       <IonCardHeader>
36 8         <IonCardTitle>Card Title</IonCardTitle>
37 9         <IonCardSubtitle>Card Subtitle</IonCardSubtitle>
38 0       </IonCardHeader>
39 1
39 2       <IonCardContent>Here's a small text description for the card content. Nothing more, nothing less.</IonCardContent>
39 3     </IonCard>
39 4   );
39 5 }
39 6 export default Example;
```

47 Pengembang juga bisa menggunakan Card sebagai Media Card dengan menambahkan elemen
 48 `` pada Card tersebut ataupun Button Card dengan menambahkan `<IonButton>` pada Card
 49 tersebut.

1 Content

Content merupakan komponen yang berguna untuk menyediakan area konten yang dapat dikontrol dan diubah menggunakan CSS. Dalam satu tampilan hanya terdapat satu konten. Konten dan komponen Ionic lainnya dapat dikostumisasi ulang dengan menggunakan CSS yang tersedia. Cara menggunakan Card terdapat pada contoh (Kode 2.15).

Kode 2.15: Contoh Potongan Kode Content

```
6 import React from 'react';
7 import { IonContent } from '@ionic/react';
8
9 function Example() {
10   return (
11     <IonContent className="ion-padding">
12       <h1>Heading 1</h1>
13
14       <p>Here's a small text description for the content. Nothing more, nothing less.</p>
15     </IonContent>
16   );
17 }
18
19 export default Example;
```

21 Pada Content, komponen ini dapat ditambahkan header, footer. Lalu komponen content ini juga
22 dapat berupa Fixed Content dan juga Fullscreen Content. Secara *default*, Content akan memenuhi
23 *header* dan *footer*.

24 Grid

25 Komponen Grid adalah sistem *layout* yang menggunakan *flexbox* untuk membangun *layout* yang
26 fleksibel dan responsif. Sistem Grid ini berguna untuk mengatur ruang antara elemen pada sebuah
27 kontainer secara dinamis berdasarkan ukuran layar dan *device* yang berbeda. Pengguna dapat
28 mengatur sendiri nilai Grid yang diinginkan. Nilai dari Grid tersebut adalah rentang angka dari 1
29 hingga 12.

30 Icon

31 Komponen Icon adalah elemen dasar yang tersedia melalui *library* Ionicons. Secara *default*, Icon
32 dipersiapkan dengan semua aplikasi Ionic Framework. Komponen ini dapat digunakan untuk
33 menampilkan ikon dari set Ionicons atau ikon yang berupa SVG. Selain itu, komponen ini juga
34 mendukung pengaturan seperti ukuran dan warna. Cara penggunaan Icon terdapat pada (Kode
35 [2.16](#)).

Kode 2.16: Contoh Kode Penggunaan Icon

```
36
37 1 import React from 'react';
38 2 import { IonIcon } from '@ionic/react';
39 3 import { logoIonic } from 'ionicons/icons';
40 4
41 5 function Example() {
42 6   return (
43 7     <>
44 8       <IonIcon icon={logoIonic}></IonIcon>
45 9       <IonIcon icon={logoIonic} size="large"></IonIcon>
46 0       <IonIcon icon={logoIonic} color="primary"></IonIcon>
47 1       <IonIcon icon={logoIonic} size="large" color="primary"></IonIcon>
48 2     </>
49 3   );
50 4 }
51 5 export default Example;
```

- Untuk Icon yang diinginkan, pengembang bisa menggunakan Icon dari Ionic yang bisa didapatkan di ionic.io/ionicons ataupun bisa menambahkannya sendiri dengan cara melakukan *import* pada Icon yang dimiliki.

4 List

- List digunakan untuk menampilkan data dalam bentuk baris, seperti daftar kontak, daftar putar, atau menu. Komponen ini mendukung berbagai macam interaksi, termasuk menggeser *item* untuk menampilkan opsi, menarik untuk menyusun ulang *item* dalam daftar, dan menghapus *item*. Komponen ini dapat ditambah berbagai elemen ke dalam daftar, seperti teks, tombol, ikon, dan gambar dengan ukuran yang kecil. Cara penggunaan List terdapat pada (Kode 2.17).

Kode 2.17: Contoh Kode Penggunaan List

```

10 import React from 'react';
11 import { IonItem, IonLabel, IonList } from '@ionic/react';
12
13
14 function Example() {
15   return (
16     <IonList>
17       <IonItem>
18         <IonLabel>Pokemon Yellow</IonLabel>
19       </IonItem>
20       <IonItem>
21         <IonLabel>Mega Man X</IonLabel>
22       </IonItem>
23       <IonItem>
24         <IonLabel>The Legend of Zelda</IonLabel>
25       </IonItem>
26       <IonItem>
27         <IonLabel>Pac-Man</IonLabel>
28       </IonItem>
29       <IonItem>
30         <IonLabel>Super Mario World</IonLabel>
31       </IonItem>
32     </IonList>
33   );
34 }
35 export default Example;

```

- Komponen ini dapat ditambahkan properti ‘inset’ untuk memberikan *margin* pada tiap tepi, ‘lines’ untuk memberikan *border* bawah, dan juga ‘mode’ untuk menentukan *style* platform apa yang diinginkan. Secara *default*, properti ‘inset’ memiliki nilai ‘false’, properti ‘lines’ memiliki nilai ‘undefined’, dan juga properti ‘mode’ memiliki nilai ‘undefined’. Pengembang dapat mengubah nilai dari properti tersebut dengan nilai yang berada pada (Tabel 2.2)

Tabel 2.2: Tabel Properti dari List

No.	Nama Properti	Atribut	Nilai Properti
1	inset	inset	‘true’ atau ‘false’
2	lines	lines	‘full’ atau ‘inset’ atau ‘none’
3	mode	mode	‘ios’ atau ‘md’

42 Menu

- Menu merupakan *navigation drawer* yang meluncur dari sisi tampilan saat dijalankan. Komponen ini adalah pola navigasi umum yang dapat secara permanen ditampilkan di layar saat diperlukan. Komponen Menu dapat digunakan untuk membuat tata letak aplikasi yang lebih terstruktur dan

- 1 meningkatkan pengalaman pengguna dengan menyediakan akses mudah ke berbagai bagian dari aplikasi. Cara penggunaan Menu terdapat pada (Kode 2.18).

Kode 2.18: Contoh Kode Penggunaan Menu

```

3  import React from 'react';
4  import { IonButtons, IonContent, IonHeader, IonMenu, IonMenuButton, IonPage, IonTitle, IonToolbar } from '@ionic/react';
5  function Example() {
6      return (
7          <>
8              <IonMenu contentId="main-content">
9                  <IonHeader>
10                     <IonToolbar>
11                         <IonTitle>Menu Content</IonTitle>
12                     </IonToolbar>
13                 </IonHeader>
14                 <IonContent className="ion-padding">This is the menu content.</IonContent>
15             </IonMenu>
16             <IonPage id="main-content">
17                 <IonHeader>
18                     <IonToolbar>
19                         <IonButtons slot="start">
20                             <IonMenuButton></IonMenuButton>
21                         </IonButtons>
22                         <IonTitle>Menu</IonTitle>
23                     </IonToolbar>
24                 </IonHeader>
25                 <IonContent className="ion-padding">Tap the button in the toolbar to open the menu.</IonContent>
26             </IonPage>
27         </>
28     );
29 }
30
31 export default Example;

```

33 Pada kode tersebut, halaman akan membuka *side menu* ketika dipanggil. Tombol menu akan
 34 berada di kiri dikarenakan pada komponen **IonButtons**, tombol **IonMenuButtons** memiliki properti
 35 ‘slot’ yang bernilai ‘start’. Secara *default*, letak komponen *side menu* terletak di kiri halaman.
 36 Letak dari *side menu* dapat diubah dengan cara memberikan properti ‘side’ dengan nilai ‘end’ pada
 37 komponen **IonMenu**.

38 Router

39 Router digunakan untuk menangani *routing* di dalam sebuah proyek JavaScript Vanilla dan Stencil.
 40 Komponen ini mengontrol semua interaksi dengan riwayat browser dan menggabungkan pem-
 41 baruan melalui **ion-router-outlet**. Ion-router menggunakan sintaksis deklaratif menggunakan
 42 JSX/HTML untuk mendefinisikan pohon rute. Selain itu, **ion-router** juga memiliki lifecycle
 43 hooks dan properti untuk mengatur perilaku animasi transisi komponen. Router tidak akan per-
 44 nah menyentuh DOM. Untuk menggunakan Router, pengembang dapat menggunakan **<Route>**
 45 **</Route>**

46 Toolbar

47 Toolbar merupakan komponen yang digunakan untuk menampilkan judul, tombol, ikon, tombol
 48 kembali, tombol menu, kotak pencarian, segmen, dan indikator proses di aplikasi. Toolbars umumnya
 49 ditempatkan di atas atau di bawah konten dan menyediakan konten dan tindakan untuk layar saat
 50 ini. Ketika toolbar ditempatkan di *header*, Toolbar akan muncul di bagian atas konten, sedangkan
 51 jika ditempatkan di *footer*, Toolbar akan muncul di bagian bawah. Cara menggunakan Toolbar
 52 terdapat pada (Kode 2.19).

Kode 2.19: Contoh Kode Penggunaan Toolbar

```

1 import React from 'react';
2 import { IonFooter, IonHeader, IonTitle, IonToolbar } from '@ionic/react';
3
4 function Example() {
5   return (
6     <>
7       <IonHeader>
8         <IonToolbar>
9           <IonTitle>Header Toolbar</IonTitle>
10        </IonToolbar>
11      </IonHeader>
12
13      <IonFooter>
14        <IonToolbar>
15          <IonTitle>Footer Toolbar</IonTitle>
16        </IonToolbar>
17      </IonFooter>
18    </>
19  );
20}
21 export default Example;

```

24 Pada kode tersebut, Header Toolbar akan diletakan pada bagian *header* dan Footer Toolbar
 25 akan diletakan pada bagian *footer*.

26 2.3.2 Capacitor Native

27 Native merupakan kemampuan untuk menambahkan fungsionalitas perangkat asli ke dalam aplikasi
 28 menggunakan *plugin* API untuk Swift pada iOS, Java pada Android, dan JavaScript untuk web.
 29 Dengan menggunakan *plugin* ini, pengembang dapat membuat pengalaman “*native*” yang disesuaikan
 30 dengan mudah. Ionic menyediakan Capacitor sebagai sebuah *runtime native* yang memungkinkan
 31 menambahkan fungsionalitas perangkat asli ke dalam aplikasi.

32 Camera

33 Plugin Camera pada Ionic 7 adalah sebuah plugin yang memungkinkan pengguna untuk mengambil
 34 foto dengan kamera atau memilih foto yang sudah ada dari album foto. Plugin ini dapat di-
 35 *install* dengan perintah `npm install @capacitor/camera` dan `npx cap sync`, serta menambahkan
 36 beberapa izin pada file ‘Info.plist’ untuk iOS dan ‘AndroidManifest.xml’ untuk Android. Selain
 37 itu, *plugin* ini juga memerlukan PWA Elements agar dapat berfungsi. Cara untuk meng-*install*
 38 Camera terdapat pada (Kode 2.20), sedangkan cara untuk menggunakan Camera terdapat pada
 39 contoh (Kode 2.21).

Kode 2.20: Kode untuk Menginstal Plugin Camera

```

40
41 npm install @capacitor/camera

```

Kode 2.21: Contoh Kode Capacitor Camera

```

43
44 import { Camera, CameraResultType } from '@capacitor/camera';
45
46 const takePicture = async () => {
47   const image = await Camera.getPhoto({
48     quality: 90,
49     allowEditing: true,
50     resultType: CameraResultType.Uri
51   });
52
53   // image.webPath will contain a path that can be set as an image src.
54   // You can access the original file using image.path, which can be
55   // passed to the Filesystem API to read the raw data of the image,
56   // if desired (or pass resultType: CameraResultType.Base64 to getPhoto)

```

```

14 var imageUrl = image.webPath;
15
16 // Can be set to the src of an image now
17 imageElement.src = imageUrl;
18 };

```

7 Pada kode tersebut, *variable* ‘image’ akan menggunakan metode `getPhoto()` di mana metode
 8 ini berfungsi untuk mengambil sebuah gambar dari album atau menangkap gambar menggunakan
 9 kamera. Properti `quality` merupakan kualitas dari gambar yang akan dikembalikan dengan nilai
 10 berikisar dari 0 hingga 100 dalam bentuk JPEG. Properti `allowEditing` berfungsi apabila pengguna
 11 ingin melakukan *crop images* pada gambar. Properti `resultType` berfungsi untuk mengembalikan
 12 data dalam bentuk Base64, DataUrl, dan Uri.

13 Filesystem

14 Filesystem API menyediakan alat seperti NodeJS untuk bekerja dengan *file* pada perangkat.
 15 Pengembang dapat menggunakan *plugin* ini untuk melakukan operasi *file* umum seperti membaca,
 16 tulis, dan mengelola isi direktori. Cara untuk meng-*install* Filesystem terdapat pada (Kode 2.22),
 17 sedangkan cara untuk menggunakan Filesystem terdapat pada contoh (Kode 2.23).

Kode 2.22: Kode untuk Menginstal Plugin Filesystem

```

18
19 1 npm install @capacitor/actionsheet

```

Kode 2.23: Contoh Kode Penggunaan Filesystem

```

21 import { Filesystem, Directory, Encoding } from '@capacitor/filesystem';
22
23 const writeSecretFile = async () => {
24   await Filesystem.writeFile({
25     path: 'secrets/text.txt',
26     data: 'This is a test',
27     directory: Directory.Documents,
28     encoding: Encoding.UTF8,
29   });
30 };
31
32 const readSecretFile = async () => {
33   const contents = await Filesystem.readFile({
34     path: 'secrets/text.txt',
35     directory: Directory.Documents,
36     encoding: Encoding.UTF8,
37   });
38
39   console.log('secrets:', contents);
40 };
41
42 const deleteSecretFile = async () => {
43   await Filesystem.deleteFile({
44     path: 'secrets/text.txt',
45     directory: Directory.Documents,
46   });
47 };
48
49 const readFilePath = async () => {
50   // Here's an example of reading a file with a full file path. Use this to
51   // read binary data (base64 encoded) from plugins that return File URIs, such as
52   // the Camera.
53   const contents = await Filesystem.readFile({
54     path: 'file:///var/mobile/Containers/Data/Application/22A433FD-D82D-4989-8BE6-9FC49DEA20BB/Documents/text.txt',
55   });
56
57   console.log('data:', contents);
58 };
59

```

1 Preference

2 Plugin Preference adalah *plugin* yang berguna untuk menyimpan data sederhana dalam bentuk kunci
 3 atau nilai yang dapat diakses secara bersamaan. Preferences API menyediakan area penyimpanan
 4 data yang mendukung kunci atau nilai untuk aplikasi Ionic. Untuk menggunakan Plugin Preference
 5 dalam aplikasi Ionic, perlu meng-*install plugin* Capacitor Preferences dengan cara yang terdapat
 6 pada (Kode 2.24), sedangkan cara untuk menggunakan Capacitor terdapat pada (Kode 2.25).

Kode 2.24: Kode untuk Menginstal Plugin Preference

```
7 1 npm install @capacitor/preferences
```

Kode 2.25: Contoh Kode Plugin Preference

```
10 import { Preferences } from '@capacitor/preferences';
11 1
12 2
13 3 const setName = async () => {
14 4   await Preferences.set({
15 5     key: 'name',
16 6     value: 'Max',
17 7   });
18 8 };
19 9
20 0 const checkName = async () => {
21 1   const { value } = await Preferences.get({ key: 'name' });
22 2
23 3   console.log(`Hello ${value}!`);
24 4 };
25 5
26 6 const removeName = async () => {
27 7   await Preferences.remove({ key: 'name' });
28 8 };
```

30 2.4 RSS

31 RSS merupakan singkatan dari Really Simple Syndication adalah format sindikasi konten web yang
 32 merupakan dialek dari XML. Semua file RSS harus disesuaikan dengan spesifikasi XML 1.0 yang
 33 dipublikasikan oleh World Wide Web Consortium (W3C). Dokumen RSS terdiri dari elemen <rss>
 34 yang memiliki atribut wajib bernama version untuk menentukan versi RSS yang digunakan. Di
 35 bawah elemen <rss> terdapat elemen <channel> yang berisi informasi tentang saluran (metadata)
 36 dan isinya seperti pada (Tabel 2.3).

Tabel 2.3: Tabel Elemen RSS

Element	Deskripsi	Contoh
title	Nama dari <channel>	GoUpstate.com News Headlines
link	URL dari web HTML yang berhubungan dengan <channel> tersebut	http://www.goupstate.com/
description	Frasa atau kalimat yang menjelaskan <channel>	The latest news from GoUpstate.com, a Spartanburg Herald-Journal Web site.

1

BAB 3

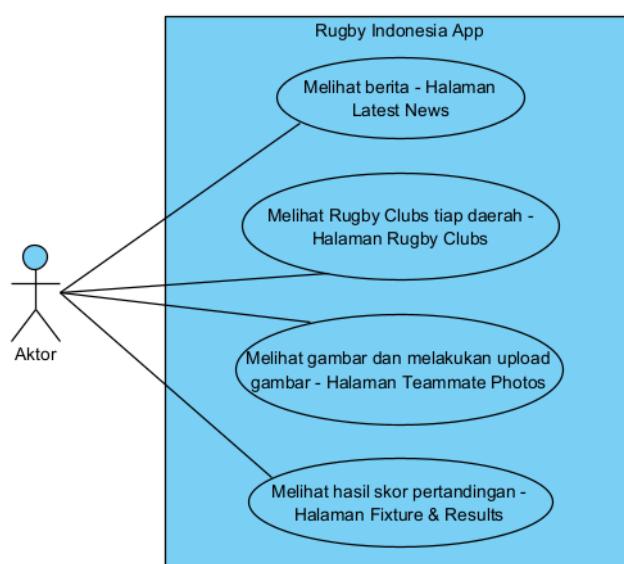
2

ANALISIS

- 3 Bab analisis ini akan menjelaskan tentang analisis dari aplikasi Rugby Indonesia yang ada serta
4 analisis dari aplikasi Rugby Indonesia yang akan dibangun. Analisis yang akan dibahas pada bab
5 ini meliputi Use Case Diagram, analisis kebutuhan sistem kini serta analisis sistem usulan.

6 3.1 Analisis Sistem Kini

- 7 Analisis sistem kini yang dilakukan pada subbab ini dilakukan dengan cara menganalisis aplikasi
8 Rugby Indonesia berdasarkan hasil tangkapan layar yang terdapat pada Google Play Store. Hal
9 ini dilakukan karena peneliti tidak memiliki kode dari aplikasi yang sudah ada. Aplikasi Rugby
10 Indonesia yang ada pada Google Play Store memiliki halaman utama yang berisi berita-berita terkini
11 terkait permainan *rugby* yang ada di Indonesia, *toolbar* yang berisikan tombol *home* untuk kembali
12 ke halaman utama, dan tombol menu untuk berpindah ke menu lainnya. Selain berfungsi untuk
13 membaca berita, aplikasi yang ada ini berfungsi untuk menangkap gambar dan meng-*upload*-nya
14 pada halaman *Teammate Photo*, melihat *Rugby Clubs* yang berada di tiap daerah, serta dapat
15 melihat skor dari tiap pertandingan pada halaman *Fixture & Results*. Fungsi dari aplikasi Rugby
16 Indonesia dapat dilihat pada (Gambar 3.1).



Gambar 3.1: Use Case Diagram Rugby Indonesia App

1 3.1.1 Halaman Utama

- 2 Pada halaman Utama dari Rugby Indonesia, terdapat berita-berita terkini yang diambil dari
 3 halaman rugbyindonesia.or.id/berita/, tombol ‘*read more*’ untuk melihat berita yang ingin
 4 dibaca lebih lengkap. Halaman ini merupakan halaman awal saat pengguna membuka aplikasi
 5 Rugby Indonesia pertama kali. Skenario penggunaan awal aplikasi Rugby Indonesia terdapat pada
 6 (Tabel 3.1), di mana pada tabel skenario tersebut, saat aplikasi pertama kali dibuka, aplikasi akan
 7 langsung mengarahkan pengguna ke dalam halaman *Latest News*.

Tabel 3.1: Tabel Skenario Awal dari Aplikasi Rugby Indonesia

No.	Aksi Aktor	Reaksi Sistem
1	Membuka aplikasi Rugby Indonesia	Aplikasi Rugby Indonesia menampilkan halaman utama yang berisi berita Rugby Indoensia.
2	Pengguna mengklik tombol home	Aplikasi Rugby Indonesia akan menampilkan halaman utama.
3	Pengguna mengklik tombol menu	Aplikasi Rugby Indonesia akan menampilkan <i>side menu</i> yang berisi menu-menu dari aplikasi Rugby Indonesia.
4	Pengguna mengklik tombol ‘ <i>read more</i> ’	Aplikasi Rugby Indonesia akan menampilkan berita yang dipilih oleh pengguna secara lengkap.

8 3.1.2 Halaman Teammate Photos

- 9 Pada halaman *teammate Photos*, pengguna dapat melihat serta meng-*upload* gambar maupun
 10 tangkapan gambar yang ingin diunggah ke dalam halaman ini. Pengguna juga dapat memberikan
 11 *frame* terhadap foto yang ingin diunggahnya. Skenario pengguna saat berada pada halaman
 12 Teammate Photos terdapat pada (Tabel 3.2).

Tabel 3.2: Tabel Skenario dari Halaman Teammate Photos

No.	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan menu Teammate Photos pada <i>side menu</i> .	Aplikasi Rugby Indonesia menampilkan halaman <i>Teammate Photos</i>
2	Pengguna mengklik tombol ‘ <i>take a photo</i> ’	Aplikasi akan membuka kamera
3	Pengguna mengklik tombol ‘ <i>load from library</i> ’	Aplikasi akan membuka <i>library</i> gambar
4	Pengguna mengklik gambar yang ada	Aplikasi akan menampilkan gambar tersebut lebih besar.

13 3.1.3 Halaman Rugby Clubs

- 14 Pada halaman Rugby Clubs, pengguna dapat melihat informasi terkait klub *rugby* yang berada di
 15 tiap daerah. Skenario pengguna saat berada pada halaman Rugby Clubs terdapat pada (Tabel 3.3).

Tabel 3.3: Tabel Skenario dari Halaman Rugby Clubs

No.	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan menu Rugby Clubs pada <i>side menu</i> .	Aplikasi Rugby Indonesia menampilkan halaman Rugby Clubs

¹ 3.1.4 Halaman Fixture & Results

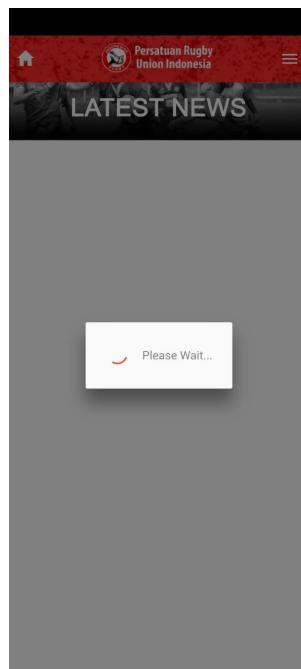
- ² Pada halaman Fixture & Results, pengguna dapat melihat hasil pertandingan skor dari permainan
³ rugby yang sudah diselenggarakan. Skenario pengguna saat berada pada halaman Fixture & Results
⁴ terdapat pada (Tabel 3.4).

Tabel 3.4: Tabel Skenario dari Halaman Fixture & Results

No.	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol Fixture & Results pada side menu.	Aplikasi Rugby Indonesia menampilkan halaman Fixture & Results
2	Pengguna mengklik salah satu pertandingan yang ada	Aplikasi akan memperlihatkan skor dari hasil pertandingan tersebut

⁵ 3.2 Analisis Masalah Aplikasi Rugby Indonesia

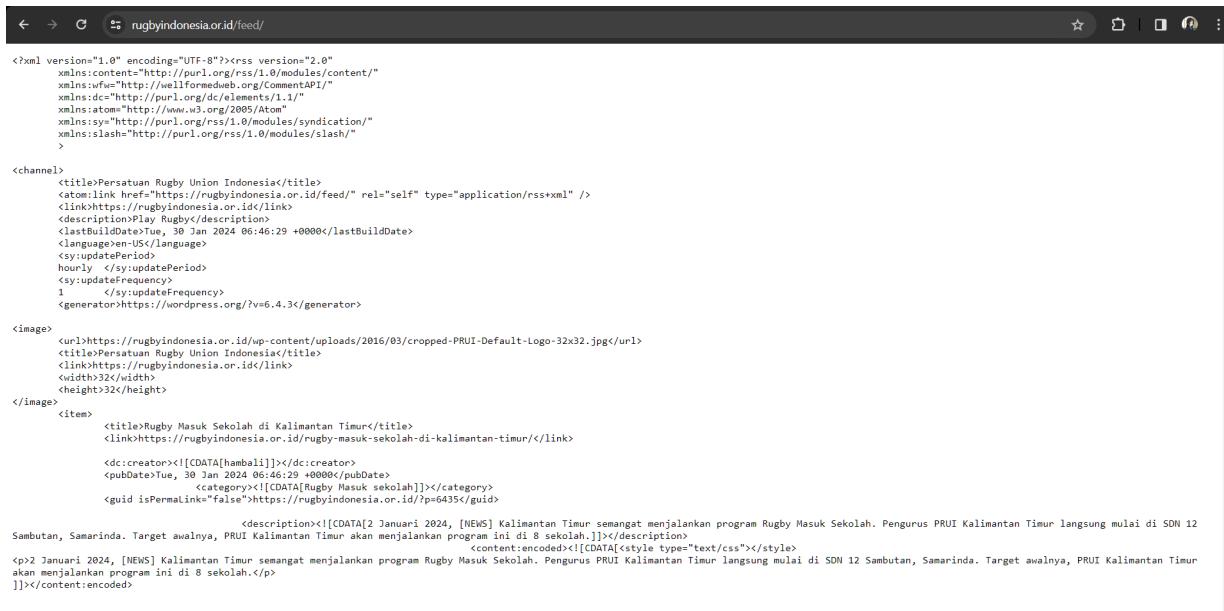
- ⁶ Aplikasi Rugby Indoensia saat ini menggunakan *framework* Ionic versi 3, Apache Cordova, dan
⁷ Angular. *Framework* yang digunakan oleh aplikasi Rugby Indonesia saat ini sudah cukup lawas,
⁸ sehingga aplikasi Rugby Indonesia saat ini sudah tidak dapat dibuka kembali. Ketika membuka
⁹ aplikasi Rugby Indonesia, pengguna akan terhambat pada bagian halaman utama di mana pada
¹⁰ bagian tersebut, pengguna hanya akan melihat *loading circle* yang terus berputar dengan tulisan
¹¹ ‘please wait’ seperti pada (Gambar 3.2).



Gambar 3.2: Halaman Utama Aplikasi Rugby Indonesia

3.3 Analisis RSS Rugby Indonesia

Situs web Rugby Indonesia saat ini memiliki RSS bertipe XML yang terdapat pada <https://dnartworks.rugbyindonesia.or.id/indonesianrugby/news/list.xml>. RSS tersebut dapat digunakan untuk melakukan pembangunan aplikasi Rugby Indonesia dengan cara mengambil beberapa objek seperti title dan description yang terdapat pada RSS tersebut. tangkapan layar RSS dari Rugby Indonesia terdapat pada (Gambar 3.3).



```

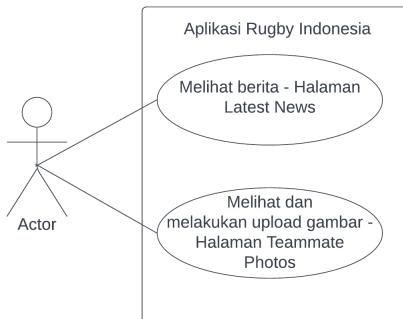
<?xml version="1.0" encoding="UTF-8"?><rss version="2.0"
  xmlns:content="http://purl.org/rss/1.0/modules/content/"
  xmlns:sy="http://wellformedweb.org/CommentAPI/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:slash="http://purl.org/rss/1.0/modules/syndication/"
  xmlns:slash="http://purl.org/rss/1.0/modules/slash/"/>
<channel>
  <title>Persatuan Rugby Union Indonesia</title>
  <atom:link href="https://rugbyindonesia.or.id/feed/" rel="self" type="application/rss+xml" />
  <link>https://rugbyindonesia.or.id</link>
  <description>Rugby Indonesia</description>
  <lastBuildDate>Tue, 30 Jan 2024 06:46:29 +0000</lastBuildDate>
  <language>en-us</language>
  <sy:updatePeriod>
    hourly </sy:updatePeriod>
  <sy:updateFrequency>
    1 </sy:updateFrequency>
  <generator>https://wordpress.org/?v=6.4.3</generator>
</channel>
<image>
  <url>https://rugbyindonesia.or.id/wp-content/uploads/2016/03/cropped-PRUI-Default-Logo-32x32.jpg</url>
  <title>Persatuan Rugby Union Indonesia</title>
  <link>https://rugbyindonesia.or.id</link>
  <width>32</width>
  <height>32</height>
</image>
<item>
  <title>Rugby Masuk Sekolah di Kalimantan Timur</title>
  <link>https://rugbyindonesia.or.id/rugby-masuk-sekolah-di-kalimantan-timur/</link>
  <dc:creator><![CDATA[hambali]]></dc:creator>
  <pubDate>Tue, 30 Jan 2024 06:46:29 +0000</pubDate>
  <category><![CDATA[Rugby Masuk sekolah]]></category>
  <guid isPermaLink="false">https://rugbyindonesia.or.id/?p=6435</guid>
  <description><![CDATA[2 Januari 2024, [NEWS] Kalimantan Timur semangat menjalankan program Rugby Masuk Sekolah. Pengurus PRUI Kalimantan Timur langsung mulai di SDN 12 Sambutan, Samarinda. Target awalnya, PRUI Kalimantan Timur akan menjalankan program ini di 8 sekolah.]]></description>
  <content:encoded><![CDATA[<style type="text/css"></style>
<p>2 Januari 2024, [NEWS] Kalimantan Timur semangat menjalankan program Rugby Masuk Sekolah. Pengurus PRUI Kalimantan Timur langsung mulai di SDN 12 Sambutan, Samarinda. Target awalnya, PRUI Kalimantan Timur akan menjalankan program ini di 8 sekolah.</p>
]]></content:encoded>
</item>

```

Gambar 3.3: Tangkapan Layar RSS Rugby Indonesia

3.4 Analisis Sistem Usulan

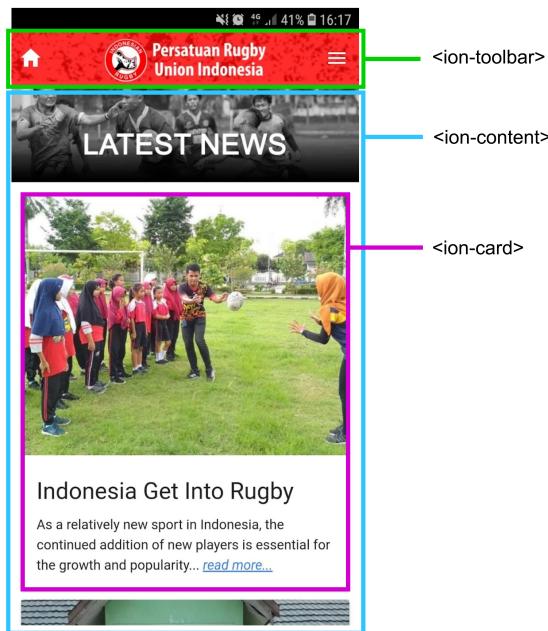
Aplikasi Rugby Indonesia yang akan dibangun hampir sama seperti aplikasi dengan sistem terkini, namun dengan tidak adanya halaman Fixture & Results, Rugby Clubs dan fitur Push Notification dikarenakan website dari Rugby Indonesia sendiri yang sudah dipangkas sehingga tidak memiliki halaman-halaman tersebut. Maka dari itu, fungsi dari aplikasi Rugby Indonesia akan berubah, dan dapat dilihat seperti pada (Gambar 3.4).



Gambar 3.4: UCD Sistem Usulan Rugby Indonesia

1 3.4.1 Halaman Utama

- 2 Pada halaman utama, skenario dari penggunaan aplikasi Rugby Indoensia pada halaman utama
 3 sama seperti tabel skenario halaman utama dari sistem kini pada (Tabel 3.1). Halaman ini memiliki
 4 beberapa komponen yang terdapat pada Ionic 7, komponen tersebut berupa Toolbar, Content, dan
 5 Card. Analisis dari komponen yang digunakan terdapat pada (Gambar 3.5).



Gambar 3.5: Analisis dari Halaman Utama Rugby Indonesia

6 3.4.2 Halaman Teammate Photos

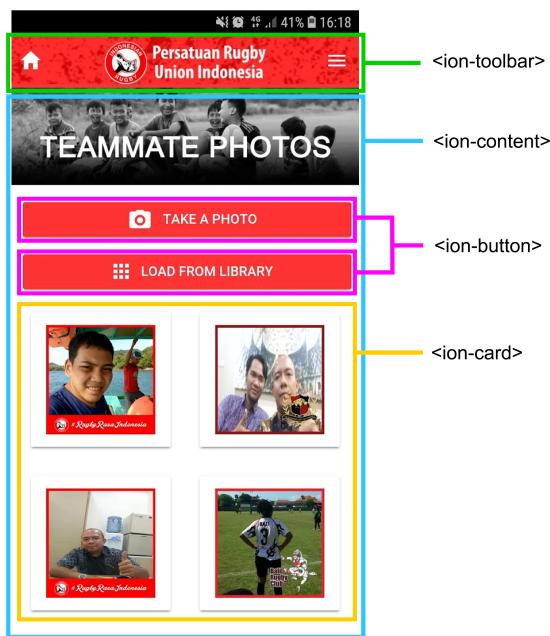
- 7 Pada halaman Teammate Photos, skenario dari aplikasi Rugby Indonesia sama dengan skenario
 8 dari halaman Teammate Photos yang sudah ada pada (Tabel 3.2), namun dengan adanya sedikit
 9 penambahan, yaitu skenario di saat pengguna mengupload gambar. Penambahan ini dilakukan
 10 karena peneliti tidak menemukan gambar di saat pengguna melakukan *upload* gambar. Penambahan
 11 dari skenario disaat pengguna melakukan upload gambar terdapat pada (Tabel 3.5).

Tabel 3.5: Tabel Skenario saat Pengguna Mengupload Gambar

No.	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol “Take a Photo”.	Aplikasi akan membuka kamera pada <i>smartphone</i> .
2	Pengguna mengambil gambar dari kamera.	Aplikasi akan menampilkan Cards yang berisi frame
3	Pengguna memilih salah satu frame lalu mengklik tombol “OK”	Aplikasi akan menampilkan konfirmasi
4	Pengguna mengklik tombol “YA”	Sistem akan mengunggah gambar ke dalam aplikasi.

- 12 Skenario pada (Tabel 3.5) juga berlaku ketika pengguna menekan tombol “LOAD FROM

- 1 LIBRARY”, hanya saja perbedaannya terdapat pada reaksi sistem, yaitu Aplikasi akan membuka *gallery* dari *smartphone* pengguna.
- 2 Pada halaman Teammate Photos, terdapat beberapa komponen dari Ionic 7 yang digunakan.
- 3 Halaman ini memiliki beberapa komponen, yaitu toolbar, button, content, dan juga card. Analisis dari komponen yang digunakan terdapat pada (Gambar 3.6).



Gambar 3.6: Analisis dari Halaman Teammate Photos

DAFTAR REFERENSI

- [1] React Teams (2023) React Reference Overview. <https://react.dev/reference/react>. [Online; diakses 2-Desember-2023].
- [2] Ionic Teams (2023) Introduction to Ionic. <https://ionicframework.com/docs/>. [Online; diakses 29-Oktober-2023].
- [3] Sulistyowati (2010) *Olahraga Rugby*, 1st edition. CV Aneka Ilmu, Jl. Raya Semarang - Demak Km 8,5 Semarang.
- [4] Ionic Teams (2023) Ionic. <https://github.com/ionic-team/ionic-framework>. [Online; diakses 29-Oktober-2023].
- [5] Ionic Teams (2023) Capacitor: Cross-platform Native Runtime for Web Apps. <https://capacitorjs.com/docs>. [Online; diakses 28-November-2023].

LAMPIRAN A

KODE PROGRAM

Kode A.1: MyCode.c

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[1] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = ( --aaa + &daa ) / ( bbb++ - ccc % 2 );
14             strcpy(a,"hello_$.@");
15     }
16     count = ~mask | 0x00FF00AA;
17 }
18
19 // Fonts for Displaying Program Code in LATEX
20 // Adrian P. Robson, nepswb.co.uk
21 // 8 October 2012
22 // http://nepswb.co.uk/docs/progfonts.pdf
23

```

Kode A.2: MyCode.java

```

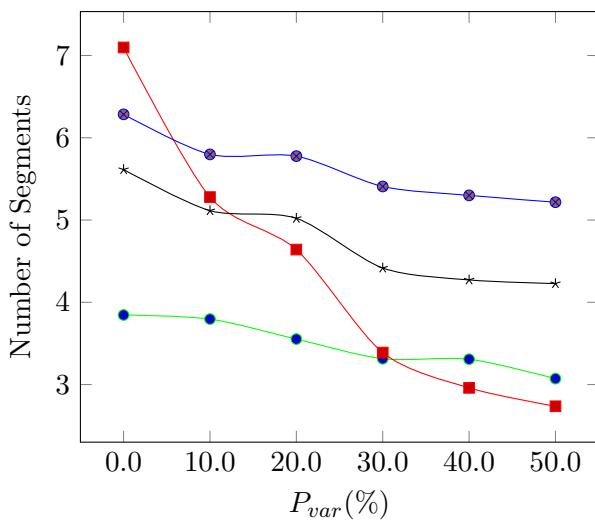
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id;                                //id of the set
8     protected MyEdge FurthestEdge;                   //the furthest edge
9     protected HashSet<MyVertex> set;                //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID;           //store the ID of all vertices
12    protected ArrayList<Double> closeDist;          //store the distance of all vertices
13    protected int totaltrj;                          //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36}

```

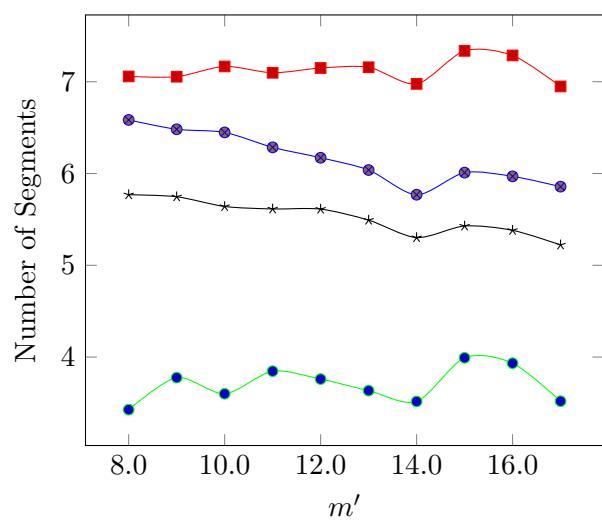

LAMPIRAN B

HASIL EKSPERIMENT

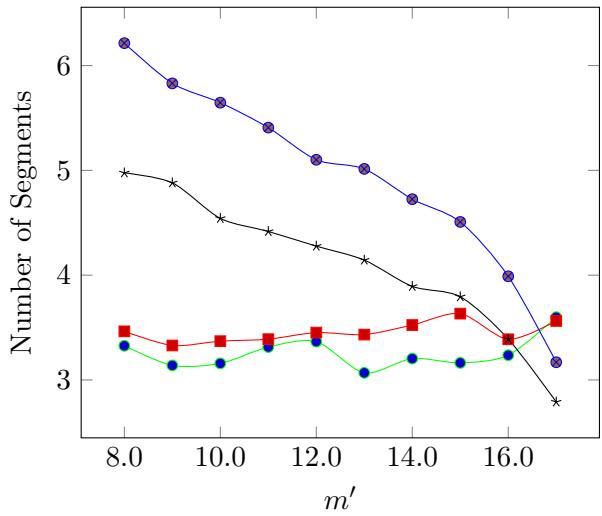
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



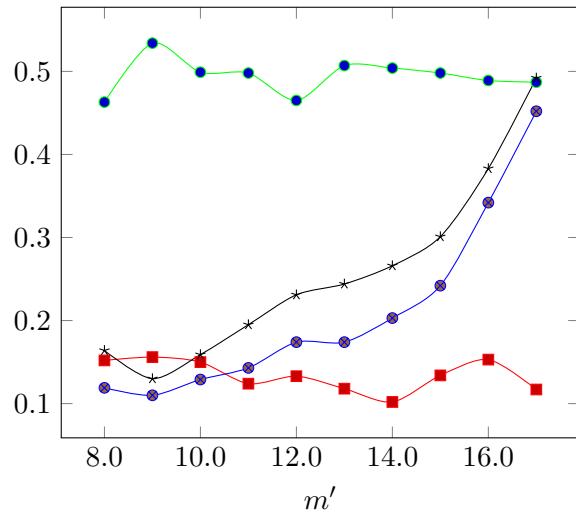
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4