

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	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	4
2 LANDASAN TEORI	5
2.1 Rugby Indonesia App	5
2.2 ReactJS	6
2.2.1 Hooks	6
2.2.2 Components	6
2.3 Ionic 7 Framework	8
2.3.1 UI Components	8
2.3.2 Capacitor Native	13
3 ANALISIS	17
3.1 Analisis Sistem Terkini	17
3.1.1 Halaman Utama	18
3.1.2 Halaman Teammate Photos	18
3.1.3 Halaman Rugby Clubs	18
3.1.4 Halaman Fixture & Results	19
3.2 Analisis Sistem Usulan	19
3.2.1 Halaman Utama	19
3.2.2 Halaman Teammate Photos	22
DAFTAR REFERENSI	27
A KODE PROGRAM	29
B HASIL EKSPERIMEN	31

DAFTAR GAMBAR

1.1 Halaman aplikasi Rugby Indonesia	2
2.1 Halaman aplikasi Rugby Indonesia	5
2.2 Contoh Action Sheet	9
3.1 Use Case Diagram Rugby Indonesia App	17
3.2 UCD Sistem Usulan Rugby Indonesia	19
3.3 Analisis dari Halaman Utama Rugby Indonesia	20
3.4 Analisis dari Halaman Teammate Photos	23
B.1 Hasil 1	31
B.2 Hasil 2	31
B.3 Hasil 3	31
B.4 Hasil 4	31

¹

BAB 1

²

PENDAHULUAN

³ 1.1 Latar Belakang

⁴ *Rugby* adalah olahraga tim yang berasal dari abad ke-19 sebagai variasi dari permainan sepak
⁵ bola. Dalam *rugby*, tujuan dari olahraga ini adalah meletakkan bola di belakang garis *try* lawan.¹
⁶ Olahraga ini dapat dimainkan dengan tangan dan juga tendangan, tetapi pemain hanya boleh
⁷ melempar bola atau diserahkan ke belakang saat dibawa menggunakan tangan.²

⁸ Persatuan Rugby Union Indonesia (PRUI) adalah organisasi yang bertanggung jawab atas
⁹ pengelolaan dan pengembangan Rugby Union di Indonesia.³ Mereka memiliki tim nasional putra
¹⁰ dan putri.⁴ Tim nasional rugby union Indonesia mewakili Indonesia dalam rugby union dan dijuluki
¹¹ "Rhinos". Tim ini adalah anggota penuh World Rugby dan belum pernah bermain di Piala Dunia
¹² Rugby.⁵ Rugby union di Indonesia adalah olahraga minor namun berkembang, yang telah ada selama
¹³ beberapa dekade, dan mengalami fluktuasi dalam kesuksesannya.⁶ Pada tahun 2023, terdapat 35
¹⁴ klub rugby, 700 pemain terdaftar, 45 pelatih terdaftar, dan 23 wasit terdaftar di Indonesia.⁷

¹⁵ Ionic Framework adalah *toolkit UI open-source* untuk membangun aplikasi modern, *cross-platform*
¹⁶ yang berkualitas tinggi dari satu kode sumber dengan JavaScript dan web. Ionic menyediakan alat
¹⁷ dan layanan untuk mengembangkan aplikasi *hybrid mobile*, *desktop*, dan *progressive web* berdasarkan
¹⁸ teknologi dan praktik pengembangan web modern, menggunakan teknologi web seperti CSS, HTML5,
¹⁹ dan Sass. Ionic 7 adalah versi *stable release* terbaru dari Ionic, yang memperkenalkan cara kerja
²⁰ yang lebih efisien dengan kontrol formulir seperti *Toggle* atau *Input*. Komponen *Item* dan *Label*
²¹ tidak lagi diperlukan, dan setiap kontrol formulir menangani konten label secara langsung. Selain
²² itu, fitur tertentu seperti teks bantuan atau mode pengisian *input* telah dipindahkan dari *ion-item*
²³ ke kontrol formulir yang sesuai seperti *ion-input*, *ion-textarea*, dan *ion-select*. Perubahan ini
²⁴ mengurangi *boilerplate* kode dengan menghilangkan persyaratan *ion-item* dan *ion-label*. Komponen
²⁵ Ionic Framework secara otomatis menyesuaikan tampilan dan nuansa mereka dengan platform di
²⁶ mana mereka berjalan, memungkinkan gestur dan perilaku *native* yang sama dengan yang biasa
²⁷ digunakan pengguna. Ionic memiliki lebih dari 100 komponen *UI* yang telah dirancang sebelumnya,
²⁸ tipografi, dan tema dasar yang menyesuaikan dengan setiap platform. Ini dioptimalkan untuk
²⁹ *mobile* dengan animasi yang diakselerasi oleh *hardware*, *lazy loading*, dan *scrolling 60FPS*. Ionic CLI

¹ https://id.wikipedia.org/wiki/Sepak_bola_rugbi

² <https://www.sehataqua.co.id/apa-itu-olahraga-rugby/>

³ <https://www.asiarugby.com/unions/indonesia/>

⁴ <https://rugbyindonesia.or.id/tentang/>

⁵ https://en.wikipedia.org/wiki/Indonesia_national_rugby_union_team

⁶ https://en.wikipedia.org/wiki/Rugby_union_in_Indonesia

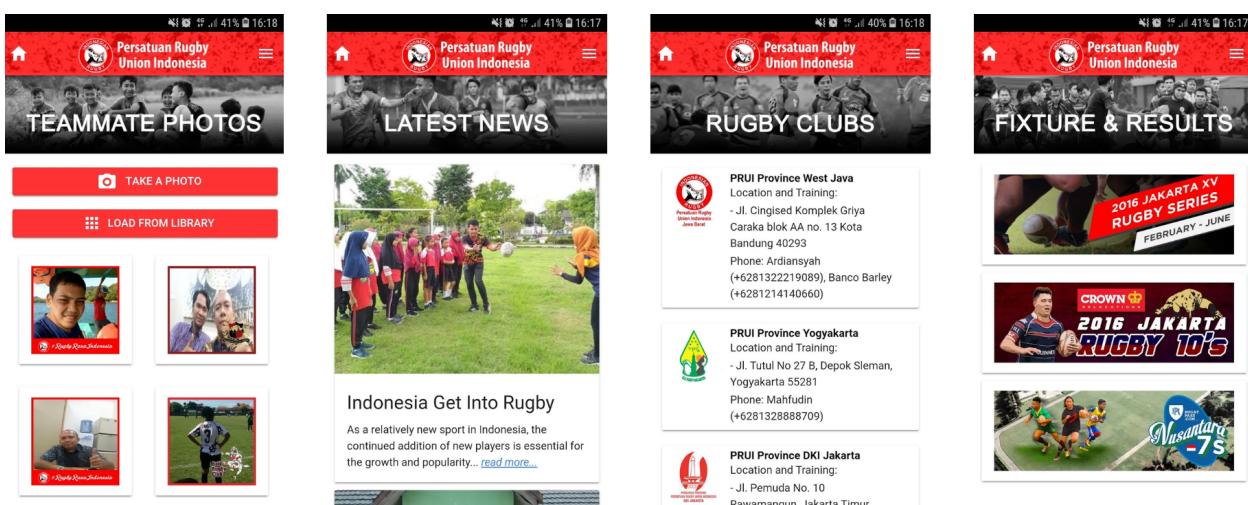
⁷ <https://rugbyindonesia.or.id/>

digunakan untuk membuat, membangun, dan menguji aplikasi serta memanfaatkan *Live Reload*, ⁸ *deployment*, dan dokumen yang baik.

Capacitor adalah *runtime native cross-platform* yang memudahkan pembuatan aplikasi *mobile* yang performanya tinggi dan berjalan secara *native* di iOS, Android, dan platform lainnya menggunakan *web tooling modern*. Capacitor merupakan evolusi selanjutnya dari aplikasi *hybrid*, yang menciptakan aplikasi *Web Native* dengan pendekatan *native container* modern untuk tim yang ingin membangun aplikasi *web-first* tanpa mengorbankan akses penuh ke SDK *native* ketika dibutuhkan. Capacitor menyediakan kumpulan API yang konsisten dan berfokus pada web yang memungkinkan aplikasi tetap dekat dengan standar web sebanyak mungkin, sambil mengakses fitur perangkat *native* yang kaya pada platform yang mendukungnya. Capacitor dapat menambahkan fungsionalitas *native* dengan mudah menggunakan *Plugin API* untuk Swift di iOS, Java di Android, dan JavaScript untuk web. Capacitor 3.0 memiliki peningkatan kinerja, pengalaman untuk mengembangkan yang lebih baik, dan keterlibatan komunitas yang lebih besar. Capacitor dapat diintegrasikan dengan mudah ke dalam proyek JavaScript modern yang ada atau proyek Capacitor yang baru.

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

- Halaman *Latest News* yang diambil dari <https://rugbyindonesia.or.id/berita/> dengan memanfaatkan protokol RSS. (Lihat gambar 1.1)
- Halaman *Fixture & Results*, namun sekarang sudah tidak ada. (Lihat gambar 1.1)
- Halaman *Teammate Photos* dengan fungsi:
 - Pengguna dapat langsung mengambil foto dari aplikasi tersebut.
 - Pengguna dapat langsung memberikan *frame* terhadap foto tersebut.
 - Pengguna dapat langsung mengunggah foto tersebut ke dalam galeri publik.
- Halaman *Rugby Clubs* yang memiliki fungsi di mana pengguna dapat melihat klub *rugby* serta info dari klub tersebut pada tiap daerah. (Lihat gambar 1.1)



Gambar 1.1: Halaman-halaman dari aplikasi Rugby Indonesia

Pada saat ini, aplikasi tersebut masih tersedia di Google Play Store⁹, namun aplikasi tersebut tidak dapat dipasang pada perangkat android saat ini dikarenakan website <https://rugbyindonesia.or.id>.

⁸<https://github.com/ionic-team/ionic-framework>

⁹<https://play.google.com/store/apps/details?id=id.or.rugbyindonesia.androidapp&hl=id>

1 *or.id* sudah berubah dan juga *framework* yang digunakan sudah terlalu lama. Maka dari itu pada
2 skripsi ini, akan dibuat ulang sebuah perangkat lunak Rugby Indonesia yang terbaru, sehingga
3 perangkat lunak tersebut dapat *compatible* dengan perangkat android saat ini.

4 Perangkat lunak ini akan dibuat dengan memanfaatkan bantuan *framework* Ionic 7 dan Capacitor
5 dengan:

- 6 • Halaman *Latest News*, di mana pengguna dapat melihat berita terbaru seputar Rugby Indone-
7 sia.
- 8 • Halaman *Teammate Photos* di mana pengguna dapat mengambil foto dan langsung mengung-
9 gahnya ke dalam halaman *teammate photos*.

10 **1.2 Rumusan Masalah**

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

- 12 1. Bagaimana cara membangun ulang serta mengembangkan perangkat lunak Rugby Indonesia
13 dengan memanfaatkan *framework* Ionic 7?
- 14 2. Bagaimana cara menggunakan Capacitor pada pembangunan perangkat lunak Rugby Indonesia
15 agar pengguna dapat mengunggah foto dengan mudah?

16 **1.3 Tujuan**

17 Tujuan yang ingin dicapai pada penulisan tugas akhir ini yaitu:

- 18 1. Dapat mengetahui bagaimana Ionic 7 memungkinkan pengembangan aplikasi Rugby Indonesia.
- 19 2. Mengidentifikasi cara kerja dari Capacitor pada pembangunan perangkat lunak Rugby Indo-
20 nesia.

21 **1.4 Batasan Masalah**

22 Batasan masalah yang terdapat pada penggerjaan tugas akhir ini yaitu:

- 23 1. Perangkat lunak ini dibuat untuk perangkat Android saja, tidak untuk iOS. Sehingga pengujian
24 dari perangkat lunak ini hanya dilakukan pada platform berbasis android saja. Perangkat
25 lunak ini hanya dibuat untuk perangkat lunak android dikarenakan peneliti tidak memiliki
26 perangkat lunak yang menggunakan sistem operasi iOS.
- 27 2. Pengguna hanya bisa mengunggah foto dan melihat foto unggahan dari pengguna lain,
28 pengguna tidak dapat menghapus ataupun mengubah foto tersebut. Hal ini dikarenakan pada
29 aplikasi sebelumnya, pengguna hanya dapat melakukan hal tersebut.

30 **1.5 Metodologi**

31 Langkah-langkah yang dilakukan dalam penggerjaan tugas akhir ini yaitu:

- 32 1. Melakukan studi literatur serta mendalami ReactJS sebagai salah satu perpustakaan Java-
33 Script.
- 34 2. Melakukan studi literatur mengenai *framework* Ionic 7 dan juga Capacitor yang terdapat
35 pada Ionic.

- 1 3. Melakukan analisis terhadap perangkalt lunak yang ada dan melakukan perancangan terhadap perangkat lunak yang akan dibuat.
- 2 4. Membangun aplikasi Rugby Indonesia yang sudah menggunakan framework Ionic 7 serta Capacitor.
- 3 5. Melakukan pengujian dan eksperimen.
- 4 6. Menulis dokumen tugas akhir.

7 **1.6 Sistematika Pembahasan**

8 Penulisan setiap bab pada dokumen tugas akhir ini yaitu:

9 1. Bab Pendahuluan

10 Bab 1 berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan
11 sistematiska pembahasan yang digunakan untuk menyusun skripsi ini.

12 2. Bab Dasar Teori

13 Bab 2 berisi teori-teori yang digunakan dalam pembuatan skripsi ini. Teori-teori tersebut
14 yaitu Rugby Union Indonesia, ReactJS, Ionic 7 Framework, Capacitor, UI Components.

15 3. Bab Analisis

16 Bab 3 berisi analisis yang dilakukan pada skripsi ini, meliputi analisis sistem kini, analisis
17 kebutuhan aplikasi Rugby Indonesia yang akan dibangun, serta permasalahan pembangunan
18 sistem usulan.

19 4. Bab Perancangan

20 Bab 4 berisi perancangan aplikasi meliputi perancangan kelas beserta dengan diagram kelas,
21 deskripsi kelas dan fungsinya, serta perancangan struktur HTML.

22 5. Bab Implementasi dan Pengujian

23 Bab 5 berisi implementasi dan pengujian aplikasi meliputi lingkungan implementasi, hasil
24 implementasi, pengujian fungsional, dan pengujian eksperimental.

25 6. Bab Kesimpulan dan Saran

26 Bab 6 berisi kesimpulan dari hasil pembangunan aplikasi ini dan saran untuk pengembangan
27 selanjutnya.

1

BAB 2

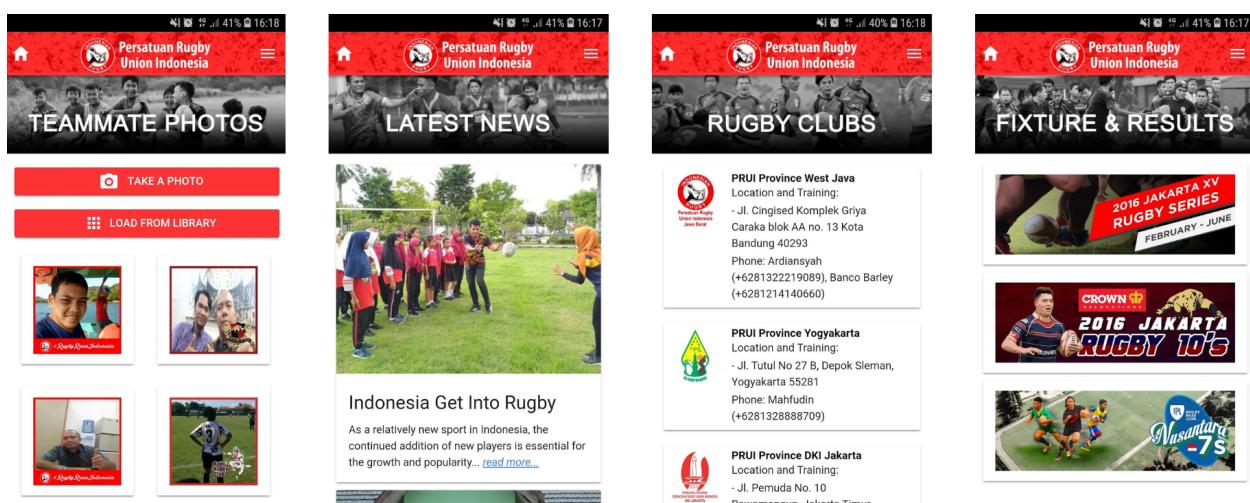
2

LANDASAN TEORI

- 3 Pada bab 2 ini, akan dijelaskan dasar teori terkait dengan aplikasi Rugby Indonesia saat ini, Ionic
4 dan Capacitor.

5 2.1 Rugby Indonesia App

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



Gambar 2.1: Halaman-halaman dari aplikasi Rugby Indonesia

- 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.

- Pengguna dapat langsung memberikan *frame* terhadap foto tersebut.
 - Pengguna dapat langsung mengunggah foto tersebut ke dalam galeri publik.
4. Halaman *Rugby Clubs* yang memiliki fungsi di mana pengguna dapat langsung mendaftar ke dalam *Rugby Clubs* yang berada di Indonesia.
5. Fungsi *Push Notifications*.

6 2.2 ReactJS

7 ReactJS atau React adalah sebuah library JavaScript yang digunakan untuk membangun user
 8 interface yang interaktif. ReactJS berisi kumpulan snippet kode JavaScript yang disebut “komponen”
 9 yang bisa digunakan berulang kali untuk mendesain antarmuka pengguna. ReactJS bukanlah
 10 framework JavaScript, karena hanya bertugas untuk merender komponen area tampilan aplikasi.
 11 ReactJS dapat digunakan untuk membuat aplikasi web dan mobile.

12 2.2.1 Hooks

13 Hooks adalah fitur yang memungkinkan dalam penggunaan state dan fitur React lainnya tanpa
 14 membuat sebuah kelas. Hooks berfungsi untuk “mengaitkan” state dan fitur-fitur lifecycle React
 15 dari function component.

16 State Hooks

17 State Hooks adalah salah satu jenis Hooks yang memungkinkan kita untuk menggunakan state
 18 pada function component tanpa harus membuat sebuah kelas.

19 Berikut adalah cara untuk menggunakan State Hooks:

Kode 2.1: Contoh penggunaan kode State Hooks

```
20 1 function ImageGallery() {
21 2   const [index, setIndex] = useState(0);
22 3   // ...
```

25 Pada kode tersebut, ‘useState’ digunakan untuk mendeklarasikan sebuah variable state yang
 26 dapat langsung di update.

27 2.2.2 Components

28 Components pada React adalah bagian-bagian kecil yang memungkinkan Anda untuk membagi an-
 29 tarmuka pengguna menjadi bagian-bagian independen dan dapat digunakan kembali. Mereka dapat
 30 berupa kelas atau fungsi. Komponen menerima masukan yang disebut “props” dan mengembalikan
 31 elemen React yang mendefinisikan tampilan. Terdapat 4 komponen yang terdapat pada react, yaitu:

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

1 Fragment

2 Fragment pada komponen React adalah fitur yang memungkinkan pengembang untuk mengelom-
 3 pokkan sejumlah elemen anak tanpa perlu menambahkan node ekstra ke DOM. Hal ini berguna
 4 ketika ingin mengembalikan beberapa elemen dari sebuah komponen tanpa harus membungkusnya
 5 dalam sebuah elemen DOM tambahan seperti ‘div’. Penggunaan Fragment dapat membantu
 6 menghindari masalah “div soup” di dalam struktur tampilan. Selain itu, Fragment juga dapat
 7 digunakan untuk merender daftar dalam pembungkus tanpa harus menggunakan ‘<div>’. Ada dua
 8 cara untuk mendefinisikan Fragment, yaitu dengan menggunakan <React.Fragment> atau dengan
 9 menggunakan sintaksis singkat <>...</>. Fragment juga dapat memiliki key dan mendukung
 10 penggunaan atribut. Dengan demikian, penggunaan Fragment membantu dalam memperbaiki
 11 masalah yang muncul saat ingin mengembalikan beberapa elemen dari sebuah komponen React.
 12 Contoh kode saat menggunakan Fragment adalah sebagai berikut:

Kode 2.2: Contoh penggunaan kode Fragment

```
13
14 1  <>
15 2  <OneChild />
16 3  <AnotherChild />
17 4 </>
```

19 Pada kode tersebut, <OneChild /> dan <AnotherChild /> akan terlihat rapi tanpa menggu-
 20 nakan <div>.

21 StrictMode

22 StrictMode adalah sebuah komponen yang digunakan untuk menyoroti potensi masalah dalam sebuah
 23 aplikasi. Mode ini tidak berdampak dalam build produksi dan dapat diaktifkan untuk bagian-bagian
 24 tertentu dalam aplikasi. StrictMode membantu dalam mengidentifikasi komponen-komponen dengan
 25 siklus hidup yang tidak aman dan melakukan berbagai pemeriksaan tambahan untuk turunannya.
 26 Mode ini berguna saat mengembangkan kode baru atau melakukan debugging. StrictMode dapat
 27 diterapkan pada bagian mana pun dalam aplikasi, bukan hanya pada keseluruhan aplikasi. Mode ini
 28 membantu dalam menulis kode React dengan cara yang lebih baik dengan memberikan peringatan
 29 terkait praktik terbaik. Mode ini juga dapat digunakan baik pada komponen fungsional maupun
 30 kelas. Namun, StrictMode merender setiap komponen dalam aplikasi dua kali, sehingga sebaiknya
 31 hanya digunakan saat pengembangan atau debugging. Contoh kode saat menggunakan StrictMode
 32 adalah sebagai berikut:

Kode 2.3: Contoh penggunaan kode StrictMode

```
33
34 1 import { StrictMode } from 'react';
35 2 import { createRoot } from 'react-dom/client';
36 3
37 4 const root = createRoot(document.getElementById('root'));
38 5 root.render(
39 6   <StrictMode>
40 7     <App />
41 8   </StrictMode>
42 9 );
```

2.3 Ionic 7 Framework

- 2 Ionic 7 adalah sebuah framework untuk membangun aplikasi mobile hybrid menggunakan HTML5,
3 CSS, dan AngularJS. Framework ini dirilis pada tanggal 29 Maret 2023 dan memiliki beberapa
4 perbaikan yang diusulkan oleh komunitas Ionic. Beberapa fitur baru di Ionic 7 antara lain:

- **Inline Overlays:** Cara yang lebih efisien untuk bekerja dengan form-control seperti Toggle atau Input. Komponen Item dan Label tidak lagi diperlukan, dan setiap form-control menangani konten label secara langsung. Perubahan ini mengurangi boilerplate kode dengan menghilangkan persyaratan ion-item dan ion-label.
 - **Performa yang Lebih Baik:** Ionic 7 secara signifikan meningkatkan performa Tabs. Pada Ionic React dan Ionic Vue, pengembang dapat mengharapkan peningkatan performa hingga 70% saat beralih tab. Pengembang Ionic Angular dapat mengharapkan waktu inisialisasi komponen Ionic yang lebih baik berkat optimasi di Stencil.
 - **Kompatibilitas Vite yang Lebih Baik:** Ionic 7 menghapus titik masuk Common JS untuk Ionic React dan Ionic Vue untuk membuat setiap paket lebih mudah digunakan dengan Vite dan Vitest.

¹⁶ Ionic 7 mendukung Angular 14+, React 17+, dan Vue 3.0.6+

¹⁷ 2.3.1 UI Components

¹⁸ 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. Komponen yang terdapat pada ²⁰ Ionic 7 yaitu:

22 Action Sheet

23 Action Sheet (ion-action-sheet) merupakan sebuah komponen yang berguna untuk memunculkan
24 dialog. Dialog tersebut akan melakukan pemberhentian sementara terhadap aplikasi yang sedang
25 dijalankannya dan pengguna harus memilih pilihan yang berada di dalam dialog tersebut. Cara
26 penggunaan dari Action Sheet adalah sebagai berikut:

Kode 2.4: Contoh kode untuk membuat Action Sheet

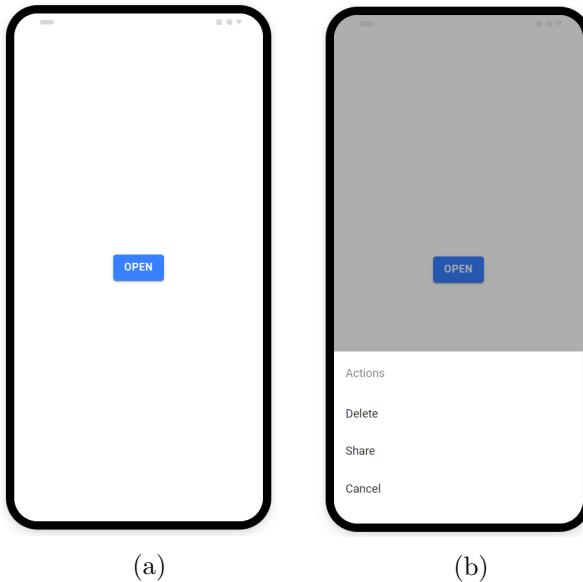
```
27
28 1 import React from 'react';
29 2 import { IonActionSheet, IonButton } from '@ionic/react'
30 3
31 4 function Example() {
32 5   return (
33 6     <>
34 7       <IonButton id="open-action-sheet">Open</IonButton>
35 8       <IonActionSheet
36 9         trigger="open-action-sheet"
37 0         header="Actions"
38 1         buttons={[
39 2           {
40 3             text: 'Delete',
41 4             role: 'destructive',
42 5             data: {
43 6               action: 'delete',
44 7             },
45 8           },
46 9           {
47 0             text: 'Share',
48 1             data: {
49 2               action: 'share',
50 3             }
51 4           }
52 5         ]}
53 6       </IonActionSheet>
54 7     </>
55 8   )
56 9 }
57 0
58 1
```

```

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;

```

17 Jika kode tersebut dijalankan, maka tampilan dari Action Sheet akan sebagai berikut:



Gambar 2.2: (a) Halaman yang hanya berisi tombol open, dan (b) Halaman Action Sheet ketika tombol open diklik

18 Properti role pada Action Sheet adalah sebuah properti yang diberikan untuk memberikan role
 19 ‘cancel’, ‘destructive’, dan ‘selected’ pada tombol yang berada di dalam Action Sheet. Nilai ‘cancel’
 20 digunakan untuk tombol yang akan membatalkan aksi yang dilakukan, sedangkan nilai ‘destructive’
 21 digunakan untuk tombol yang akan menghapus atau mengubah data yang ada. Nilai ‘selected’
 22 digunakan untuk menandai sebuah opsi yang telah dipilih sebelumnya.

23 **Button**

24 Button merupakan elemen interaktif yang dapat digunakan dalam berbagai aplikasi untuk menyedi-
 25 akan fitur tombol standar. Berikut adalah contoh dari penggunaan Button:

Kode 2.5: Contoh kode untuk membuat Button

```

26 <IonButton>Open</IonButton>
27
28

```

29 Pada komponen button terdapat properti “expand” dan “icons”. Secara default, komponen
 30 button memiliki style `display: inline-block` dan tidak memiliki icons. Namun dengan properti
 31 expand, style pada komponen button dapat diubah dengan cara memberikan properti expand pada

- 1 komponen button seperti `<IonButton expand="block">Block</IonButton>`. Properti icons juga
 2 dapat ditambahkan pada awal, akhir, ataupun pada button tersebut hanya terdapat icon saja,
 3 seperti:

Kode 2.6: Contoh kode untuk membuat Button

```
4
5 1 <IonButton>
6 2   <IonIcon slot="start" icon={star}></IonIcon>
7 3   Left Icon
8 4 </IonButton>
```

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

12 Card

13 Card merupakan komponen yang digunakan untuk menampilkan konten seperti teks, gambar,
 14 tombol, dan daftar dalam sebuah kotak. Komponen ini biasanya terdiri dari header, judul, gambar,
 15 dan konten utama. Card dapat digunakan sebagai komponen tunggal atau digabungkan dengan
 16 komponen lain untuk membuat tampilan yang lebih kompleks. Card dapat disesuaikan dengan meng-
 17 gunakan properti CSS seperti background dan color. Berikut merupakan contoh dari penggunaan
 18 Card:

Kode 2.7: Contoh kode untuk membuat Card

```
19
20 1 import React from 'react';
21 2 import { IonCard, IonCardContent, IonCardHeader, IonCardSubtitle, IonCardTitle } from '@ionic/react';
22 3
23 function Example() {
24 5   return (
25 6     <IonCard>
26 7       <IonCardHeader>
27 8         <IonCardTitle>Card Title</IonCardTitle>
28 9         <IonCardSubtitle>Card Subtitle</IonCardSubtitle>
29 0       </IonCardHeader>
30 1
31 2         <IonCardContent>Here's a small text description for the card content. Nothing more, nothing less.</IonCardContent>
32 3       </IonCard>
33 4     );
34 5   }
35 6
36 export default Example;
```

37 Pengembang juga bisa menggunakan card sebagai media card dengan menambahkan properti
 38 `` pada card tersebut ataupun button card dengan menambahkan `<IonButton>` pada card
 39 tersebut.

40 Content

41 Content merupakan komponen yang berguna untuk menyediakan area konten yang dapat dikontrol
 42 dan diubah menggunakan CSS. Dalam satu tampilan hanya terdapat satu konten. Konten dan
 43 komponen Ionic lainnya dapat dikostumisasi ulang dengan menggunakan CSS yang tersedia. Berikut
 44 adalah contoh dari penggunaan Content:

Kode 2.8: Contoh kode untuk membuat Content

```
45
46 1 import React from 'react';
47 2 import { IonContent } from '@ionic/react';
48 3
49 4 function Example() {
50 5   return (
51 6     <IonContent className="ion-padding">
```

```

17  <h1>Heading 1</h1>
28
39      <p>Here's a small text description for the content. Nothing more, nothing less.</p>
40  </IonContent>
51  );
52 }
53 export default Example;

```

9 Pada content, komponen ini dapat ditambahkan header, footer. Lalu komponen content ini
10 juga dapat berupa fixed content dan juga fullscreen content. Scara default, content akan memenuhi
11 header dan footer, sehingga tidak memiliki header dan footer, serta di dalam content tersebut, tidak
12 terdapat properti dalam posisi fixed.

13 Grid

14 Komponen Grid pada Ionic 7 adalah sistem layout yang menggunakan flexbox untuk membangun
15 layout yang fleksibel dan responsif. Grid system ini berguna untuk mengatur ruang antara elemen
16 pada sebuah kontainer secara dinamis berdasarkan ukuran layar dan device yang berbeda. Pengguna
17 dapat mengatur sendiri nilai grid yang diinginkan. Nilai dari grid tersebut adalah rentang angka
18 dari 1 hingga 12.

19 Icon

20 Komponen icon dalam Ionic 7 adalah elemen dasar yang tersedia melalui library Ionicons, yang
21 secara default diperlukan dengan semua aplikasi Ionic Framework. Anda dapat menggunakan icon
22 ini untuk menampilkan icon dari set Ionicons atau SVG custom. Selain itu, komponen ini juga
23 mendukung pengaturan seperti ukuran dan warna. Berikut adalah contoh kode dari penggunaan
24 icon:

Kode 2.9: Contoh kode untuk membuat Icon

```

25 import React from 'react';
26 import { IonIcon } from '@ionic/react';
27 import { logoIonic } from 'ionicons/icons';
28
29
30 function Example() {
31     return (
32         <>
33             <IonIcon icon={logoIonic}></IonIcon>
34             <IonIcon icon={logoIonic} size="Large"></IonIcon>
35             <IonIcon icon={logoIonic} color="primary"></IonIcon>
36             <IonIcon icon={logoIonic} size="large" color="primary"></IonIcon>
37         </>
38     );
39 }
40 export default Example;

```

42 Untuk icon yang diinginkan, pengembang bisa menggunakan icon dari ionic yang bisa didapatkan
43 di ionic.io/ionicons ataupun bisa menambahkannya sendiri dengan cara melakukan import
44 pada icon yang dimiliki.

45 List

46 List digunakan untuk menampilkan data dalam bentuk baris, seperti daftar kontak, daftar putar,
47 atau menu. Komponen ini mendukung berbagai macam interaksi, termasuk menggeser item untuk
48 menampilkan opsi, menarik untuk menyusun ulang item dalam daftar, dan menghapus item.

- 1 Komponen ini dapat ditambah berbagai elemen ke dalam daftar, seperti teks, tombol, ikon, dan gambar dengan ukuran yang kecil. Berikut adalah contoh kode penggunaan list:

Kode 2.10: Contoh kode untuk membuat List

```

31 import React from 'react';
32 import { IonItem, IonLabel, IonList } from '@ionic/react';
33
34 function Example() {
35   return (
36     <IonList>
37       <IonItem>
38         <IonLabel>Pokemon Yellow</IonLabel>
39       </IonItem>
40       <IonItem>
41         <IonLabel>Mega Man X</IonLabel>
42       </IonItem>
43       <IonItem>
44         <IonLabel>The Legend of Zelda</IonLabel>
45       </IonItem>
46       <IonItem>
47         <IonLabel>Pac-Man</IonLabel>
48       </IonItem>
49       <IonItem>
50         <IonLabel>Super Mario World</IonLabel>
51       </IonItem>
52     </IonList>
53   );
54 }
55 export default Example;

```

30 Komponen ini dapat ditambahkan properti ‘inset’ untuk memberikan margin pada tiap tepi, 31 ‘lines’ untuk memberikan border bawah, dan juga ‘mode’ untuk menentukan style platform apa 32 yang diinginkan. Secara default, properti ‘inset’ memiliki nilai false, properti ‘lines’ memiliki nilai 33 undefined, dan juga properti ‘mode’ memiliki nilai undefined.

34 Menu

35 Menu merupakan navigation drawer yang meluncur dari sisi tampilan saat ini. Komponen ini adalah 36 pola navigasi umum yang dapat secara permanen ditampilkan di layar saat diperlukan. Komponen 37 Menu dapat digunakan untuk membuat tata letak aplikasi yang lebih terstruktur dan meningkatkan 38 pengalaman pengguna dengan menyediakan akses mudah ke berbagai bagian dari aplikasi.

Kode 2.11: Contoh kode untuk membuat Menu

```

39 import React from 'react';
40 import { IonButtons, IonContent, IonHeader, IonMenu, IonMenuButton, IonPage, IonTitle, IonToolbar } from '@ionic/react';
41
42 function Example() {
43   return (
44     <>
45       <IonMenu contentId="main-content">
46         <IonHeader>
47           <IonToolbar>
48             <IonTitle>Menu Content</IonTitle>
49           </IonToolbar>
50         </IonHeader>
51         <IonContent className="ion-padding">This is the menu content.</IonContent>
52       </IonMenu>
53     <IonPage id="main-content">
54       <IonHeader>
55         <IonToolbar>
56           <IonButtons slot="start">
57             <IonMenuButton></IonMenuButton>
58           </IonButtons>
59         <IonTitle>Menu</IonTitle>
60       </IonHeader>
61       <IonContent className="ion-padding">Tap the button in the toolbar to open the menu.</IonContent>
62     </IonPage>
63   </>
64 );
65

```

```

27 }
28 export default Example;

```

4 Pada kode tersebut [2.11], halaman akan membuka side menu ketika dipanggil. Tombol menu
 5 akan berada di kiri dikarenakan pada komponen `IonButtons`, tombol `IonMenuButtons` memiliki
 6 properti ‘slot’ yang bernilai ‘start’. Secara default, letak komponen side menu terletak di kiri
 7 halaman. Letak dari side menu dapat diubah dengan cara memberikan properti ‘side’ dengan nilai
 8 ‘end’ pada komponen `IonMenu`.

9 Router

10 Router digunakan untuk menangani routing di dalam sebuah proyek JavaScript vanilla dan Stencil.
 11 Komponen ini mengontrol semua interaksi dengan riwayat browser dan menggabungkan pembaruan
 12 melalui `ion-router-outlet`. `Ion-router` menggunakan sintaks deklaratif menggunakan JSX/HTML
 13 untuk mendefinisikan pohon rute. Selain itu, `ion-router` juga memiliki lifecycle hooks dan properti
 14 untuk mengatur perilaku animasi transisi komponen. Router tidak akan pernah menyentuh DOM.
 15 Untuk menggunakan router, pengembang dapat menggunakan `<Route> </Route>`

16 Toolbar

17 Toolbar merupakan komponen yang digunakan untuk menampilkan judul, tombol, ikon, tombol
 18 kembali, tombol menu, kotak pencarian, segmen, dan indikator progres di aplikasi. Toolbars
 19 umumnya ditempatkan di atas atau di bawah konten dan menyediakan konten dan tindakan untuk
 20 layar saat ini. Ketika toolbar ditempatkan di header, toolbar akan muncul di bagian atas konten,
 21 sedangkan jika ditempatkan di footer, toolbar akan muncul di bagian bawah. Berikut adalah contoh
 22 penggunaan Toolbar:

Kode 2.12: Contoh kode untuk membuat Toolbar

```

23
24 1 import React from 'react';
25 2 import { IonFooter, IonHeader, IonTitle, IonToolbar } from '@ionic/react';
26 3
27 4 function Example() {
28 5   return (
29 6     <>
30 7       <IonHeader>
31 8         <IonToolbar>
32 9           <IonTitle>Header Toolbar</IonTitle>
33 0         </IonToolbar>
34 1       </IonHeader>
35 2
36 3       <IonFooter>
37 4         <IonToolbar>
38 5           <IonTitle>Footer Toolbar</IonTitle>
39 6         </IonToolbar>
40 7       </IonFooter>
41 8     </>
42 9   );
43 0 }
44 1 export default Example;

```

46 Pada kode tersebut, header toolbar akan diletakan pada bagian header dan footer toolbar akan
 47 diletakan pada bagian footer.

48 2.3.2 Capacitor Native

49 Native merupakan kemampuan untuk menambahkan fungsionalitas perangkat asli ke dalam aplikasi
 50 menggunakan plugin API untuk Swift pada iOS, Java pada Android, dan JavaScript untuk web.

- 1 Dengan menggunakan plugin ini, pengembang dapat membuat pengalaman “native” yang disesuaikan
- 2 dengan mudah. Ionic menyediakan Capacitor sebagai sebuah runtime native yang memungkinkan
- 3 menambahkan fungsionalitas perangkat asli ke dalam aplikasi.

4 Camera

- 5 Plugin Camera pada Ionic 7 adalah sebuah plugin yang memungkinkan pengguna untuk mengambil
- 6 foto dengan kamera atau memilih foto yang sudah ada dari album foto. Plugin ini dapat diinstal
- 7 dengan perintah npm install @capacitor/camera dan npx cap sync untuk platform iOS, serta
- 8 menambahkan beberapa izin pada file Info.plist untuk iOS dan AndroidManifest.xml untuk Android.
- 9 Selain itu, plugin ini juga memerlukan PWA Elements agar dapat berfungsi. Berikut merupakan
- 10 contoh kode dari penggunaan Camera Plugin:

11 Berikut kode untuk menginstall Camera:

Kode 2.13: Kode untuk menginstal plugin Camera

```
12
13 1 npm install @capacitor/camera
```

Kode 2.14: Contoh kode Capacitor Camera

```
15 1 import { Camera, CameraResultType } from '@capacitor/camera';
16 2
17 3 const takePicture = async () => {
18 4   const image = await Camera.getPhoto({
19 5     quality: 90,
20 6     allowEditing: true,
21 7     resultType: CameraResultType.Uri
22 8   });
23 9
24 9
25 0   // image.webPath will contain a path that can be set as an image src.
26 1   // You can access the original file using image.path, which can be
27 2   // passed to the Filesystem API to read the raw data of the image,
28 3   // if desired (or pass resultType: CameraResultType.Base64 to getPhoto)
29 4   var imageUrl = image.webPath;
30 5
31 6   // Can be set to the src of an image now
32 7   imageElement.src = imageUrl;
33 8 }
```

35 Pada kode tersebut [2.14], variable image akan menggunakan metode `getPhoto()` di mana

36 metode ini berfungsi untuk mengambil sebuah gambar dari album atau menangkap gambar meng-

37 gunakan kamera. Properti `quality` merupakan kualitas dari gambar yang akan dikembalikan

38 dengan nilai berikisar dari 0 hingga 100 dalam bentuk JPEG. Properti `allowEditing` berfungsi

39 apabila pengguna ingin melakukan crop images pada gambar. Properti `resultType` berfungsi untuk

40 mengembalikan data dalam bentuk Base64, DataUrl, dan Uri.

41 Filesystem

42 Filesystem API menyediakan alat NodeJS-like untuk bekerja dengan file pada perangkat. Pengem-

43 bang dapat menggunakan plugin ini untuk melakukan operasi file umum seperti membaca, tulis,

44 dan mengelola isi direktori.

45 Berikut kode untuk menginstall Filesystem:

Kode 2.15: Kode untuk menginstal plugin Filesystem

```
46 1 npm install @capacitor/action-sheet
```

1 Berikut contoh penggunaan Filesystem Plugin:

Kode 2.16: Contoh kode dari penggunaan Filesystem

```

2  import { Filesystem, Directory, Encoding } from '@capacitor/filesystem';
3
4  const writeSecretFile = async () => {
5    await Filesystem.writeFile({
6      path: 'secrets/text.txt',
7      data: 'This is a test',
8      directory: Directory.Documents,
9      encoding: Encoding.UTF8,
10     });
11   };
12 }
13
14 const readSecretFile = async () => {
15   const contents = await Filesystem.readFile({
16     path: 'secrets/text.txt',
17     directory: Directory.Documents,
18     encoding: Encoding.UTF8,
19   });
20
21   console.log('secrets:', contents);
22 };
23
24 const deleteSecretFile = async () => {
25   await Filesystem.deleteFile({
26     path: 'secrets/text.txt',
27     directory: Directory.Documents,
28   });
29 }
30
31 const readFilePath = async () => {
32   // Here's an example of reading a file with a full file path. Use this to
33   // read binary data (base64 encoded) from plugins that return File URIs, such as
34   // the Camera.
35   const contents = await Filesystem.readFile({
36     path: 'file:///var/mobile/Containers/Data/Application/22A433FD-D82D-4989-8BE6-9FC49DEA20BB/Documents/text.txt',
37   });
38
39   console.log('data:', contents);
40 };

```

42 Preference

43 Plugin Preference pada Ionic 7 adalah alat yang memungkinkan Anda untuk menyimpan data
 44 sederhana dalam bentuk kunci-nilai yang dapat diakses secara bersamaan. Preferences API
 45 menyediakan area penyimpanan data yang mendukung kunci-nilai untuk aplikasi Ionic. Beberapa
 46 fitur utama dari Plugin Preference meliputi:

- 47 Mengatur grup preferences: Preferences groups digunakan untuk mengatur kunci-nilai pairs.
 48 Menggunakan nilai 'NativeStorage' memberikan kompatibilitas belakang dengan cordova-
 49 plugin-nativestorage.
 - 50 Mengakses hasil preferences: Menggunakan getResult() method sehingga mendapatkan nilai
 51 dari preferences yang terkait dengan kunci tertentu
 - 52 Menyimpan dan mengatur preferences: Menggunakan metode set() untuk menyimpan atau
 53 mengatur nilai preferences
- 54 Untuk menggunakan Plugin Preference dalam aplikasi Ionic, perlu menginstal plugin Capacitor
 55 Preferences dengan cara:

Kode 2.17: Kode untuk menginstal plugin Preference

```

56
57 1 npm install @capacitor/preferences

```

59 Berikut merupakan contoh kode dari penggunaan Preference:

Kode 2.18: Contoh kode dari plugin Preference

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

1

BAB 3

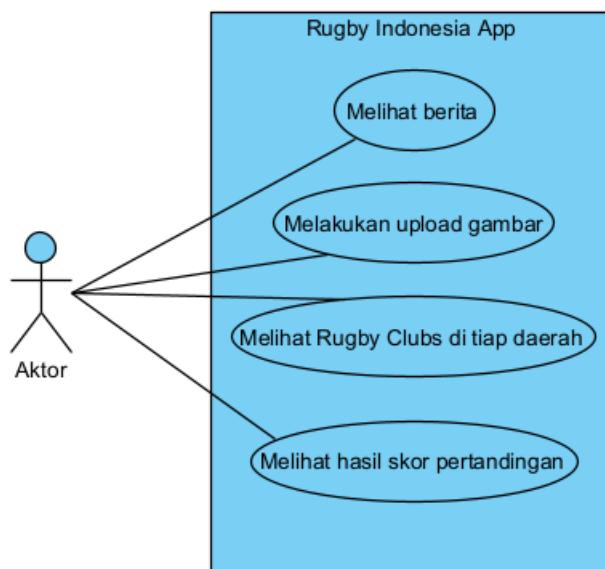
2

ANALISIS

- 3 Bab analisis ini akan menjelaskan tentang analisis dari aplikasi Rugby Indonesia yang sudah ada
4 serta analisis dari aplikasi Rugby Indonesia yang akan dibangun. Analisis yang akan dibahas pada
5 bab ini meliputi use case diagram, analisis kebutuhan sistem yang ada serta analisis kebutuhan
6 sistem usulan, serta analisis terhadap aplikasi Rugby Indonesia yang akan di bangun.

7 **3.1 Analisis Sistem Terkini**

- 8 Aplikasi Rugby Indonesia yang ada digunakan bagi para pemain rugby di Indonesia memiliki
9 halaman utama yang berisi berita-berita terkini terkait permainan rugby yang ada di Indonesia,
10 toolbar yang berisikan tombol home untuk kembali ke halaman utama, dan tombol menu untuk
11 berpindah ke menu lainnya. Selain berfungsi untuk membaca berita, aplikasi yang ada ini berfungsi
12 untuk menangkap gambar dan menguploadnya pada halaman teammate photo, melihat rugby clubs
13 yang berada di tiap daerah, serta dapat melihat skor dari tiap pertandingan.



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.

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 side menu yang berisi menu-menu dari aplikasi Rugby Indonesia.
4	Pengguna mengklik tombol “read more”	Aplikasi Rugby Indonesia akan menampilkan berita yang dipilih oleh pengguna secara lengkap.

6 3.1.2 Halaman Teammate Photos

- 7 Pada halaman teammate photos, pengguna dapat melihat serta mengupload gambar maupun
 8 tangkapan gambar yang ingin diunggah ke dalam halaman ini. Pengguna juga dapat memberikan
 9 frame terhadap foto yang ingin diunggahnya.

Tabel 3.2: Tabel Skenario dari Halaman Teammate Photos

No.	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol teammate photos pada side menu.	Aplikasi Rugby Indonesia menampilkan halaman Teammate Photos
2	Pengguna mengklik tombol “take a photo”	Aplikasi akan membuka camera
3	Pengguna mengklik tombol “load from library”	Aplikasi akan membuka library gambar
4	Pengguna mengklik gambar yang ada	Aplikasi akan menampilkan gambar tersebut lebih besar.

10 3.1.3 Halaman Rugby Clubs

- 11 Pada halaman rugby clubs, pengguna dapat melihat informasi terkait klub rugby yang berada di
 12 tiap daerah

Tabel 3.3: Tabel Skenario dari Halaman Rugby Clubs

No.	Aksi Aktor	Reaksi Sistem
1	Pengguna menekan tombol rugby clubs pada side menu.	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.

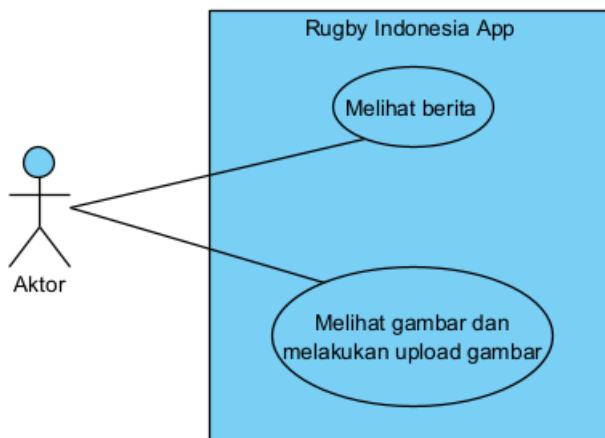
Tabel 3.4: Tabel Skenario dari Halaman Fixture & Results

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

- 4 Aplikasi Rugby Indoensia saat ini menggunakan framework Ionic versi 3, Apache Cordova, dan
- 5 Angular. Aplikasi ini memungkinkan untuk ditulis menggunakan bahasa pemrograman seperti
- 6 HTML, CSS, dan JavaScript.

⁷ 3.2 Analisis Sistem Usulan

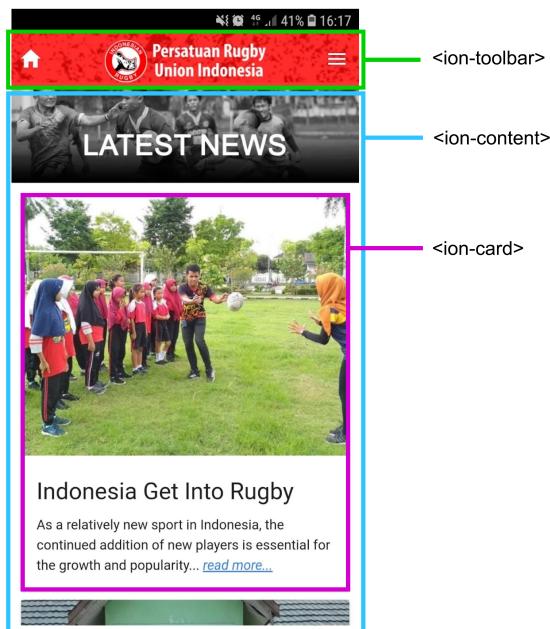
- 8 Aplikasi Rugby Indonesia yang akan dibangun hampir sama seperti aplikasi dengan sistem terkini,
- 9 namun dengan tidak adanya halaman Fixture & Results, Rugby Clubs dan fitur Push Notification
- 10 dikarenakan website dari Rugby Indonesia sendiri yang sudah dipangkas sehingga tidak memiliki
- 11 halaman-halaman tersebut.



Gambar 3.2: UCD Sistem Usulan Rugby Indonesia

¹² 3.2.1 Halaman Utama

- 13 Pada halaman utama, skenario dari penggunaan aplikasi Rubgy Indoensia pada halaman utama
- 14 sama seperti tabel skenario halaman utama dari sistem terkini [3.1]. Halaman ini memiliki beberapa
- 15 komponen yang terdapat pada Ionic 7, komponen-komponen yang terdapat pada halaman utama
- 16 ini yaitu komponen toolbar, content, serta card.



Gambar 3.3: Analisis dari Halaman Utama Rugby Indonesia

- ¹ Pada saat aplikasi Rugby Indonesia dijalankan, secara default aplikasi akan membuka file
² main.tsx yang berada di app/src. File tersebut akan merender App.tsx untuk menjadi tampilan
³ aplikasi.

Kode 3.1: Kode dari main.tsx

```
4
5 1 import React from 'react';
6 2 import { createRoot } from 'react-dom/client';
7 3 import App from './App';
8 4 import { defineCustomElements } from '@ionic/pwa-elements/loader';
9 5
10 6 // Call the element loader before the render call
11 7 defineCustomElements(window);
12 8
13 9 const container = document.getElementById('root');
14 10 const root = createRoot(container!);
15 11 root.render(
16 12   <React.StrictMode>
17 13     <App />
18 14   </React.StrictMode>
19 15 );
```

- Pada kode tersebut [3.1], metode `defineCustomElements` dipanggil dengan parameter `window`. metode `defineCustomElements` berfungsi sebagai element loader. Kode `document.getElementById('root')` digunakan untuk mendapatkan elemen dengan id 'root'. Kode `createRoot` digunakan untuk membuat root element di mana kode ini akan menerima elemen yang akan dijadikan root.

Kode `root.render` digunakan untuk merender aplikasi React di dalam root element yang sudah dibuat. Kode `<React.StrictMode>` digunakan untuk mengaktifkan Strict Mode. Ini membantu mendeteksi masalah pada waktu kompilasi dan memberikan peringatan tambahan saat pengembangan. Kode `<App />` adalah komponen utama yang dirender di dalam Strict Mode. Komponen ini yang akan merender file `App.tsx`.

Pada file `App.tsx`, file ini akan melakukan route ke tiap halaman menggunakan `IonRouteOutlet` [2.3.1]. Secara default, route utama dari `App.tsx` akan menuju ke dalam `path="/folder/:name"` di

- 1 mana path ini akan membuka Page.tsx. Pengembang dapat mengubah path dari route tersebut
- 2 dengan path yang dibutuhkan.

Kode 3.2: Kode dari App.tsx

```

3 import { IonApp, IonRouterOutlet, IonSplitPane, setupIonicReact } from '@ionic/react';
4 import { IonReactRouter } from 'ionic/react-router';
5 import { Redirect, Route } from 'react-router-dom';
6 import Menu from './components/Menu';
7 import Page from './pages/Page';
8
9
10 /* Core CSS required for Ionic components to work properly */
11 import '@ionic/react/css/core.css';
12
13 /* Basic CSS for apps built with Ionic */
14 import '@ionic/react/css/normalize.css';
15 import '@ionic/react/css/structure.css';
16 import '@ionic/react/css/typography.css';
17
18 /* Optional CSS utils that can be commented out */
19 import '@ionic/react/css/padding.css';
20 import '@ionic/react/css/float-elements.css';
21 import '@ionic/react/css/text-alignment.css';
22 import '@ionic/react/css/text-transformation.css';
23 import '@ionic/react/css/flex-utils.css';
24 import '@ionic/react/css/display.css';
25
26 /* Theme variables */
27 import './theme/variables.css';
28
29 /* This import is for page */
30 import LatestNews from './pages/latest_news/latest_news';
31 import TeammatePhotos from './pages/teammate_photos/teammate_photos';
32
33 setupIonicReact();
34
35 const App: React.FC = () => {
36   return (
37     <IonApp>
38       <IonReactRouter>
39         <IonSplitPane contentId="main">
40           <Menu />
41           <IonRouterOutlet id="main">
42             <Route path="/" exact={true}>
43               <Redirect to="/latest_news" />
44             </Route>
45             <Route path="/latest_news" exact={true}>
46               <LatestNews />
47             </Route>
48             <Route path="/teammate_photos" exact={true}>
49               <TeammatePhotos />
50             </Route>
51             <Route path="/folder/:name" exact={true}>
52               <Page />
53             </Route>
54           </IonRouterOutlet>
55         </IonSplitPane>
56       </IonReactRouter>
57     </IonApp>
58   );
59 }
60
61 export default App;

```

Pada kode tersebut [3.2], terdapat metode `setupIonicReact()` yang berguna untuk mengkonfigurasi Ionic agar bekerja dengan React. Komponen `IonSplitPane` pada kode berguna untuk memisahkan panel menu dengan halaman. Komponen `IonRouterOutlet` pada kode berguna untuk melakukan routing ke halaman yang diinginkan. Pada kode ini, secara default aplikasi akan membuka halaman ‘Latest News’ dikarenakan ketika path bernilai “/”, maka halaman yang dibuka adalah halaman Latest News.

Pada file `latest_news.tsx`, file ini akan mengambil berita dari <https://rugbyindonesia.or.id/berita/> dan akan ditampilkan pada halaman Latest News. Setiap berita yang terdapat pada

1 halaman Latest News akan ditampilkan menggunakan IonCard.

Kode 3.3: Kode dari latest_news.tsx

```

2 import { IonButtons, IonContent, IonHeader, IonMenuButton, IonPage, IonTitle, IonToolbar, IonBackButton, IonCard, IonCardHeader,
3     IonCardTitle, IonCardContent} from '@ionic/react';
4 import './latest_news.css';
5
6 import bannerImage from '../../../../../images/sub-header-news.png';
7 import latestNews1 from '../../../../../images/latest_news_image_1.jpeg';
8 import homeIcon from '../../../../../images/home_icon.png';
9
10
11 const Page: React.FC = () => {
12
13     return (
14         <IonPage>
15             <IonHeader>
16                 <IonToolbar>
17                     <IonButtons slot="start">
18                         <a href="/latest_news">
19                             <div className='home-icon'>
20                                 <img src={homeIcon} alt="home-icon" />
21                             </div>
22                         </a>
23                     </IonButtons>
24                     <IonButtons slot="end">
25                         <IonMenuButton />
26                     </IonButtons>
27                 <IonTitle>
28                     Persatuan Rugby
29                     Union Indonesia
30                 </IonTitle>
31             </IonToolbar>
32         </IonHeader>
33         <IonContent className="ion-padding">
34             <div className="news_section">
35                 <img src = {bannerImage} alt='latest-news-banner' />
36             </div>
37             <IonCard>
38                 <img alt='latest-news-image-1' src={latestNews1}/>
39                 <IonCardHeader>
40                     <IonCardTitle>Rugby Masuk Sekolah resmi dimulai di DKI Jakarta</IonCardTitle>
41                 </IonCardHeader>
42
43                 <IonCardContent>
44                     Program Rugby Masuk Sekolah resmi dimulai di DKI Jakarta dengan serah terima Bola dan Baju Pelatih Rugby Masuk
45                     Sekolah dari PB PRUI ke PRUI DKI Jakarta pada Hari Sabtu, 25 November 2023 di Lapangan Pondok Bambu,
46                     Jakarta. Wakil Ketua II PB PRUI, Pak Agus Djamhoer menyerahkan paket Rugby Masuk Sekolah ini kepada Pak
47                     Tito Vau selaku Ketua PRUI DKI Jakarta pada acara Kejuaraan Daerah Rugby tingkat Pelajar DKI Jakarta.
48                     <br><br><br><br>
49                     DKI Jakarta memiliki 17 pelatih yang sudah mengikuti sertifikasi pelatih Rugby Masuk Sekolah dan siap
50                     mengajarkan T1 Rugby ke seluruh tingkatan sekolah di Jakarta. Pada saat ini tercatat sudah lebih dari 13
51                     sekolah di Jakarta dan jumlah sekolah ini akan terus ditingkatkan seiring dengan waktu program ini
52                     berjalan.
53                 </IonCardContent>
54             </IonCard>
55         </IonContent>
56     </IonPage>
57 );
58 };
59
60 export default Page;

```

62 Kode ini 3.3 akan menampilkan halaman Latest News yang memiliki komponen toolbar, button,
63 content, dan card. Content dari halaman ini berisi berita-berita yang terdapat pada <https://rugbyindonesia.or.id/berita/>.

65 3.2.2 Halaman Teammate Photos

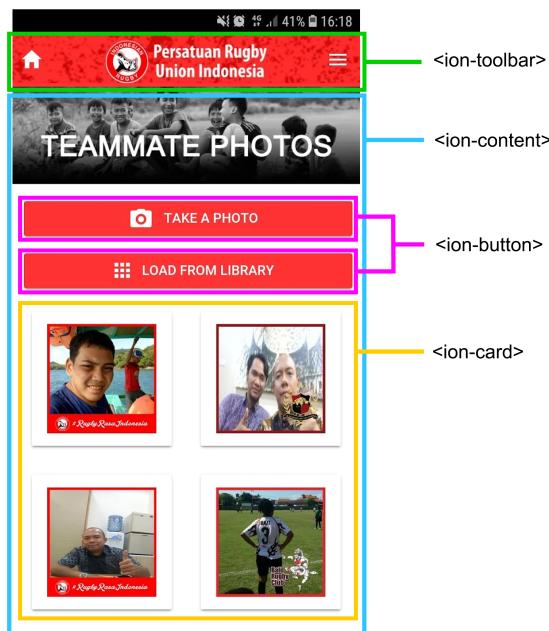
66 Pada halaman Teammate Photos, skenario dari aplikasi Rugby Indonesia sama dengan skenario
67 dari halaman Teammate Photos yang sudah ada [3.2], namun dengan adanya sedikit penambahan,
68 yaitu skenario di saat pengguna mengupload gambar. Penambahan ini dilakukan karena peneliti
69 tidak menemukan gambar di saat pengguna mengunggah gambar.

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.

1 Skenario tersebut [3.5] juga berlaku ketika pengguna menekan tombol “LOAD FROM LIBRARY”,
 2 hanya saja perbedaannya terdapat pada reaksi sistem, yaitu Aplikasi akan membuka gallery dari
 3 smartphone pengguna.

4 Pada halaman Teammate Photos, terdapat beberapa komponen dari Ionic 7 yang digunakan.
 5 Halaman ini memiliki beberapa komponen, yaitu toolbar, button, content, dan juga card.



Gambar 3.4: Analisis dari Halaman Teammate Photos

6 Pada saat pengguna memilih menu Teammate Photos, aplikasi akan menjalankan file teamma-
 7 te_photos.tsx. File ini memiliki fungsi untuk menangkap gambar, dan melihat hasil tangkapan
 8 gambar milik pengguna tersebut atau pun orang lain.

Kode 3.4: Kode dari teammate_photos.tsx

```

9 import React, { useState } from 'react';
10 import { IonButtons, IonContent, IonHeader, IonMenuButton, IonPage, IonTitle, IonToolbar, IonBackButton, IonButton, IonIcon} from
11   '@ionic/react';
12
13 import { Camera, Photo, CameraResultType, CameraSource } from '@capacitor/camera';
14 import { Directory, Filesystem } from '@capacitor/filesystem';
15
16 import './teammate_photos.css';

```

```

17
28 import bannerImage from '../../../../../images/sub-header-photo.png';
39 import homeIcon from '../../../../../images/home_icon.png';
40
51 import { PhotoImages } from "./photoImages";
62 import PhotoGallery from './photoGallery';
73 import { appsSharp, camera, menuSharp } from 'ionicons/icons';
84
95 const Page: React.FC = () => {
106   const [images, setImages] = useState<PhotoImages[]>([[]]);
117
128   const takePicture = async () => {
139     const image = await Camera.getPhoto({
140       quality: 90,
141       allowEditing: true,
142       resultType: CameraResultType.Uri
123     });
184
195     const fileName = new Date().getTime() + '.jpeg';
206
227     const savedFileImage = await savePicture(image, fileName);
228     var imageUrl = savedFileImage.filePath!;
229     console.log(imageUrl);
240     // Can be set to the src of an image now
251     // imageElement.src = imageUrl;
262     setImages([...images, savedFileImage]);
273
284   };
295
306   const savePicture = async(photo: Photo, fileName: string):Promise<PhotoImages> => {
317     let base64data:string;
328     base64data = await base64FromPath(photo.webPath!);
339
340     const savedPicture = await Filesystem.writeFile({
351       path: fileName,
362       directory: Directory.Data,
373       data: base64data
384     });
395
406     return{
417       filePath: fileName,
428       webviewPath: photo.webPath
439     }
440   }
451
462   async function base64FromPath(path:string): Promise<string>{
473     const response = await fetch(path);
484     const blob = await response.blob();
495
506     return new Promise((resolve, reject) => {
517       const reader = new FileReader();
528       reader.onerror = reject;
539       reader.onload = () => {
540         if (typeof reader.result === 'string')
551         {
562           resolve(reader.result);
573         }
584         else
595         {
606           reject('This method did not return a string');
617         }
628       };
639
640       reader.readAsDataURL(blob);
651     })
662   }
673
684   return (
695     <IonPage>
706       <IonHeader>
717         <IonToolbar>
728           <IonButtons slot="start">
739             <a href='/latest_news'>
740               <div className='home-icon'>
751                 <img src={homeIcon} alt="home-icon" />
762               </div>
773             </a>
784           </IonButtons>
795           <IonButtons slot="end">
806             <IonMenuButton />
817           </IonButtons>
828         <IonTitle>

```

```

89             Persatuan Rugby
90                 Union Indonesia
91             </IonTitle>
92         </IonToolbar>
93     </IonHeader>
94     <IonContent className="ion-padding">
95         <div className='teammate-photos-banner'>
96             <img src={bannerImage} alt="Teammate-Photos-Banner" />
97         </div>
98         <br/><br/>
99         <IonButton color="primary" expand="block" onClick={onClick => takePicture()}>
100             <IonIcon slot="start" icon={camera}></IonIcon>
101             Take Photo
102         </IonButton> /* Use the onClick prop directly */
103
104         <IonButton color='primary' expand='block'>
105             <IonIcon slot="start" icon={appsSharp}></IonIcon>
106             Load from Library
107         </IonButton>
108
109         <PhotoGallery photos={images} />
110     </IonContent>
111 </IonPage>
112 );
113 };
114
115 export default Page;

```

29 Pada kode tersebut [3.4], kode `const [images, setImages] = useState<PhotoImages[]>([[]);` menggunakan useState untuk menyimpan daftar foto yang diambil dalam bentuk objek
30 PhotoImages. Fungsi `takePicture` menggunakan Capacitor Camera untuk mengambil foto.

Kode 3.5: Kode dari photoImages.tsx

```

32 export interface PhotoImages
33 {
34     filePath: string;
35     webviewPath?: string;
36 }
37

```

Kode 3.6: Kode dari photoGallery.tsx

```

39 import { IonCol, IonGrid, IonImg, IonRow } from "@ionic/react";
40 import { PhotoImages } from "./photoImages";
41 import React from "react";
42
43 type Props = {
44     photos: PhotoImages[],
45 }
46
47
48 const PhotoGallery: React.FC<Props> =({photos}) => {
49     return (
50         <IonGrid>
51             <IonRow>
52                 {photos.map((photo, idx) =>(
53                     <IonCol size = "6" key={idx}>
54                         <IonImg src={photo.webviewPath}/>
55                     </IonCol>
56                 )));
57             </IonRow>
58         </IonGrid>
59     );
60 }
61
62
63 export default PhotoGallery;

```


DAFTAR REFERENSI

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

```

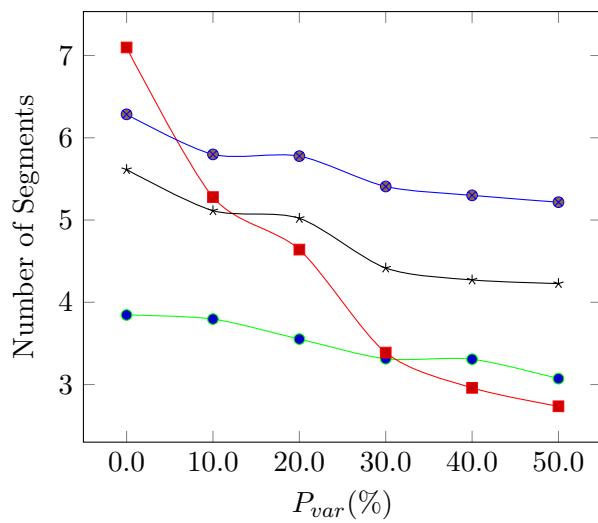
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id;                                //id of the set
8     protected MyEdge FurthestEdge;                   //the 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

```

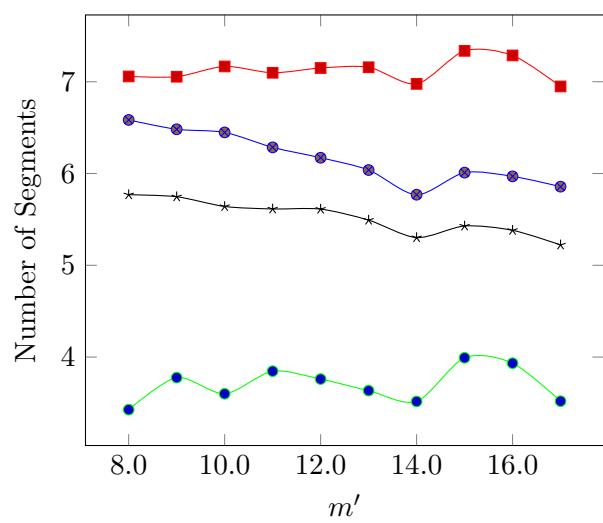

LAMPIRAN B

HASIL EKSPERIMENT

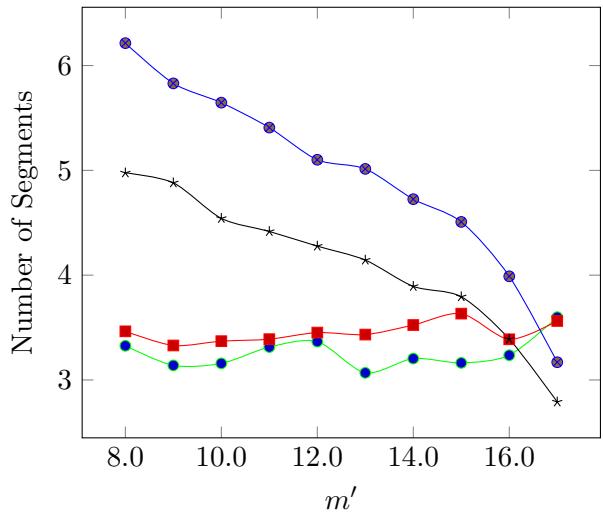
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



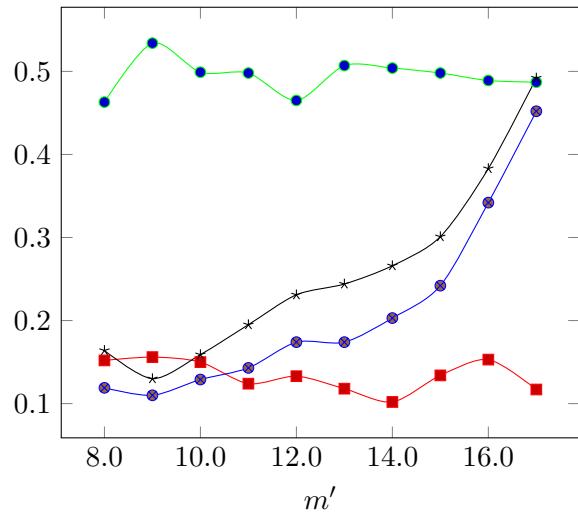
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4